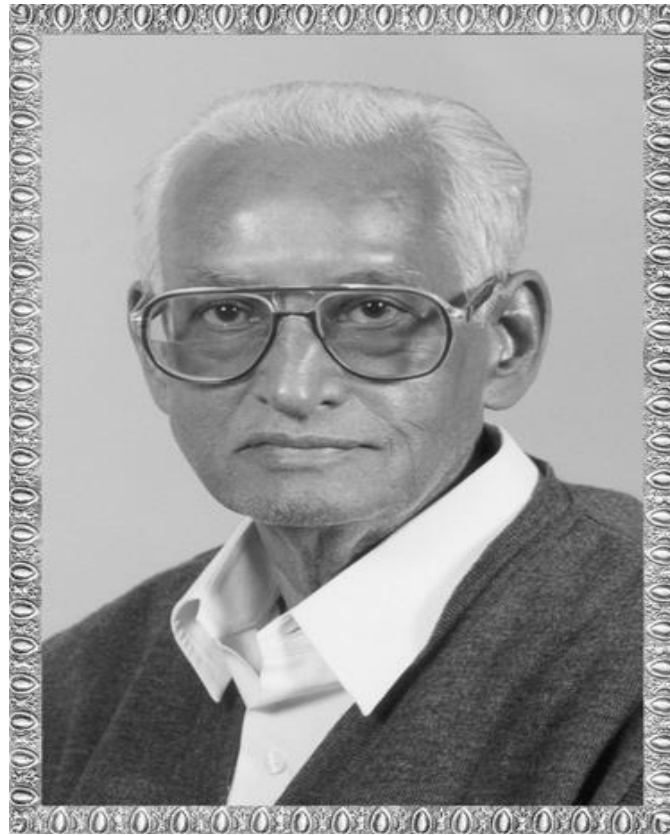


SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)



Lt. Shree Chimanbhai Shukla

B.C.A./B.Sc.I.T. - SEMESTER-1

SUB :- PROGRAMMING IN C

Shree H.N.Shukla College
2, Vaishali Nagar,
Nr. Amrapali Railway
Crossing,
Raiya Road, Rajkot.
Ph. (0281)2440478



Shree H.N.Shukla College
3, Vaishali Nagar,
Nr. Amrapali Railway
Crossing,
Raiya Road, Rajkot.
Ph. (0281)2471645

Website: www.hnsgroupofcolleges.org

Email : hnsinfo@hnshukla.com

CHAPTER-1

➤ INTRODUCTION TO C LANGUAGE

- ✚ Introduction of computer languages.
- ✚ Introduction of programming concept.
- ✚ Introduction of C language (History and Overview).
- ✚ Difference between traditional and modern C.
- ✚ C character set.
- ✚ C tokens.
- ✚ Keywords
- ✚ Constants
- ✚ Strings
- ✚ Identifiers and variables
- ✚ Operators (all 8 operators)
- ✚ Hierarchy of operators
- ✚ Type casting
- ✚ Data types in C
- ✚ Pre-Processors in C

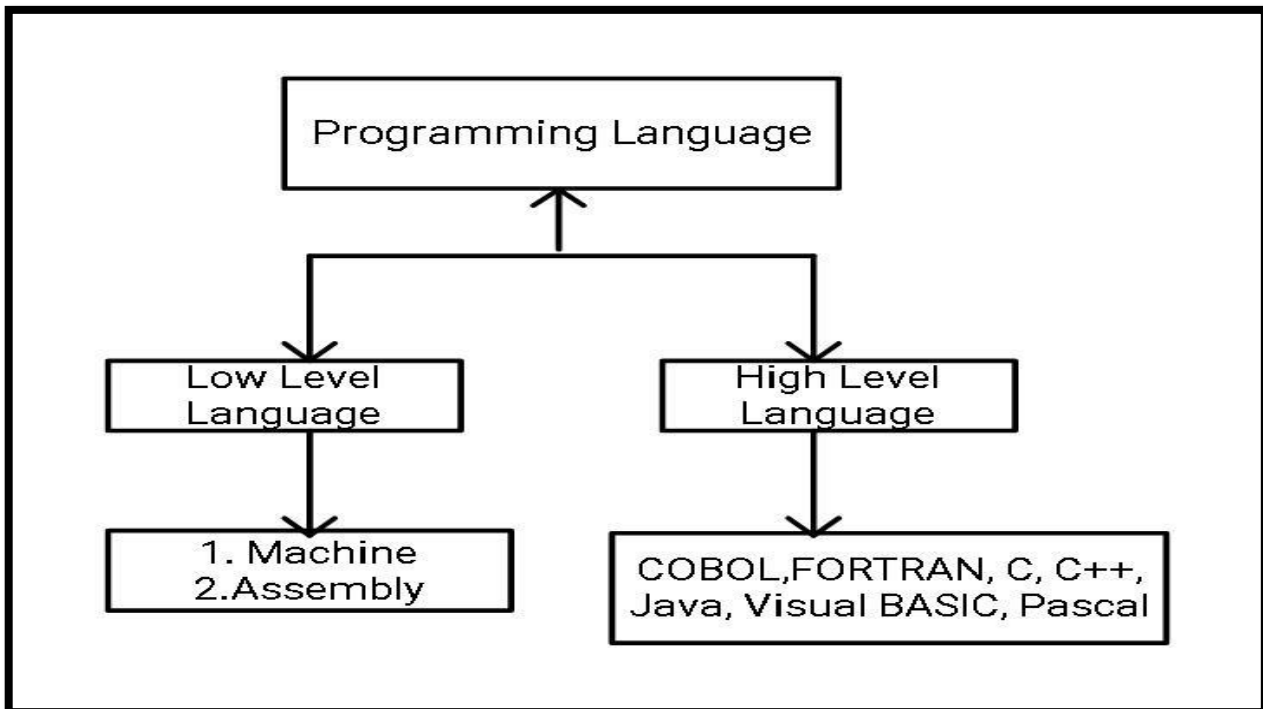
SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-1 Introduction of computer languages:

Title	About Computer Languages
Trailer (Only for understanding)	<ul style="list-style-type: none">• What is Computer Language?• How Many Types of Computer Languages?• Machine Language means ?• Assembly Language means ?• Whom you Called High level Language?

By Diagram :



Detail :- (For Exam Content)

Computer languages are divided into two categories:

- (i) Low Level Languages
- (ii) High Level Languages

(i) Low Level Languages or Machine Oriented Languages:

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Low level languages can be of two types:

- **Machine language**
- **Assembly language.**

- **Machine language programming:**
 - ❖ Machine language is written using Binary Language (0 & 1).
 - ❖ Computer can understand only low level language.
 - ❖ So that computer need to convert it.

- ❖ **Advantages of Machine language programming:**
 1. It does not need to convert.
 2. It need less memory.
 3. Programs are small & easy.

- ❖ **Disadvantages of Machine language programming:**
 1. The programs are difficult to understand and debug.
 2. Programs are not portable.
 3. Programs are hard to describe.

- **Assembly language:**
 - ❖ Assembly language is written using set of instructions.
 - ❖ We need assembler to convert the code.

- ❖ **Advantages of Assembly language:**
 1. The programs are easy to understand.
 2. Instructions can be written using set of code.
 3. It need less time to represent.

- ❖ **Disadvantages of Assembly language:**
 1. We need assembler to translate the code.
 2. Programs are not portable.

(ii) High Level Languages or problem oriented languages:

- ❖ High level languages are written using English Statements.
- ❖ Computer have to convert high level language into low level language.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

✓ **Advantages of High level language :**

1. Programs are portable.
2. Programs are easy to understand.
3. We need less time to write the programs.

✓ **Disadvantages of High level language :**

1. It require more memory.
2. We need translator to convert the code.
3. It is compulsory to understand structure of program.

Learning outcome (summary)	<ul style="list-style-type: none">• Two types of Computer languages (Low level and High Level)• Low Level include Machine and Assembly level languages• There is no need to convert the code written in machine language.
-----------------------------------	---

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	How Many Types of Computer Languages Available?	2(Two)
2	Machine Language can be written using	0 and 1 (Binary)
3	_____ is used to convert Assembly Language into Machine Language	Assembler
4	High Level Languages are written using _____	English Statements
5	_____ is used to convert High Level Language into Machine Language	Interpreter or Compiler
6	_____ Language is not suitable for Large Application	Machine Language
7	Low Level Languages are further divide into _____ and _____	Machine Level Assembly

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-2 Introduction of programming concept:

Title	What is Programming
Trailer (Only for understanding)	<ul style="list-style-type: none">• Programming means ?• Programming includes what?• What should be the structure of program

By Diagram :



Detail :- (For Exam Content)

- ❖ Programming means **“Sequence of Instructions”**.
- ❖ The basic programming of C language include mainly the following :
 - Structure of Program :-**
 - ❖ In C language we can write our program with given structure only.
 - Variable Declaration :-**
 - ❖ We can declare variables to store different values in the programming.
 - Conditional & Looping Statements :-**
 - ❖ The conditional statements are used to give conditions.
 - ❖ The looping statements are used to repeat some values at the sametime.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Learning outcome (summary)	<ul style="list-style-type: none">• Programming means “Sequence of Instructions.• The basic programming language includes:<ul style="list-style-type: none">✓ Structue of Program✓ Variable Declaration✓ Conditional & Looping Statement.
-----------------------------------	--

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Programming means	Sequence of Instruction
2	Basic Programming include mainly	Structure of Program
3	Different Programming Languages can have different Structure?(Yes/No)	yes

Q-3 Write note on History of C Language.

Title	History of C Language
Trailor (Only for understanding)	<ul style="list-style-type: none">• Founder of C Lanugage?• Where C was Developed?• Founder and developing year of different languages.

By Diagram :

History Of C

- In 1972 C was evolved from **ALGOL**, **BCPL** and **B** by **Dennis Ritchie** at the Bell Laboratories.
- C** uses many concepts from theses languages and added the concept of data types and other powerful features.
- It is strongly associated with **UNIX**.
- UNIX** is one of the most popular **network operating systems** in use today.

fppt.com

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :- (For Exam Content)

- ❖ C is one type of Programming Language.
- ❖ It was developed at “AT & T’s Bell Laboratories”.
- ❖ **Dennis.M.Ritchie** was the founder or father of C Language.
- ❖ In 1960 , There were many languages available but only for some specific purpose.For Example :- COBOL
- ❖ At that time people decided to develop particular language that is used for all purpose.
- ❖ There are following languages available at that time for different types of use.

SR NO	YEAR	LANGUAGE	DEVELOPER
1	1960	ALGOL	INTERNATIONAL COMMITTEE
2	1963	CPL	CAMBRIDGE UNIVERSITY
3	1967	BPCL	MARTIN RICHARDS & CAMBRIDGE UNIVERSITY
4	1970	B	KENT THOMPSON AT AT & T LABORATORY
5	1972	C	DENIS RITHCHIE AT AT & T LABORATORIES
6	1978	K&R C	Kernighan andRitchie
7	1989	ANSI C	ANSI Committee
8	1990	ISO C	ISO Committee

Learning outcome (summary)

- C Was developed by Dennis M. Ritchie.
- It was developed in 1972 at AT & T’s Bell Laboratory.
- C Language was developed from B Language.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

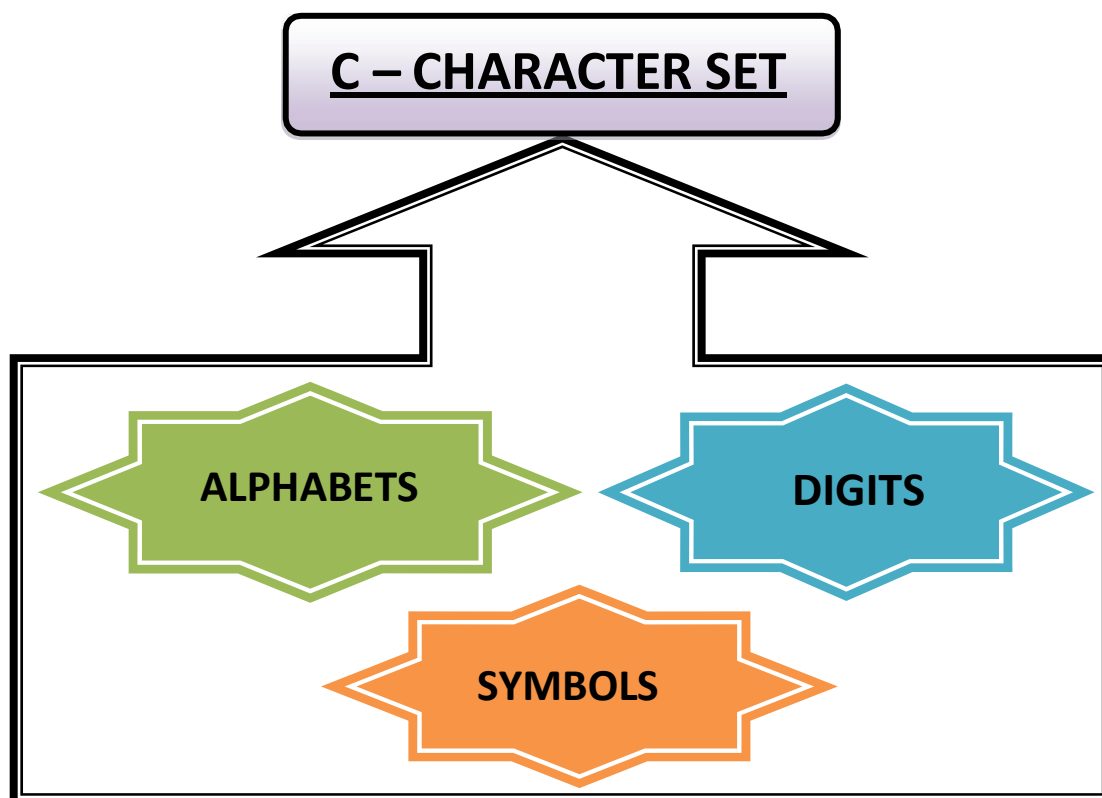
1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Who is Founder of C Language?	Dennis M.Ritchie
2	In Which year C Language was developed?	1972
3	Where C Language was developed?	At & T's Bell Laboratories
4	C Language is rich set of _____	Built – in Functions & Operators

Q-4 Briefly Explain C character set:

Title	Character set of C
Trailer (Only for understanding)	<ul style="list-style-type: none">• C Character set made up of?• What Alphabet Consist of ?• What Digit Consist of ?• What Symbol Consist of ?

By Diagram :



SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :- (For Exam Content)

- ❖ The Character set in c language always used to **represent the information.**
- ❖ The Character set in c language support unique value for each character.
- ❖ The Character set in c language also include **white space character.**
- ❖ The C Character set include mainly the following:

(i) Alphabets

(ii) Digits

(iii) Symbols

✓ **Alphabets :-**

- ❖ C language support different alphabets which can be **A to Z (incapital letters) and a to z (in small letters).**

✓ **Digits :-**

- ❖ In C language we can input sequence of numbers which can be between **0 to 9.**

✓ **Symbols :-**

- ❖ We can also **input some special symbols** as information in C language.
- ❖ The different symbols can have different meanings.

Symbol	Meaning
~	Tilde
!	Exclamation mark
#	Number Sign
\$	Dollar sign
%	Percent sign
^	Caret
&	Ampersand
*	Aestrik
(Left Parenthesis
)	Right Parenthesis
-	Underscore
<	Opening angle bracket
>	Closing angle bracket

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

?	Question mark
,	Comma
.	Period

Learning outcome (summary)	<ul style="list-style-type: none"> • C Character set includes mainly Alphabets , Digits and Symbols. • Digits can be between 0 to 9. • Alphabets can be between A to Z and a to z. • Symbol can be any special Character.
-----------------------------------	---

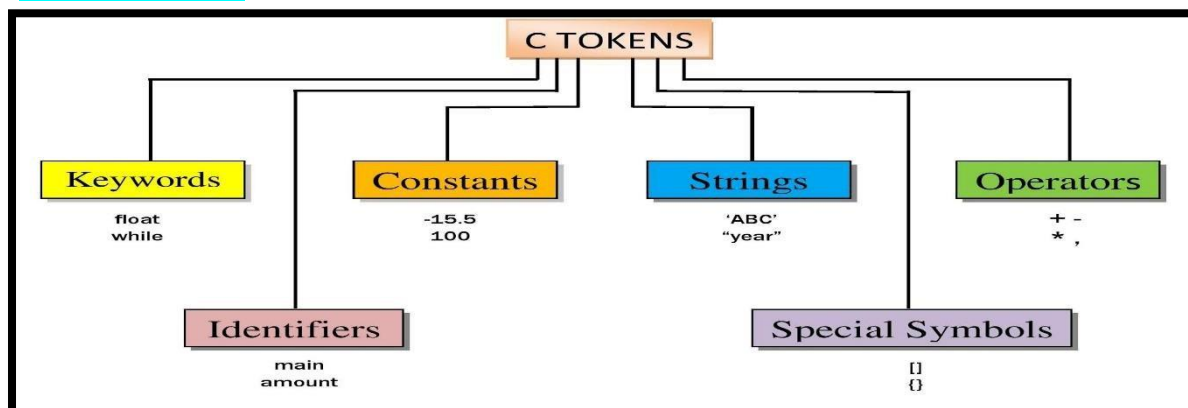
1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Character Set includes mainly _____, _____ & _____	Alphabets , Digits & Symbols
2	Digits include mainly _____	0 to 9
3	Alphabets include mainly _____	A to Z and a to z
4	Is White Space a Character?(Yes/No)	Yes

Q-5 Explain Tokens of C.

Title	What is Token?
Trailer (Only for understanding)	<ul style="list-style-type: none"> • How many token available in C? • What are the uses of each token? • Each token have its own behavior. • Different tokens have different meaning In programming.

By Diagram :



SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :- (For Exam Content)

- ❖ In a source c program, the basic elements recognized by the compiler is “token”.
- ❖ A token is **source-program text** that the compiler does not break down into component elements.

There is six types of token in C language.

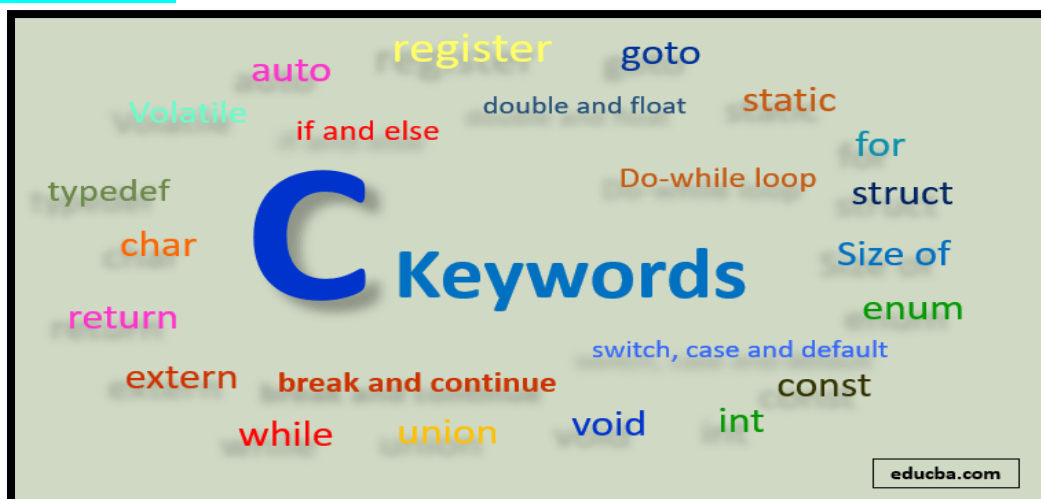
- ❖ **Keyword**
- ❖ **Identifier**
- ❖ **Constants**
- ❖ **Strings**
- ❖ **Special symbols**
- ❖ **Operators**

Learning outcome (summary)	<ul style="list-style-type: none">• Smallest individual unit of C Program is called Tokens.• We can also say “building blocks of program”.• Total 6 Tokens available in C.• Tokens must be recognized by Compiler.
-----------------------------------	---

Q-6 Explain Keywords in Brief.

Title	What is Keyword?
Trailer (Only for understanding)	<ul style="list-style-type: none">• Keyword Means What?• How many keywords available in C?• Each keyword have its own meaning• Keywords are also known as ?

By Diagram :



SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :- (For Exam Content)

- ❖ Predefined Words or **Reserve Words** are known as **keywords**.
- ❖ We **cannot change** the meaning of keywords.
- ❖ Keywords are the words that have **specific Meaning**.
- ❖ All keywords must be written in **Lowercase**.
- ❖ Keywords serve as basic building blocks for program statements.
- ❖ There are total **32** keywords available in C.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	volatile
const	float	short	unsigned

Learning outcome (summary)

- Keywords are called **reserve words** or **predefined words**.
- Total **32** Keywords available in C.
- Total 6 Tokens available in C.
- Tokens must be recognized by Compiler.

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Keyword Means	Reserve Words
2	How many keywords available in c?	32
3	Can we use keyword as variable name?(Yes/No)	No

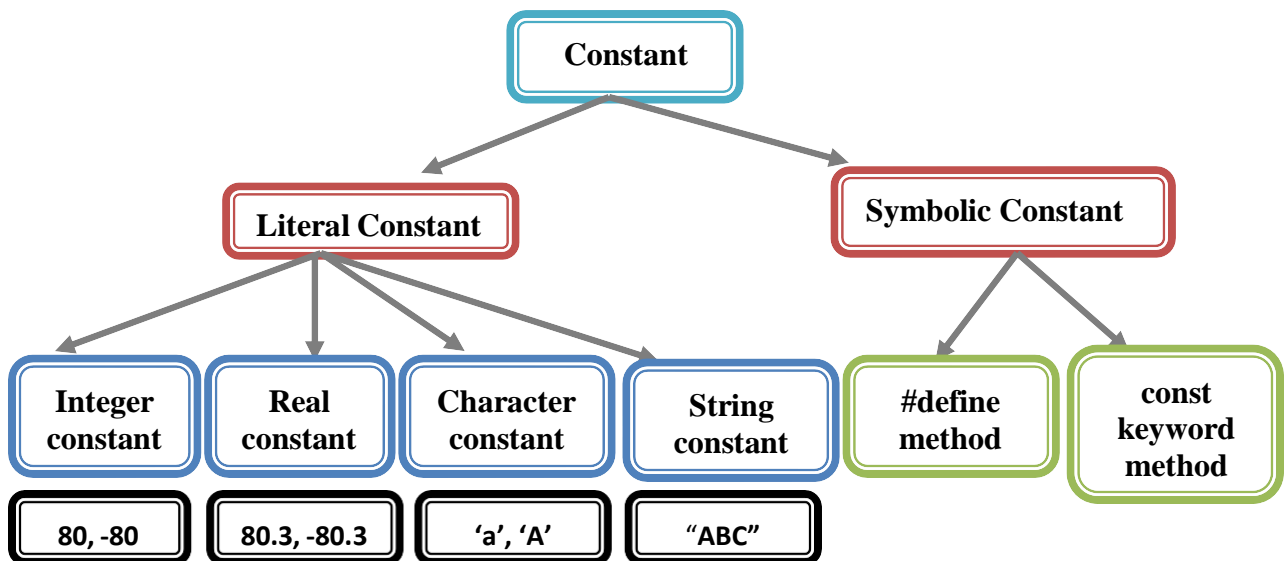
Q-7 Explain Constant in C.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Title	What is Constant?
Trailer (Only for ❖ C understanding)	<ul style="list-style-type: none">• Constant Means What?• How many types of constant available in C?• Is it possible to change value of constant during execution of the program.

By Diagram :



Detail :- (For Exam Content)

- ❖ Constant means the **fixed value**.
- ❖ The value of constant **can not be changed** during execution of program.
- ❖ There are two types of constant available in c.

1. Literal Constant:

- ❖ Literal constant is the value that is **typed directly into your program wherever it is needed**.
- ❖ Literal constant is divided further into 4 categories that is;
 - **integer constant,**
 - **real constant,**
 - **character constant,**
 - **string constant**

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

2. Symbolic Constant:

- ❖ A symbolic constant is a constant that is represented by name, just as a variable is represented.
- ❖ The real value of symbolic constant must be entered **only once when it is first defined.**
- ❖ C language has two methods of defining symbolic constant.

2.1 #define method:

- ❖ #define directive is one of C language preprocessor directive.
- ❖ There is no '=' sign while assigning a value to the variable while using #define.

Example: #define pi 3.14

#define X "hello hw r u"

2.2 const keyword method:

- ❖ **Const** keyword is used for variable declaration.
- ❖ To declare any variable using const keyword we can use equal sign (=). **Example: const x=5.5**

Learning outcome (summary)	<ul style="list-style-type: none">• Constant means "fixed value".• Value of Constant never change during execution of the program.• Two types of constants are there :- Literal & Symbolic• "Const" keyword can be used to declare variable in symbolic constant.
-----------------------------------	--

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Constant Means	"fixed Value"
2	Value of Constant can never change during execution of program?(True/False)	True
3	Types of Constant	2(two) Literal and Symbolic
4	Is following declaration true in case of literal constant ? Ex :- int i=90; Symbolic constant can be represented by either	True

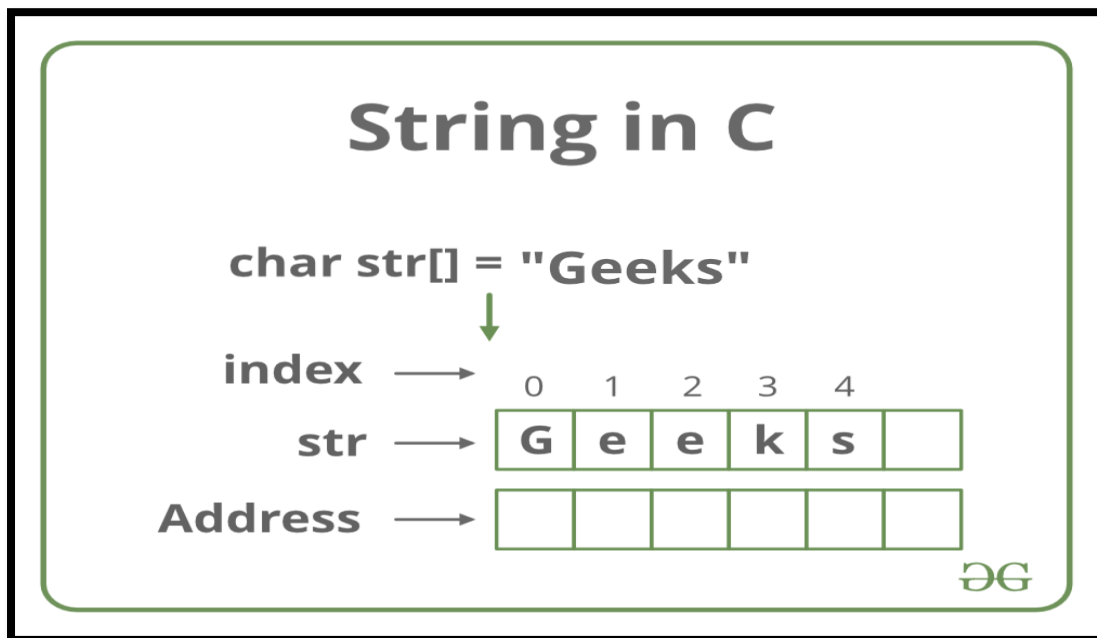
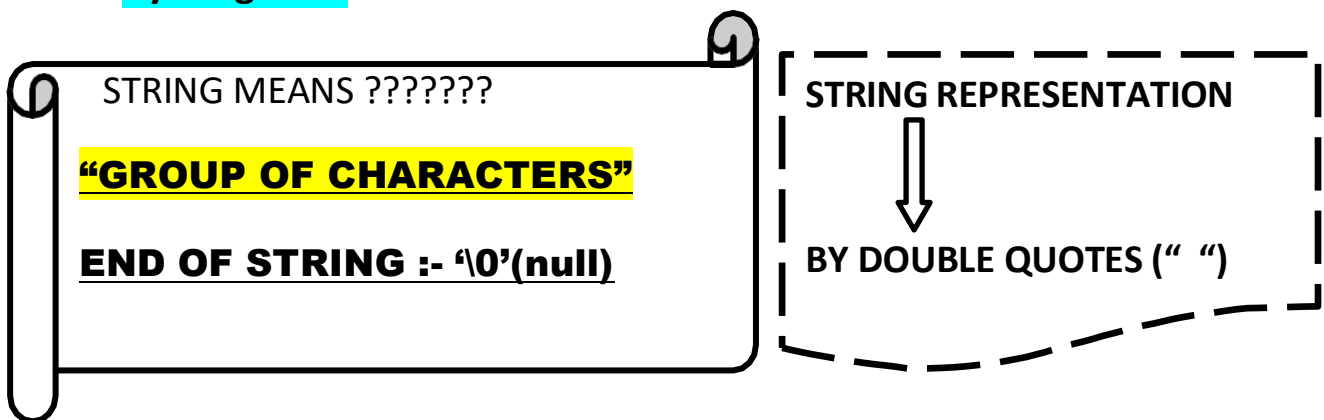
SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Explain String in C.

Title	What is String?
Trailer (Only for understanding)	<ul style="list-style-type: none">• String Means What?• How to represent End of the String?• String can be represented by ?• How to declare String Array?

By Diagram :



Detail :- (For Exam Content)

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

- ❖ In C language “**bunch of characters** “ is called string.
- ❖ In C language string can be represented by **double quotes (“ ”)**.
- ❖ To indicate end of the string ‘\0’ (**NULL**) character can be used.
- ❖ C will automatically created array of characters.
- ❖ Each character in the array will occupies **1 byte** in the memory.
- ❖ In C language we can declare string value like following :-

Example: char string[]="hello".

h	e	l	l	o
2000	2001	2002	2003	2004

- ❖ While entering the string using scanf() following points should be kept in mind:
 - 1) The length of string can not be more than size of character array.
 - 2) The string can be **fixed or dynamic**.

Learning outcome (summary)	<ul style="list-style-type: none">• String means “Group of Characters”.• String can be represented by ““(Double Quotes)• End of the string can be represented by ‘\0’(Null) Character.• User can create Fixed or Dynamic String.• String also known as “Array of Characters”.
-----------------------------------	---

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	String Means_____	“Array of Characters”
2	End of string can be represented by special character called????	Null character(‘\0’)
3	In c language , string can be denoted by????	Double Quotes(“ “)
4	Each character in String Array can occupy _____byte in memory.	One byte
5	How to declare String Array or Character Array?	Char a[] = “hello”;

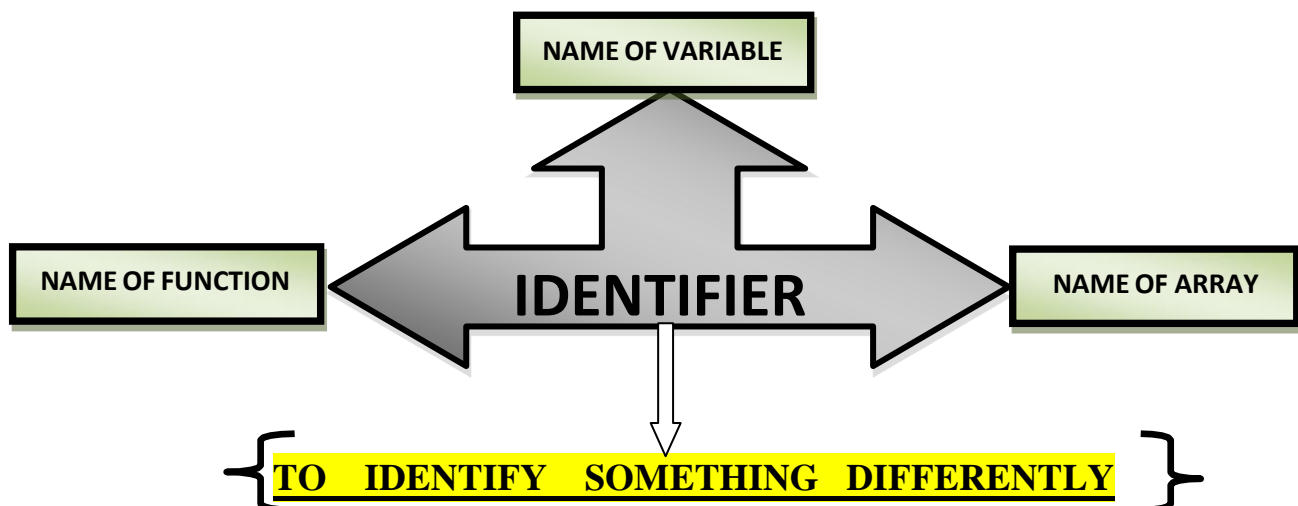
Q-9 Explain Identifier (Variable) in C.

Title	What is String?
Trailer (Only for understanding)	<ul style="list-style-type: none">• String Means What?• How to represent End of the String?• String can be represented by ?• How to declare String Array?

By Diagram :

IDENTIFIERS

- Names of variables, functions and arrays
- User – defined
- Sequence of letters and digits
- Lower case letters are preferred
- Letter should be the first character
- Example : #define N 10, #define a 15



SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :- (For Exam Content)

- ❖ Identifier is used to indicate **name of variable , function and array.**
- ❖ Variable is one type of identifier that used to **store some value.**
- ❖ Variable or Identifier can support different values of different types.
- ❖ Identifier can be **recongized by user defined name.**
- ❖ There are some rules for declaring variable or identifier.

Namig Rules for identifiers:

- 1) First character must be **an alphabet.**
- 2) Identifier Must contains **letters, digits and underscore.**
- 3) Identifier name can be maximum of **31 characters.**
- 4) Keyword **can not be used** as variable name.
- 5) **White space** does not allow as variable name.

Example:

```
int age = 20;
```

Note :- Here , age is identifier

Learning outcome (summary)

- Identifier means Give name to Variable , Function or Array.
- There are different rules for identifier.
- Identifier name can be maximum of 31 characters.
- Identifier includes Letters , digits & underscore.
- Identifier do not allow white space.

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Identifier Means	“To Identify Something”
2	Identifier Consist of _____ & _____	Letters & digits
3	Identifier must not contains _____	White Space
4	Only Permitted symbol in identifier is _____	Underscore(_)

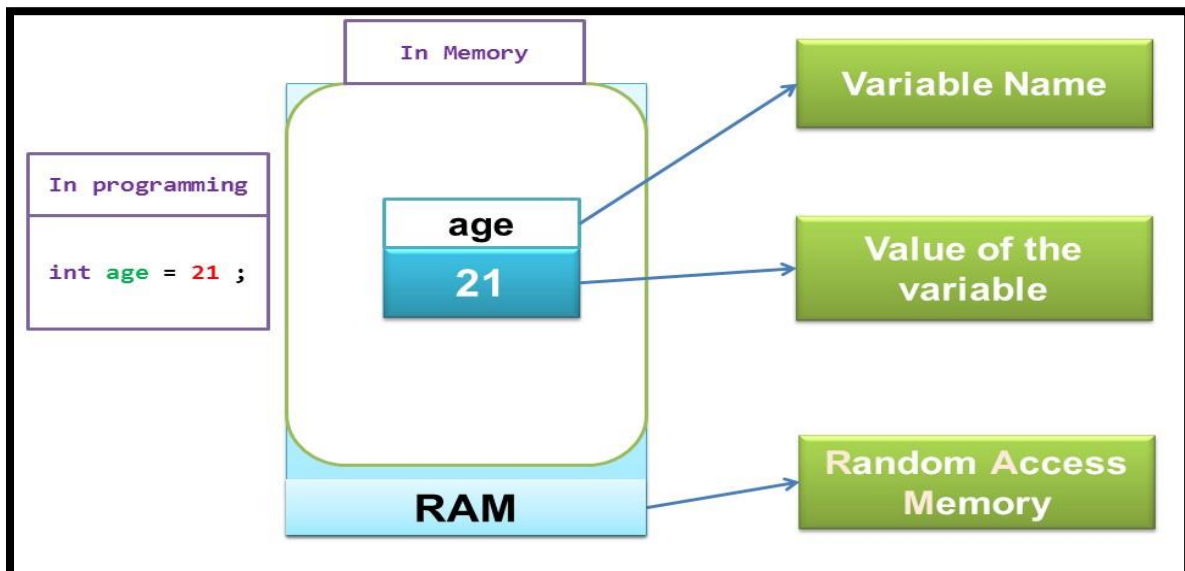
SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-10 Explain Identifier (Variable) in C.

Title	What is Variable?
Trailer (Only for understanding)	<ul style="list-style-type: none">• Variable Means What?• Why we should declare variables in the program.• What are the basic attributes of any variable ?• Does Each variable have unique Memory Address?

By Diagram :



Detail :- (For Exam Content)

- ❖ Variables is a container that used **to store data or value.**
- ❖ A variable may take different values at different times during execution.
- ❖ All the variables have three basic attributes:
 - **Name**
 - **Value**
 - **Address of memory location**

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

❖ The example of declaring the variable is given below:

```
int a=30;  
Float b;
```

❖ Rules for defining variables

- A variable can have **alphabets, digits, and underscore**.
- A variable name can start with the alphabet, and **underscore only**.
- **It can't start with a digit**.
- **No whitespace** is allowed within the variable name.
- A variable name must not be any **reserved word or keyword**, e.g. int, float, etc.

Learning outcome (summary)	<ul style="list-style-type: none">• Variable is a name of Storage Location.• It is used to store different types of values.• Any variable must have: Name , Value & Address• White space can not be used in Variable name.
-----------------------------------	---

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Variable is used to store different types of _____	values
2	Variable can store on different _____	Memory Location
3	Variable can have _____	Unique Address

Q-11 Explain Operators in detail.

Title	What is Operator?
Trailer (Only for understanding)	<ul style="list-style-type: none">• Operator Means What?• Operators generally operates on?• How many types of operators ?• Purpose of different operators?

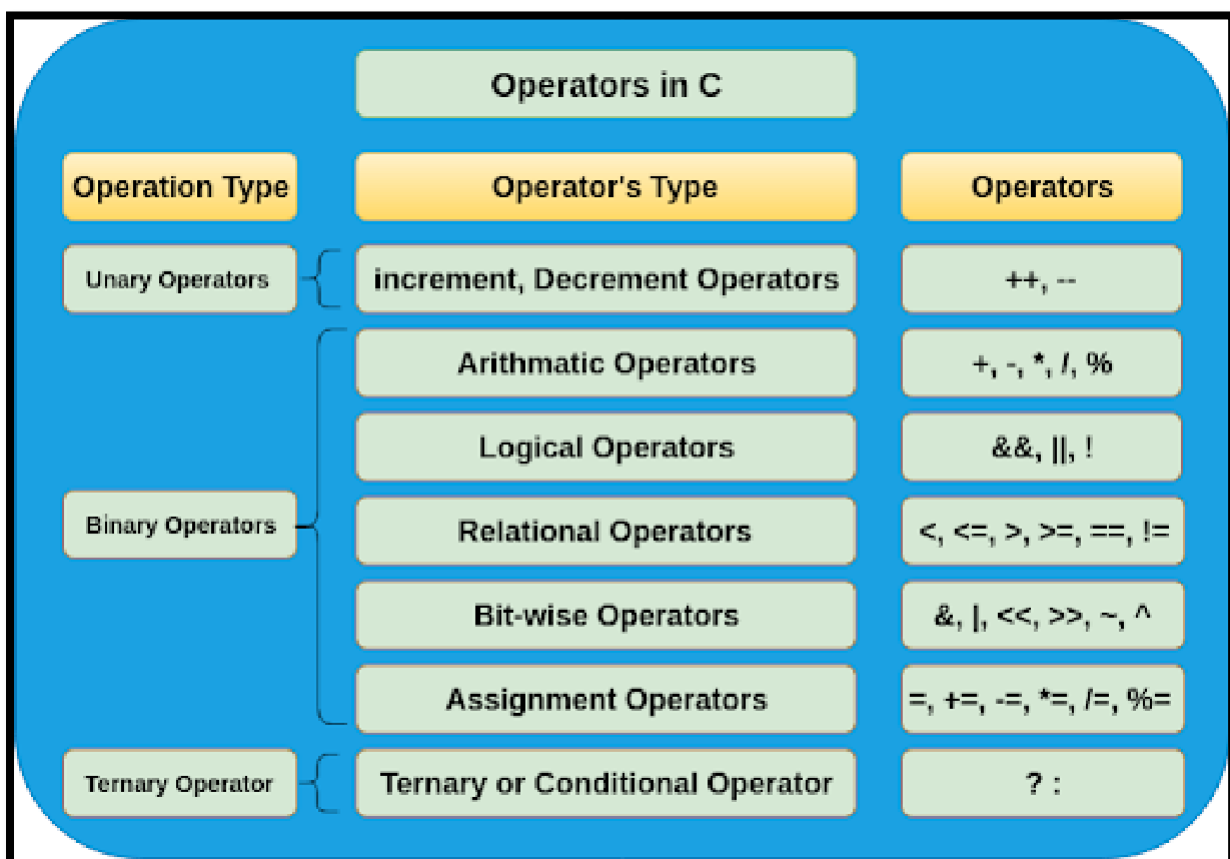
SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

By Diagram :

Operator means to “Operate something”.

Operator can have different Operand or Values



Detail :- (For Exam Content)

- ❖ Operators are used to indicate **some operations on different values**.
- ❖ Operator means to perform some operations on the data or values. C Language support mainly the following operators...
 - **Arithmetic Operator**
 - **Relational Operator**
 - **Logical Operator**

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

- **Assignment Operator**
- **Bitwise Operator**
- **Conditional Operator**
- **Special Operator**
- **Increment / Decrement Operator**

(i) **Arithmetic Operator :-**

- The Arithmetic operators are used to perform some **arithmetic operations on the value.**
- There are following arithmetic operators available in c.

Operators	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus

(ii) **Relational Operator :-**

- Relational operators are also known as **comparison operators.**
- It returns only one output at a time.
- There are following relational operators available in c.

Operators	Meaning
>	Greater than
<	Less than
>=	Greater than & Equal to
<=	Less than & Equal to
==	Equal to
!=	Not Equal to

(iii) **Logical Operator :-**

- The logical operators are operators that return only one output at a time, **that is true or false.**
- There are following logical operators available in c.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Operators	Meaning
&&	Logical AND
	Logical OR
!	Logical NOT

(iv) Assignment Operator :-

- The assignment operator is used to assign value to the variable.
- There is only one assignment operator available in c that is **=(Equalto)**
- Example :- int i=10;

(v) Bitwise Operator :-

- Bitwise operator is used to store the data **at bit level**.
- The following are bitwise operators supported by C language.

Operators	Meaning
&	Bitwise AND
	Bitwise OR
^	Bitwise Exclusive
<<	Shift Left
>>	Shift Right
~	One's Complement

(vi) Conditional Operator :-

- Conditional Operators are also known as **ternary operators**.
- There are only two conditional operators available in c.

Operators	Meaning
?	Questiona Mark (True)
:	Colon (False)

Example :- (a>b) ? Printf(“a is max”) : (“b is max”)

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

(vii) Special Operator :-

- The special operators in c language are used for **some specialpurpose.**
- There are following special operators available in c.

(i) **Comma Operator :-** This operator is used to combine value with variable.

EX :- `printf("a=%d",a);`

(ii) **Sizeof Operator :-** This operator is used to find out how many bytes ,value or variable can occupies.

EX :- `sizeof(12);`

(viii) Increment / Decrement Operator :-

- The increment operator is used to increase the value and decrement operator is used to decrease the value.
- In C Language , ++ is called increment operator and - - is called decrement operator.

Operators	Meaning
++a	Prefix increment
a++	Postfix increment
--a	Prefix decrement
a--	Postfix decrement

Learning outcome (summary)

- Operator means to Operate something.
- Operators can operate on different values.
- Logical Operators can be : And , Or ,Not
- There is only one assignment operator available in c that is Equal to (=).
- SizeOf operator is used to find out how many bytes a variable can occupies.
- Increment operator is ++ and Decrement operator is --.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Conditional Operator in c language can be _____ ? and : and _____	
2	_____ operator can find how many bytes an operand can occupies	Sizeof
3	_____ operator can used to link related expression together.	Comma
4	_____ operator is known as assignment operator	=(equal to)
5	If condition is false , logical operator can return	0

Q-12 Explain Hierarchy of Operators in detail.

Title	What is Hierarchy of Operators?
Trailer (Only for understanding)	<ul style="list-style-type: none">• Associativity Means What?• How many types of Priority Levels?• Which Operator Execute First ?• When the Associativity Rules should be apply?

By Diagram :

Precedence & Associativity of Operators

CATEGORY	OPERATORS	ASSOCIATIVITY
Parenthesis/brackets	() [] -> . ++ --	Left to right
Unary	! ~ ++ -- + - * & (type) sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Bitwise Shift	<< >>	Left to right
		< to right

34

C Programming

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :- (For Exam Content)

- ❖ In C Language there are **variety of operators**.
- ❖ In C Language each operator having **priority of operating**.
- ❖ There can be low level and high level priority.
- ❖ High level operator will **be execute first** and low level operators will **beexecute last**.
- ❖ The **priority rules** can be different for different operators.
- ❖ Operator precedence determines the grouping of terms in an expression and decides how an expression is evaluated.
- ❖ "There is no Hierarchical order for logical operators. Logical operators are used to integrate different conditional expressions.
- ❖ C operators are listed in order of precedence (highest to lowest). Their associativity indicates in what order operators of equal precedence in an expression.
- ❖ **The hierarchy of operators are as follow.**

Sr.No.	Operator	Meaning	Priority
1	(), []	Function Brackets	1
2	/, *	Division , Multiplication	2
3	%	Modulus	3
4	+, -	Addition , Subtraction	4
5	<<, >>	Shilf left , Shift Right	5
6	>, <, =, !=	Relational Operator	6
7	=	Assignment Operator	7
8	&&, , !	Logical Operator	8

Learning outcome (summary)

- Variable is a name of Storage Location.
- It is used to store different types of values.
- Any variable must have:
Name , Value & Address
- White space can not be used in Variable name.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	The Precedence of operator is called?	Hierarchy
2	Associativity rules are applied when operators have _____	Same level
3	Operators of the same precedence can be evaluated from _____ as well as from _____	Left to Right Right to Left

Q-13 Explain Type Casting in brief.

Title	What is Type Casting?
Trailer (Only for understanding)	<ul style="list-style-type: none">• Type Casting Means What?• How many types of Type Casting Available?• Whom you called Automatic Conversion?• What is the use of type cast operator?

By Diagram :

TYPE CASTING MEANS \Rightarrow **“DATA TYPE CONVERSION”**

“Converting data from one datatype to another”

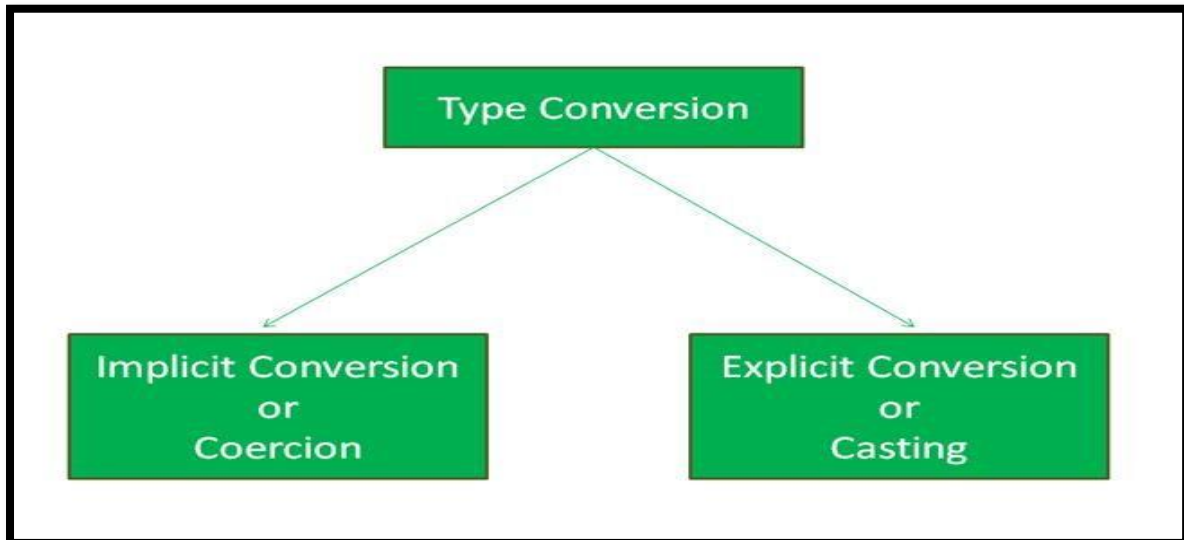


Detail :- (For Exam Content)

- ❖ Type Casting is process of **converting one data type into other data type.**
- ❖ There are two types of type casting available :

Implicit Type Conversion

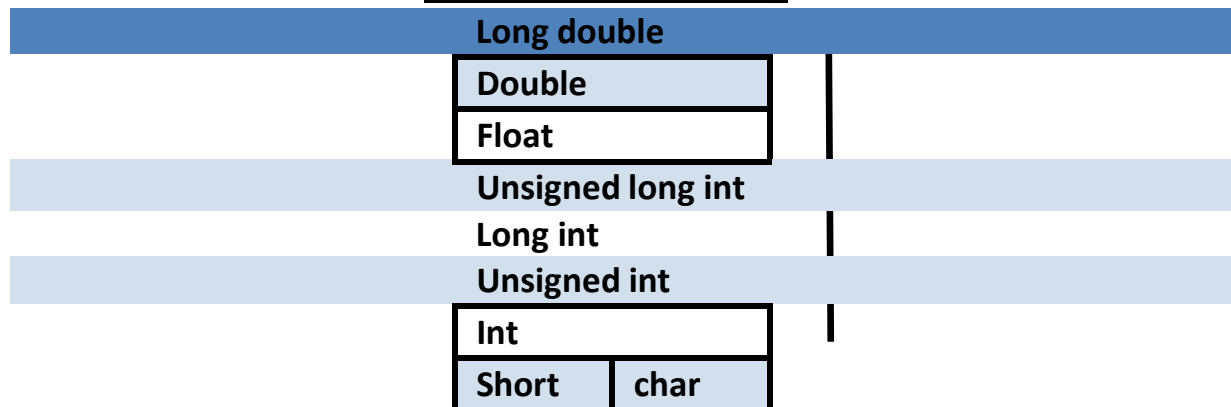
Explicit Type Conversion



1. Implicit Type Conversion :

- ❖ The implicit type casting is also known as **automatic conversion**.
- ❖ The implicit conversion is used to convert original data type into other data type.
- ❖ In this type of casting data should be converted **from smaller datatype to larger data type**.

Conversion Hierachy



2. Explicit Type Conversion :

- ❖ Sometimes we have to convert different data type into single data type.
- ❖ But Explicit casting is not automatic conversion because we have to use **typecast operator**.

- ❖ We can change type of the data from smaller to larger and larger to smaller. Ex :- b = (float) a;

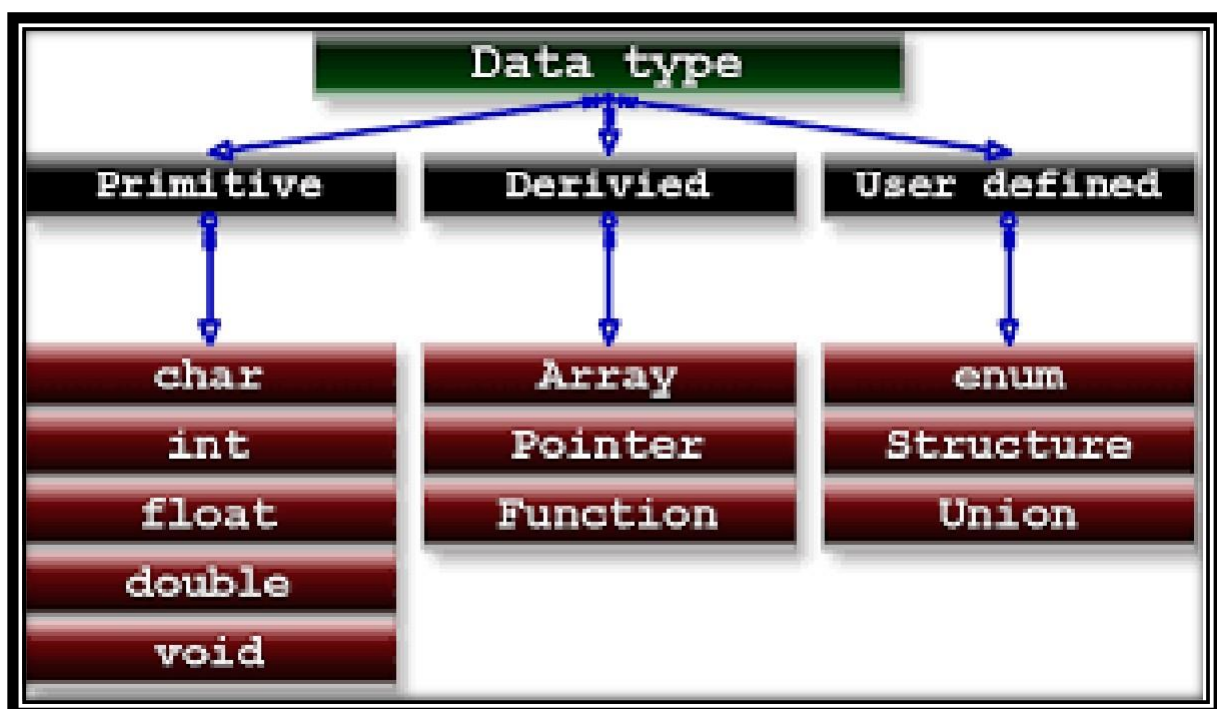
Learning outcome (summary)

- Type Casting means converting data from one data type to another.
- Two types of Type Casting : Implicit & Explicit
- Implicit Conversion also known as Automatic Conversion.
- In Explicit Conversion , we have to use Typecast operator to perform conversion.

O-13 Explain Data Type in detail.

Title	What is Data type?
Trailer (Only for understanding)	<ul style="list-style-type: none">• Data type Means What?• How many data types available in C?• How many bytes occupied by particular data type?• What Void Data type returns?

By Diagram :



SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :- (For Exam Content)

- ❖ In C Language, The data types are used to **store different types of values** .
- ❖ There are following data types available in C.
- ❖ **1.Primary / Basic / Fundamental Data types. (Integer, Character, Float etc**
1. Derived Data types. (Array, Pointer)
2. User Define Data types. (Structure, Union, enum, typedef)

1. Primary Data Type :

- The primary data types are also called fundamental or built-in or basic data types.
- The primary data types include mainly following :

(i) Integer :-

- This data type is used to store only integer value.
- It occupies **2 bytes** of memory.

(ii) Float :- (Long)

- This data type is used to store integer value & Decimal Value.
- It occupies **4 bytes** of memory.

(iii) Char :-

- This data type is used to store single or multiple characters.
- It occupies **1 byte** of memory.

(iv) Double :-

- This data type is used to store integer and decimal values.
- It occupies **8 bytes** of memory.

(v) long double :-

- This data type is used to store integer and decimal values with high range.
- It occupies **10 bytes** of memory.

2. User Defined Data Type :

- User defined data type is a type of data that decided by user self.
- User defined data type include mainly the following.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

(i) Typedef :-

- It provide facility to represent identifier of particular data type.

(ii) Enum :-

- It provide facility to declare multiple constant at the same time.

(iii) Structure:-

- It is used to create member of different data type at the same time.

(iv) Union:-

- It is used to create member of different data type at the same time.

<u>SR.NO.</u>	<u>DATA TYPE</u>	<u>MEMORY SIZE</u>	<u>RANGE</u>
<u>1</u>	Char	1 Byte	-128 to +127
<u>2</u>	Int	2 Byte	-32768 to +32767
<u>3</u>	Long	4 Byte	-2,147,483,648 to +2,147,483,647
<u>4</u>	Float	4 Byte	1.2 e -38 to 3.4 e +308
<u>5</u>	Double	8 Byte	2.3 e -308 to 1.7 e +308
<u>6</u>	Long Double	10 Bytes	3.4 e -4932 to 1.1 e +4932

Learning outcome (summary)

- Data type is used to decide type of data.
- Three categories of data types available :
Primary , User defined & derived Data type.
- Integer data type occupies 2 bytes in memory.
- Data types are used to store different types of values.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

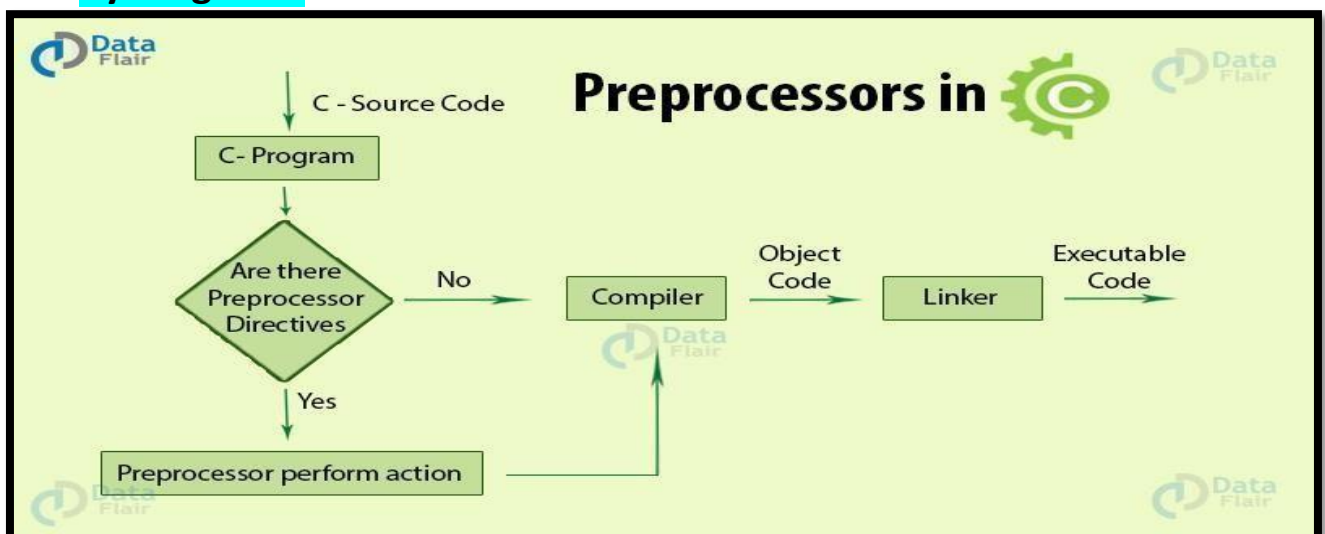
1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Data types are used to decide type of_____	Data or Value
2	C Language support_____,_____& _____categories of Data Type.	Primary User defined Derived
3	Typedef means _____	Type Definition
4	List our User defined Data types in C.	Structure Union Enum
5	Integer datatype occupies _____bytes of Memory	2
6	Float datatype occupies_____byptes of Memory	4
7	Character Datatype occupies _____byte of Memory	1

Q-14 Explain Pre- Processor in detail.

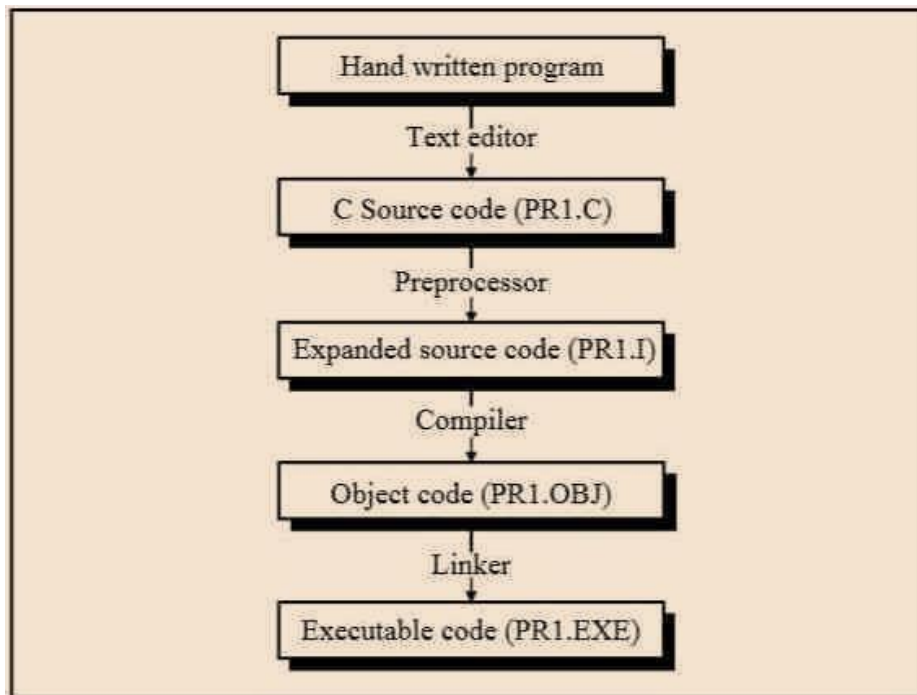
Title	What is Pre - Processor?
Trailor (Only for understanding)	<ul style="list-style-type: none"> • Pre - Processor Means What? • How many Pre - Processors available in C? • Any Pre – Processor begins with? • Pre – Processor Directives are executed directly by?

By Diagram :



SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)



Detail :- (For Exam Content)

- ❖ Pre – Processor directives are executed **directly by compiler**.
- ❖ Any pre-processor in C language always begins **with #(Hash) Sign**.
- ❖ C language support following pre-processor directives.

1. #include :

- #include is pre-processor that use **to include header files** in our program.
- If we not include particular header file then compiler give error related to **“prototype”**.
- Example :- #include<stdio.h>
- To declare any header file you have to use **angular brackets (< >)**.

2. #define :

- #define pre – processor is used to **create symbolic constant**.
- **You can not change value** of constant once it is defined.
- You can not re-define the value of **constant using # define**.
- Example :- #define pi 3.14

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Note :- If you use # define then statement will not ends with

Learning outcome (summary)	<ul style="list-style-type: none">• Pre- Processor directives are executed directly by Compiler.• There are two types of Pre-Processor directives available :- #define , #include• #define Pre- Processor is used to define symbolic constant.• #include Pre-Processor is used to include header file in the program.
-----------------------------------	--

1 Word Question – Answer

QUESTION	ANSWER
Pre-Processor of C language are :	# define , # include
Any pre-processor always begins with____Sign	#(Hash)
_____Pre-Processor is used to add Header File.	# include
_____ Pre-Processor is used to declare any contant or fixed value.	# define

PART – 2 INTRODUCTION OF LOGIC DEVELOPMENT TOOLS

Q-15 List out different logic development techniques and Explain Flow chart.

Title	Logic Development?
Trailor (Only for understanding)	<ul style="list-style-type: none">• What is Logic?• How many Logic Development Technique available?• What is Flow Chart?• How many Symbols Flow chart have?

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

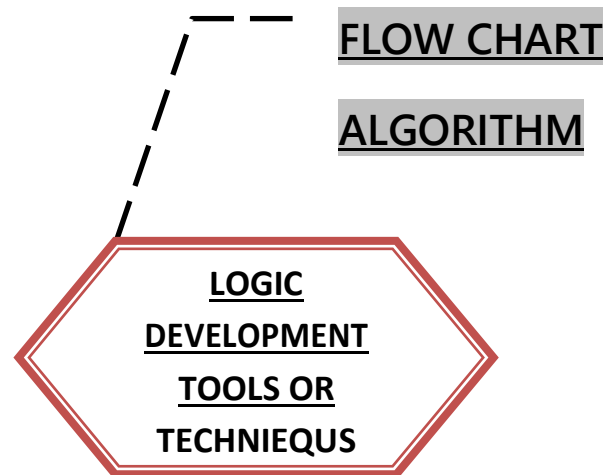
(Affiliated to Saurashtra University & Gujarat Technological University)

By Diagram :

WHAT IS LOGIC ??????

Logic is a tool to develop reasonable conclusions based on a given set of data. **OR**

“Way of thinking about something”



Detail :- (For Exam Content)

- ❖ Logically developed means that **the ideas in your paper**, the way you support these ideas and the connections you make between ideas make logical sense.
- ❖ Logic is a tool to develop reasonable conclusions based on a given set of data.
- ❖ In other words , **Logic is a “way of thinking about something”**.

Flow Chart :-

- ❖ Flow chart is **graphical representation of our data**.
- ❖ It also give idea about sequence of flow.
- ❖ Flow chart can represent logic flow and process of data.

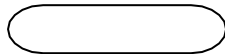
Flowchart Symbols:

- Flow chart support following symbols to represent the logic.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

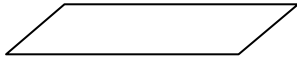
(Affiliated to Saurashtra University & Gujarat Technological University)

1. Terminator:



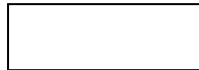
- This symbol represents beginning and ending point in the program.

2. Input/Output :



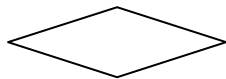
- It is used to represent input and output of the data.

3. Process :



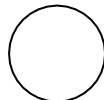
- This symbol is used to represent different process of operations.

4. Decision Symbol:



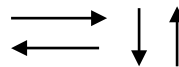
- This symbol is used to give condition and take decision.

5. Connector Symbol:



- It is used to represent logic flow of out data.

6. Data Flow Symbol / Flow Direction:



- It is used to represent direction flow of our data.

By Diagram :

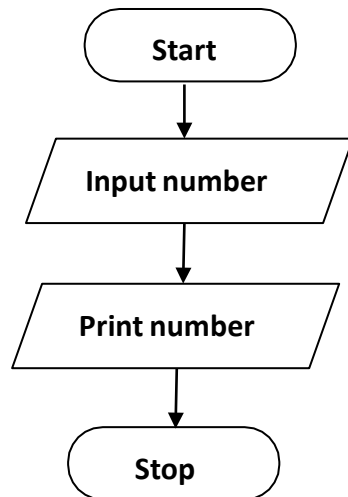
Symbols	Name	Functions
	Compute/Process	Instructions or internal operations inside the processor or memory
	Input/Output	Indicating the process of input and output of data
	Decision	Indicates a question at which decision has to be made and answered in the format of (Yes/No , True/False)
	Connector(Inspection)	This symbol is typically small and used as a connector to show a jump from one point in the process flow to another.
	Predefined Process	Shows named process which is defined elsewhere
	Terminal(Start/End)	It indicates the beginning and end of the program and process.
	Flow Lines	Shows the Process's order of operation or sequence of instruction

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Example:

A flowchart to print the number.



Learning outcome (summary)

- Logic means “Way of Thinking about Something”.
- There are two logic development techniques available : Flow Chart , Algorithm
- Flow chart is a graphical representation of data.
- Flow chart support mainly 6 symbols.

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	List out logic development tools or techniques	Flowchart ,Algorithm
2	_____ is graphical representation of the code.	Flow Chart
3	Which Symbol is used to represent process in Flow chart.	<input type="text"/>


SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-16 Explain Dry Run :

Title	Dry Run
Trailer (Only for understanding)	<ul style="list-style-type: none">• What is Dry Run?• Is it possible to perform manual compilation of code.• Is it possible to test the code without computer?

By Diagram :

DRY RUN :- 

“ PROCESS OF APPLYING PROGRAMMING LOGIC ON PAPER.” OR “MANUAL COMPILATION OF CODE”

Detail :- (For Exam Content)

- ❖ Dry run is nothing but **manual compilation of code**.
- ❖ You can test your program without using a computer by **dry running it on paper**.
- ❖ You act as the computer – following the instructions of the program, recording the values of the variables at each stage.
- ❖ You can do this with a **table** .
- ❖ The table will have column headed with the names of the *variables* in the program.
- ❖ Each row in the table will be labelled with a line number from the program.
- ❖ In this table you can record all relevant changes to the variables as the program progresses, **thereby test the logic of the program / algorithm**.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

- ❖ Do a dry run before you code your program on computer this way any logic errors will come to light during the dry run

Example:-

- * L1 Declare two variables , first num second num
- * L2 Initialise both variables to 0
- * L3 first num = 0 second num = 0
- * L4 Ask user to enter first number
- * L5 Assign user input to first num variable
- * L6 Ask user to enter second number
- * L7 Assign user input to second num variable
- * L8 Add first num to second num
- * L9 Prin
- * t result

Learning outcome (summary)	<ul style="list-style-type: none">• Dry Run means “Manual compilation of Code.”• Using Dry Run , You can test your program without running code on computer.• Simply , Write your logic on paper and then apply it on computer to test it.
-----------------------------------	--

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	What is Dry run?	“Manual Compilation of Code.”
2	In Dry run , you can test your program by performing steps on paper.(Yes/No)	Yes

Q-17 Explain Algorithm :

Title	Algorithm
Trailer (Only for understanding)	<ul style="list-style-type: none">• What is Algorithm?• What are the features of Algorithm?• Is it detail sequence of simple steps.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

By Diagram :

ALGORITHM :-

“DETAIL SEQUENCE OF SIMPLE STEPS” OR “IT

CONSIST SET OF FINITE STEPS”

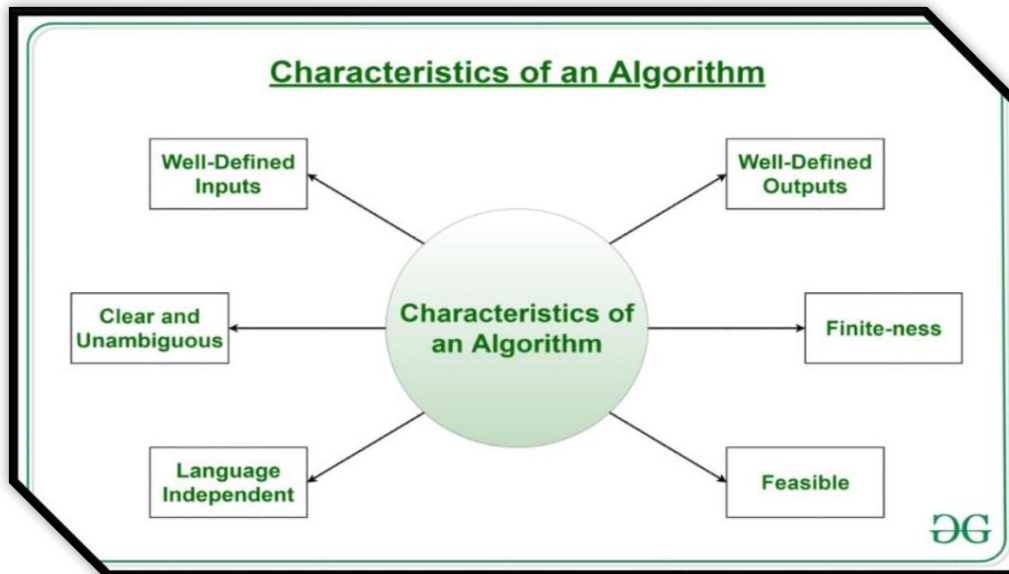
```
[procedure for calculate factorial number]
Step1: Start
Step2: Read the number num
Step3: [Initialize]
       i ← 1, fact ← 1    for loop start here
Step4: Repeat step 4 through 6 until i = n    condition
Step5: fact ← fact * i
Step6: i ← i+1    Increment/Decrement
Step7: Print fact
Step8: Stop
[end of procedure for calculate factorial number]
```

Detail :- (For Exam Content)

- ❖ Algorithm is one of the **best logic development technique**.
- ❖ It is detail **sequence of solving the problem**.
- ❖ In algorithm we can manage input and output as well as any condition.
- ❖ Algorithm is a **step-by-step procedure**, which defines a set of instructions to be executed in a certain order to get the desired output.
- ❖ Algorithms are generally created independent of underlying languages, i.e. an algorithm can be implemented in more than one programming language.

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)



➤ Features of Algorithm :-

- (1) **Input** :- There must be atleast one input.
- (2) **Output** :- There must be atleast one output.
- (3) **Effectiveness** :- Each step of algorithm must be effective.

Example: An algorithm for multiplication of two values.

Step 1: Start
Step 2: Input value -1
Step 3: Input value -2
Step 4: Calculate ans = value -1 * value -2
Step 5: Print ans
Step 6: Stop

Learning outcome (summary)

- Algorithm means "detail sequence of simple steps."
- Algorithm consist set of finite steps.
- Algorithm support step by step representation of program logic.
- Algorithm support following features:
Input

SHREE H.N. SHUKLA COLLEGE OF I.T. & MGMT.

(Affiliated to Saurashtra University & Gujarat Technological University)

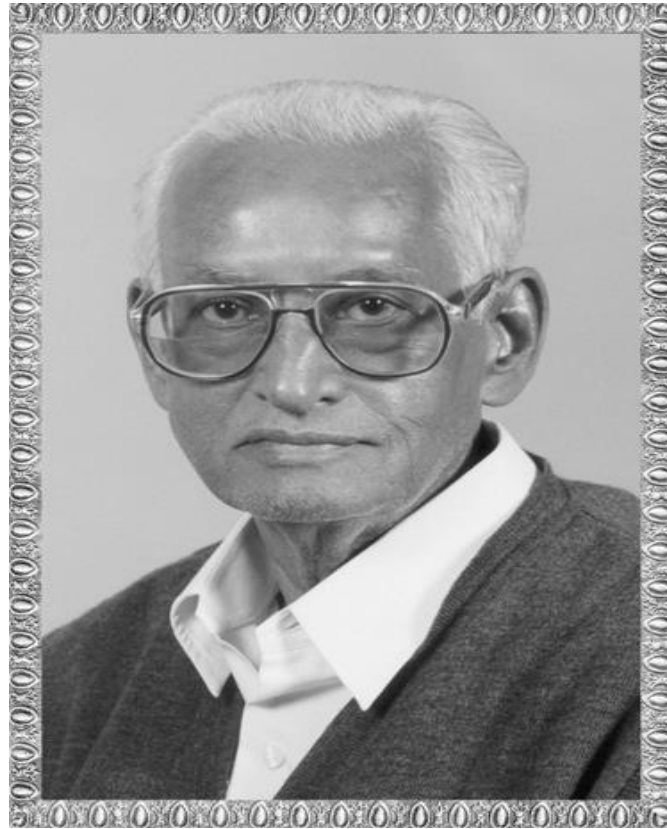
	Output Effectiveness
--	-------------------------

1 Word Question – Answer

1	What is Algorithm?	“sequence of simple steps”
2	Algorithm must_____after finite steps.	terminate

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



Lt. Shree Chimanbhai Shukla

B.C.A./B.Sc.IT/PGDCA-SEMESTER-1

SUB :- PROGRAMMING IN C

Shree H.N.Shukla College
2, Vaishali Nagar,
Nr. Amrapali Railway
Crossing,
Raiya Road, Rajkot.
Ph. (0281)2440478



Shree H.N.Shukla College
3, Vaishali Nagar,
Nr. Amrapali Railway
Crossing,
Raiya Road, Rajkot.
Ph. (0281)2471645

Website: www.hnsgroupofcolleges.org

Email : hnsinfo@hnshukla.com

First PROGRAMMING IN C

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

CS-02: PROBLEM SOLVING METHODOLOGIS AND PROGRAMMING IN C- BCA-1

Sr.No.	Topic	Detail
1	Introduction of C Language	<ul style="list-style-type: none">• Introduction of Computer Languages• Introduction of Programming Concept• Introduction of C Language (History & Overview)• Difference between traditional and modern c.• C character set• C tokens<ul style="list-style-type: none">▪ Keywords▪ Constants▪ Strings▪ Identifiers and variables▪ Operators (all 8 operators)• Hierarchy of operators• Type casting• Data types in c• PRE-PROCESSORS IN C
	Introduction of Logic Development Tools	<ul style="list-style-type: none">• Introduction of Logic.• Necessary Instructions for Developing Logic• Basics of Flow Chart• Dry-run and its Use.• Other Logic development techniques
2	Control Structures	<ul style="list-style-type: none">• Selective control structure<ul style="list-style-type: none">▪ If statements▪ Switch statement• Conditional ternary operator• Iterative (looping) control statements<ul style="list-style-type: none">▪ For loop▪ Do...while loop▪ While loop• Nesting of loops• Jumping statements<ul style="list-style-type: none">▪ Break statement▪ Continue statement▪ Goto statements

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

3	Functions	<ul style="list-style-type: none">• Types of functions• Types of library functions<ul style="list-style-type: none">▪ String Function Strcpy, strncpy, strcat, strncat, strchr, strrchr, strcmp, strncmp, strstr, strspn, strcspn, strlen, strpbrk, strstr, strtok▪ Mathematical Functions Acos, asin, atan, ceil, cos, div, exp, fabs, floor, fmod, log, modf, pow, sin, sqrt▪ Date & Time Functions▪ I/O Formatting Functions printf, scanf, getc, getchar, gets, putc, putchar,
		<ul style="list-style-type: none">▪ Miscellaneous Functions delay, clrscr, clearer, errno, isalnum, isalpha, iscntrl, isdigit, isgraph, islower, isprint, isspace, isupper, isxdigit, toupper, tolower▪ Standard Library functions abs , atof , atol , exit , free , labs , qsort , rand , strtoul , srand▪ Memory Allocation Functions malloc , realloc , calloc• Types of user defined functions• Pointers• Function call by value• Function call by reference• Recursion• Storage classes• Passing and returning values
4	Arrays in C	<ul style="list-style-type: none">• Types of arrays<ul style="list-style-type: none">▪ Single dimensional array▪ Two dimensional array▪ Multi-dimensional array▪ String arrays• Use of Arrays in Programming• Arrays and Matrices
	Structures	<ul style="list-style-type: none">• What is structure• Initializations and declarations• Memory allocation functions• Pointers with structures• Array with structures• Udf with structures• Nested structures• Introduction to union• Difference between Structure & Union

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

5	Pointers in C	<ul style="list-style-type: none">• Introduction of Pointers• Use of pointers in Dynamic Programming• Pointer to Variables• Pointer to Array• Pointer within Array• Pointer To Structure• Pointers within structure• Pointer to Pointer
	File Handling in C	<ul style="list-style-type: none">• Concept of data files• File handling Use of file handling functions fopen(),fclose,fprintf(),fscanf(),getw(),putw(),fseek(), ftell(),rewind(),freopen, remove, rename, feof, ferror, fflush, fgetpos, sprintf, snprintf, vsprintf, vsnprintf, fscanf, vfscanf, setbuf, setvbuf• I/O operations AND Command line arguments

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

CHAPTER-2

➤ CONTROL STRUCTURES

3.1 Selective Control Structures

- ◆ if statements
- ◆ Switch statement

3.2 Conditional (ternary) Operator

3.3 Iterative (looping) control statements

- ◆ For loop
- ◆ Do...while loop
- ◆ while loop

3.4 Nesting of loops

3.5 Jumping statements

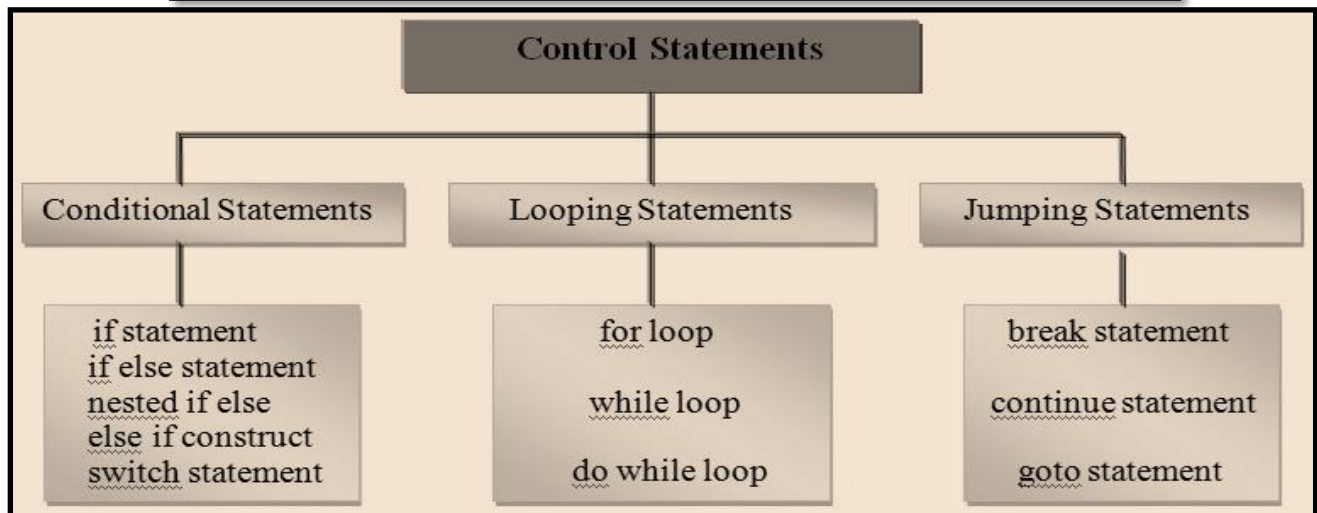
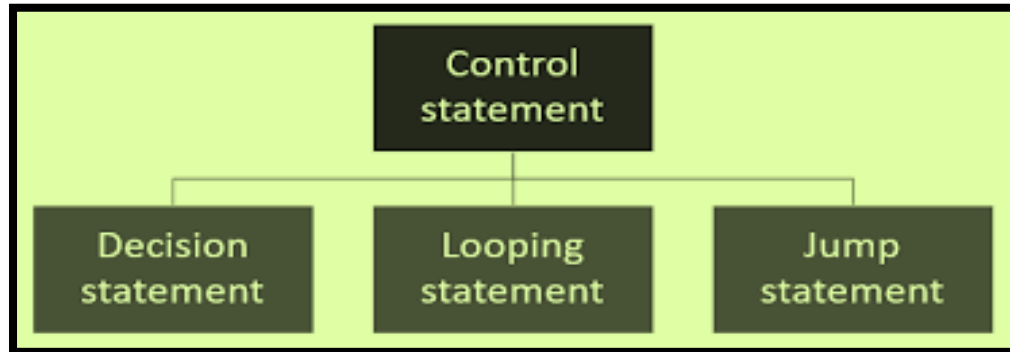
- ◆ Break statement
- ◆ Continue Statement
- ◆ Goto statement

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-1 List out different control structure available in C.

Control Structure :- "It used to indicate Flow of the Program"



◆ If Statements:

If Statement :-

" To check condition & return result "

IF types / Flavors :-

1. Simple If
2. If...Else
3. If...Else if...Else
4. Nested If

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

1. Simple if statement:

Detail :-

❖ C language provide a very powerful control structure that is “if statement.”

- ❖ It is decision statement that use to take some decision.
- ❖ If statement support different flavors or types.

(i) Simple If statement :-

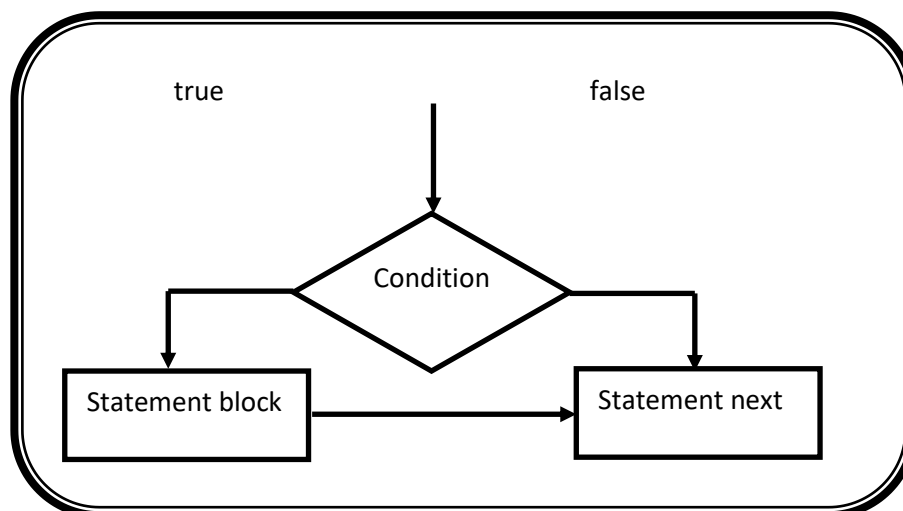
- In this statement first of all condition will be checked.
- If condition become true than the statement following if will be execute.
- If condition become false than condition will be terminated.

The general syntax of simple if statement is follows as:

Syntax:-

```
If(expression)
{
    Statement-block;
}
Statement next;
```

Flowchart :



SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

2. if-else statement:

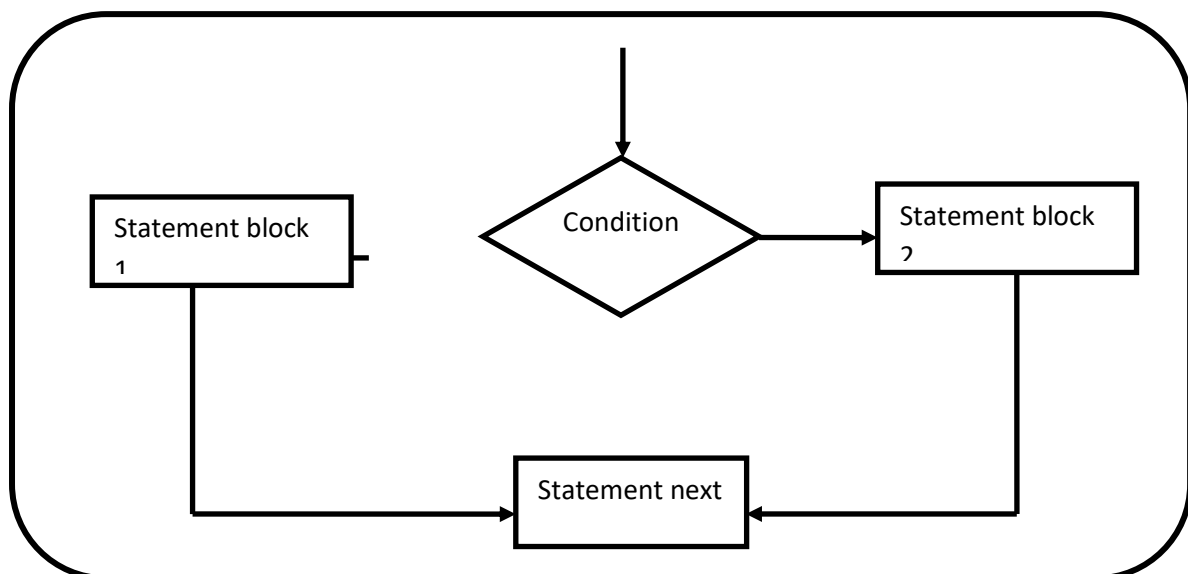
Detail :-

- In this statement first of all , condition will be checked.
- If condition become true than statement following if will be execute.
- If condition become false then statement following else will be execute.

Syntax:-

```
If(expression)
{
    Statement-block;
}
else
{
    Statement-block;
}
Next statement;
```

Flowchart



SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

3. If-else if-else statement:

Detail :-

- In this case, first of all first condition will be checked.
- If first condition become false than second condition will be checked.
- If second condition become false than other condition will be checked.
- If any condition become true than the statement following that 'if' will be executed.
- If no any conditions become true than finally the statement following else will be execute.

Syntax:-

```
If(expression1)
{
    statement block 1
}
Else if(expression2)
{
    statement block 2;
}
Else if(expression3)
{
    statement;
}
Next statements;
```

Example:

```
#include<stdio.h>
#include<conio.h>

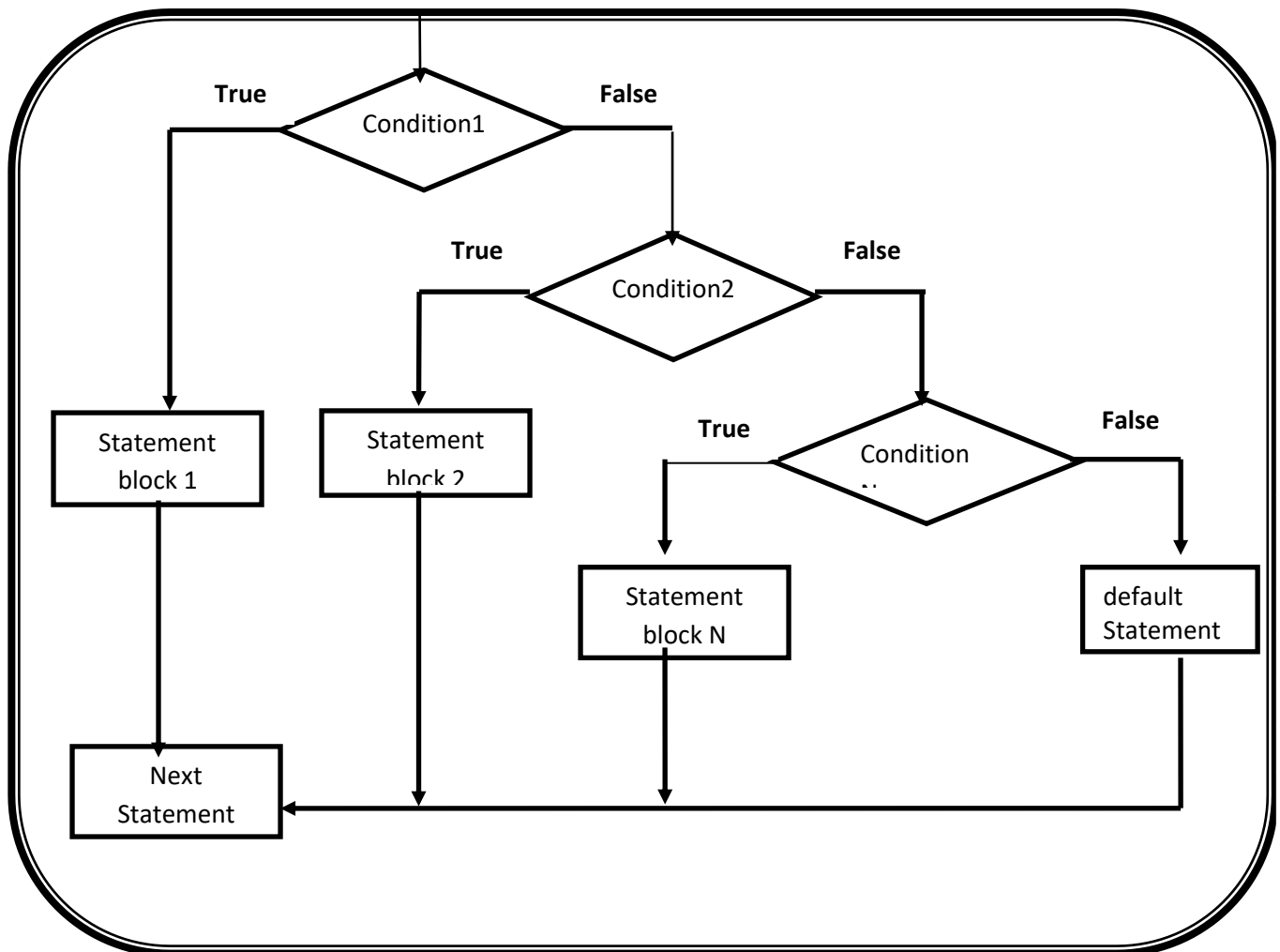
void main()
{
    int no;
    clrscr();
    printf("\n enter the number");
    scanf("%d",&no);
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
if(no<0)
    printf("\n number is negative");
else if(no==0)
    printf("\n number is zero");
else
    printf("\n number is positive");
getch();
}
```

Flowchart:



Q-2 Explain Nested If with example.

Detail :-

- ❖ In this statement , user can create one if statement inside other if statement , that is called nested if.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- ❖ It means that we create outside if than it is called “outer if” and if we create inside if than it is called “inner if”.

Syntax:-

```
If(expression1)
{
    if(expression2){
        statement block 1
    }
    Else {
        statement block 2; }
}
else
{
    statement block 3;
}
Next statements;
```

Example:-

```
#include<stdio.h>
#include<conio.h>

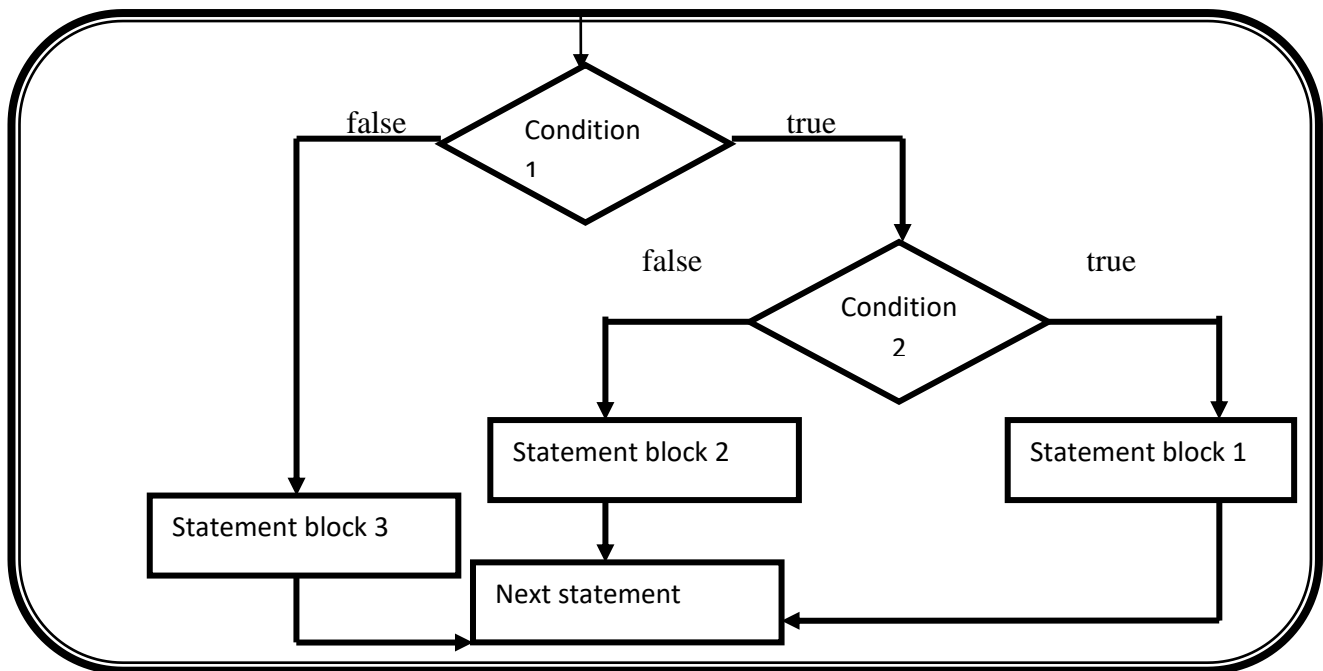
void main()
{
    int n1,n2,n3;
    clrscr();
    printf("\n enter the n1,n2,n3");
    scanf("%d",&n1,&n2,&n3);
    if(n1>n2)
    {
        If(n1>n3)
            printf("\n n1 is max");
        else
            printf("\n n3 is max");
    }
    else
    {
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
        if(n2>n3)
        printf("\n n2 is max");
        else
        printf("\n n3 is max");
    }
    getch();
}
```

Flowchart:



1 Word Question – Answer


SR.NO.	QUESTION	ANSWER
1	How Many Control Structures available in C ? Give name.	5 (two) If Statement Switch Statement
2	If any condition become false ,statement following _____ will be execute.	Else
3	In nested if first of all _____ condition will be checked.	Outer
4	How many flavors/types of If statement.Give name.	4(four) Simple if

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

		If...else If...else..if...else Nested if
5	Which indicate easy way to represent multiple conditions at the same time?	Else...if

Q-3 Explain Switch_____case statement with example.

SWITCH CASE :- 

" MULTI WAY DECISION STATEMENT "

CASE :- WE CAN CREATE ANY NUMBER OF CASE VALUES INSIDE SINGLE SWITCH STATEMENT.

DEFAULT :- IT EXECUTE WHEN NO ANY CASE VALUE BECOME TRUE

Detail :-

- ❖ The switch case statement is also known as multi condition statement.
- ❖ It provide facility to check the condition.
- ❖ Switch case make it easy to give the choice and take the decision.
- ❖ It also support multiple case values. But at a time only one case value Become true.
- ❖ If any case value match with the condition than finally statement following that case will be execute.
- ❖ If no any case value match with the condition then finally the statement following default will be execute.

Break :-

- ❖ This statement is used to end or stop process of particular case.
- ❖ Without break the programme will continue to the next case.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Syntax:-

```
Switch(Condition)
{
    Case Expression 1:
        Statement block1;
        break;

    Case Expression 2:
        Statement block2;
        break;
    .....
    .....

    default:
        Statement block;
}
```

Example:

```
#include<stdio.h>
#include<conio.h>

void main()
{
    int color=1;
    clrscr();
    printf("\n enter the color 1. RED 2. GREEN 3. BLUE");
    scanf("%d",&color);
    switch(color)
    {
        case 1:
            printf("\n RED");
            break;
        case 2:
            printf("\n GREEN");
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

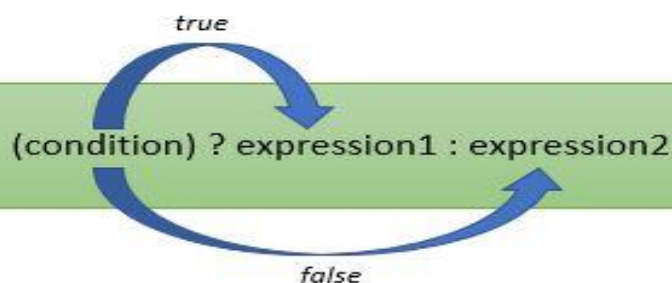
```
break;
case 3:
    printf("\n BLUE");
    break;
default:
    printf("\n NO COLOR");
    }
getch();
}
```

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Switch represent _____	Multiway Decision Statement
2	If no any case value match with condition then statement following _____ will execute.	Default
3	_____ statement is used to terminate particular case.	Break
4	In switch case statement , any case value will be followed by _____ sign.	: (Colon)
5	Write down syntax to represent switch statement	Switch(Expression)

Q-4 Explain Conditional Operator with example.

- ◆ **CONDITIONAL OPERATOR IS SIMILAR TO "IF – ELSE " STATEMENT.**
- ◆ **CONDITIONAL OPERATOR ALSO KNOWN AS "TERNARY OPERATOR"**



SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :-

- ❖ The conditional operator is also known as ternary operator.
- ❖ The conditional statement is used to check the condition.
- ❖ The conditional operator support mainly 3 arguments.
 - (i)Condition
 - (ii)True Part
 - (iii)False Part
- ❖ In C language there are two conditional operators.
 - (1) ? (Question mark) --→ True part
 - (2) : (Colon) --→ False part

- ❖ The general form of the ternary operator is as follow:

Syntax:

(Condition)? Statement-1 : statement -2;

- ❖ First of all condition will be checked.
- ❖ If the condition become true then the statement followed by ? is executed
- ❖ If condition become false than the statement followed by : is executed.

Example :

(a>b)? printf("a is maximum"): printf("b is maximum");

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Conditional operator also known as _____	Ternary operator
2	Syntax for conditional operator is _____	(condition) ?<statement -1> : <statement -2>
3	In conditional operator _____ is used to indicate false part.	: (Colon)
4	Conditional operator is similar to _____ statement.	If...else

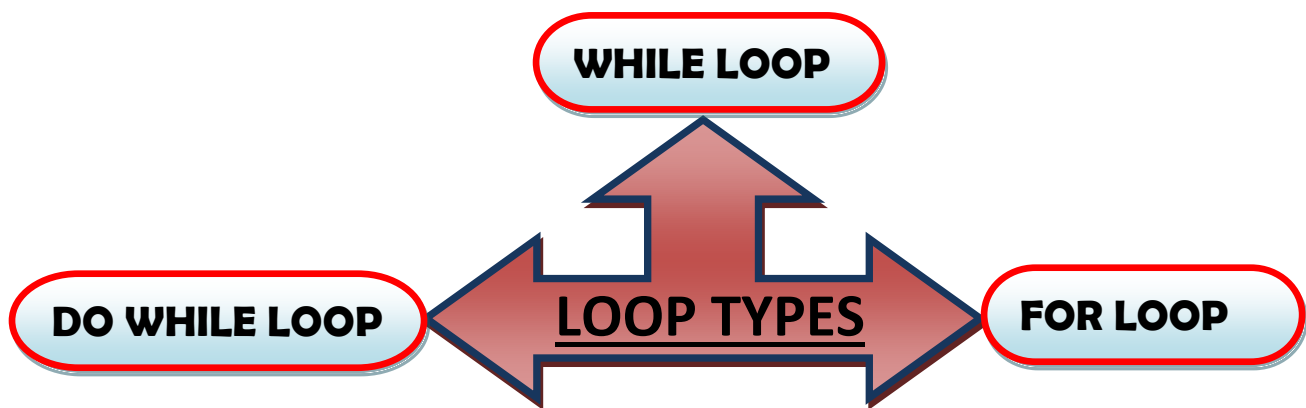
SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

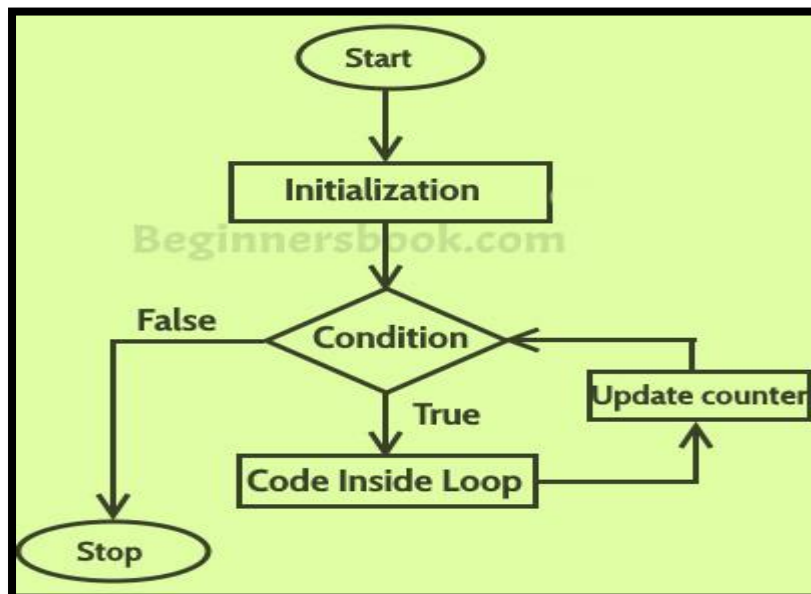
Q-5 Iterative (looping) control statements.

Loop Means “to execute block of code repeatedly”.

Looping statements also called “iterative statements”.



LOOP EXECUTION :-



Detail :-

- ❖ Loop is used to execute the statements for number of times .

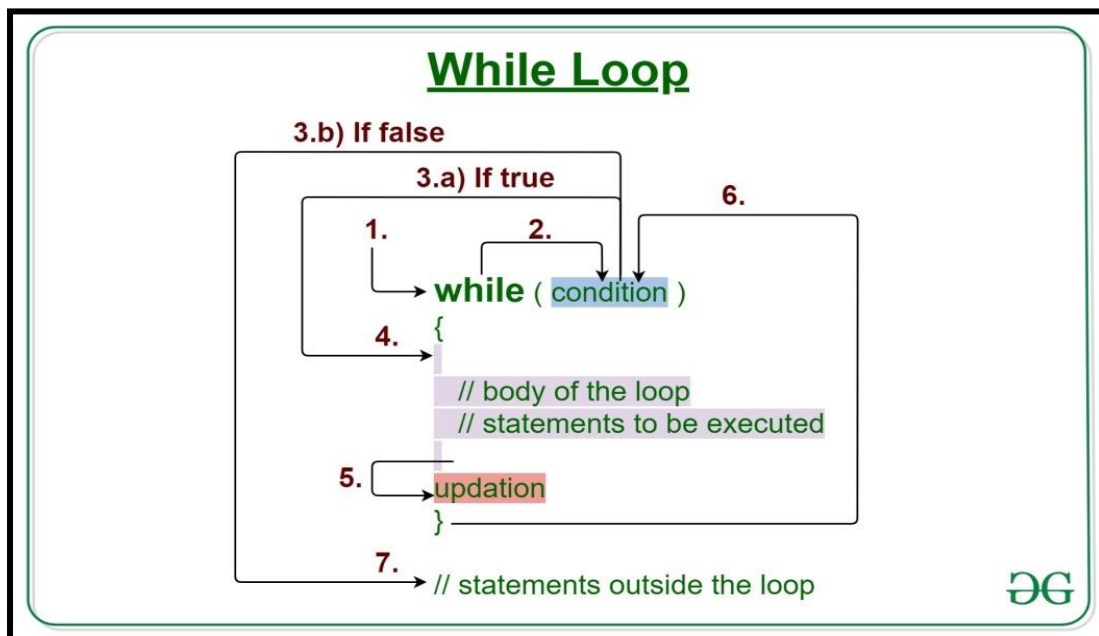
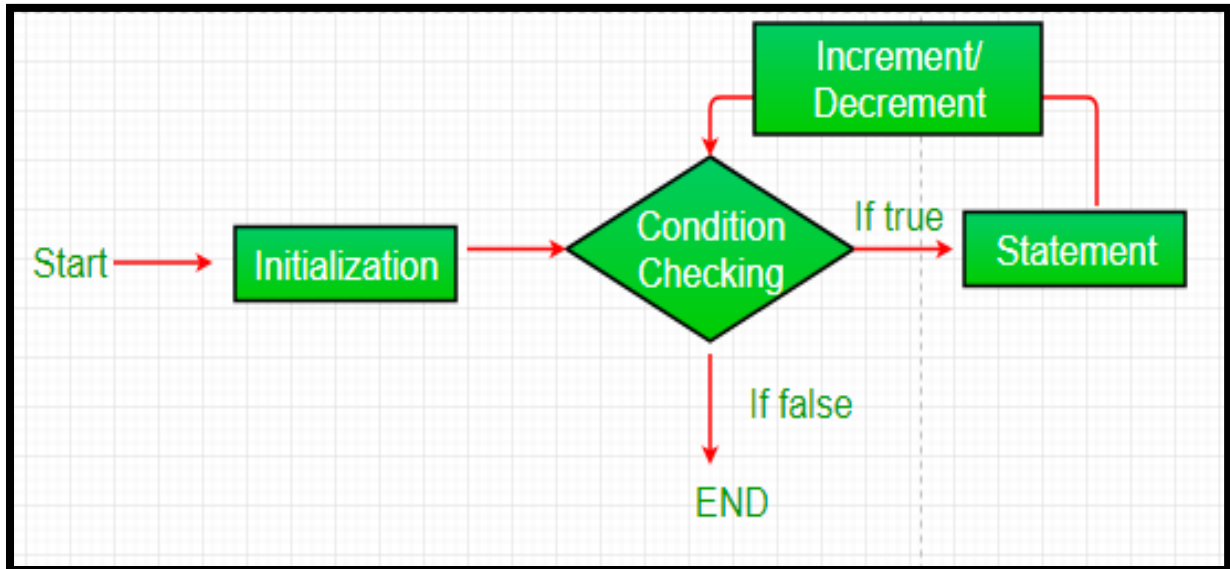
SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- ❖ Loop provide easy way to repeat the task and reduce the code.
- ❖ A loop is a group of statements that are executed until some condition satisfied.
- ❖ In C language we have to provide 3 criterias
 - (i) Starting value
 - (ii) Condition
 - (iii) Increment/Decrement
- ❖ In C language there are 3 types of looping statements.
 - (1) While loop [Entry Controlled loop]
 - (2) Do..while loop [Exit Controlled loop]
 - (3) For loop [Entry Controlled loop]
- ❖ A loop declaration and execution can be done in following ways.
 - Check condition to start a loop
 - Initialize loop with declaring a variable.
 - Executing statements inside loop.
 - Increment or decrement of value of a variable.
- ◆ **While loop:**
 - While loop is also known as **Entry controlled loop.**
 - While loop is entry controlled loop because the condition will be checked first then statement will be execute.
 - In this loop , if condition become true then statement will be execute otherwise loop will be terminated.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



Syntax:

```
while(condition)
{
    statements;
    increment/decrement;
}
```

Example:

```
int a;
```

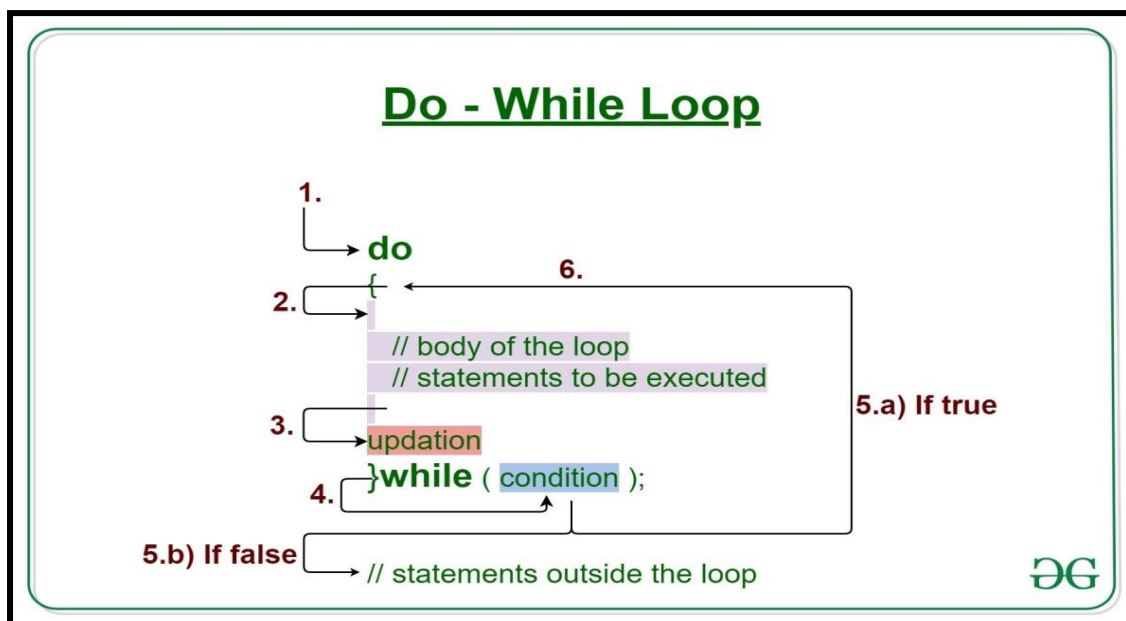
SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
clrscr();
a=1;
while(a<=5)
{
    printf("\n TechnoExam");
    a+=1    // i.e. a = a + 1
}
```

◆ Do..While loop:

- Do..While loop is also known as **Exit controlled loop**.
- Do..While loop is exit controlled loop because the statement will be execute first and condition will be checked at last.
- In this loop , if condition become false then atleast once statement



will be execute because condition will be checked at last.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Syntax:

```
do
{
    statements;
    (increment/decrement);

}while(condition);
```

Example:

```
int i=1;

clrscr();
do
{
    printf("\n %d",i);
    i++;
}while(i<=10);
```

◆ For loop:

- For loop is also known as **Entry controlled loop.**
- It is also known as faster loop in C language.
- In this loop we can give starting value , condition and increment /decrement in single line.

Syntax:

```
for(Initialization; test-condition; increment /decrement)
{
    statements;
}
```

Example:

```
int a;
```

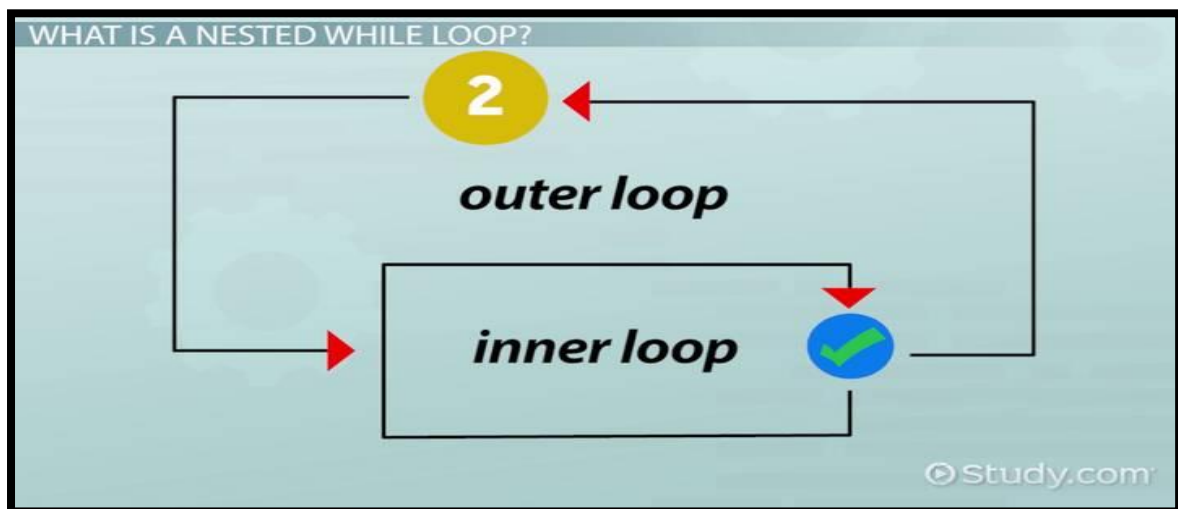

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
clrscr();  
for(i=0; i<5; i++)  
{  
    printf("\n\t Hello"); // 5 times  
}
```

Q-6 Explain Nested Loop in C Language.

Nested loops are the loops which contain another looping statement in a single loop. }



- ❖ Nested loop provide facility to create one loop inside other loop.
- ❖ You can create number of looping statement inside single loop.
- ❖ The first loop is known as outer loop and the loop inside first loop is called inner loop.
- ❖ The outer loop is used for counting rows and the inner loop is used for counting columns.

Syntax:

```
for(Initialization; test-condition; increment /decrement)  
{  
    For(Initialization;test-condition;increment/decrement)  
    {  
        statements;    }  
}
```

Example:

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
Int i ,j;  
for(i=1; i<=5; i++)  
{  
    For(j=1;j<=i;j++)  
    {  
        Printf("* ");  
    }  
}
```

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Nested loop means_____	Loop within Loop
2	The first loop in nested loop is called _____	Outer loop
3	The loop inside the first loop is called _____	Inner loop

Q-7 Explain Break , continue & Goto Statements.

OR

Explain Jumping Statements in detail.

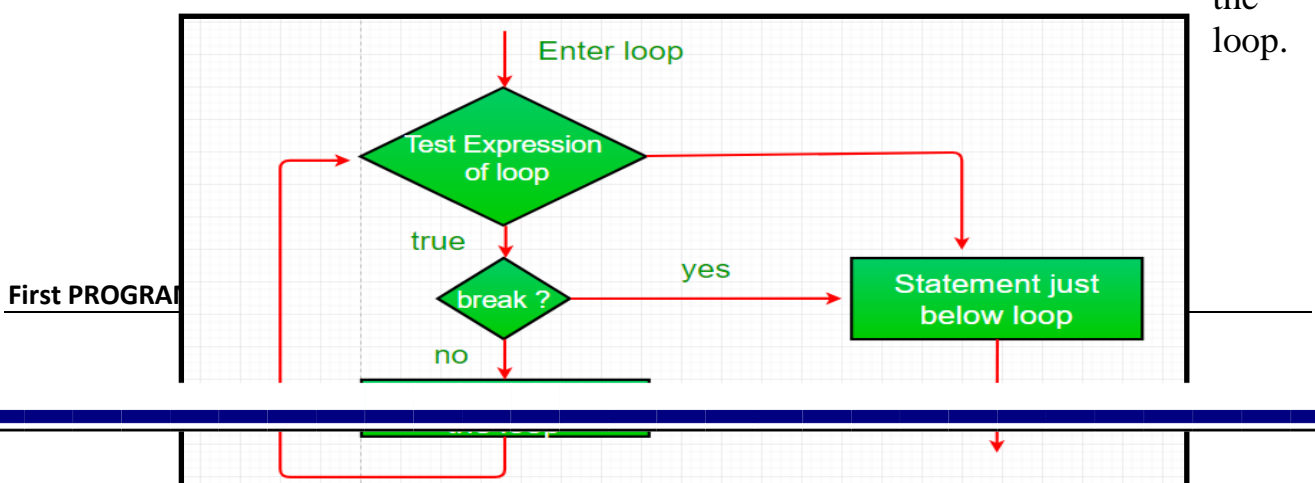
Detail :-

- ❖ The jumping statements are used to transfer the program control.
- ❖ There are following jumping statements available in C.
 - Break
 - Continue
 - Goto

◆ break Statement:

- Sometimes it is necessary to stop execution of the program.
- The break statement is unconditional statement.
- Break statement is used for terminating the loop or jump out from

the loop.



SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Syntax:

Break;

Example:

```
int i=1;
clrscr();
While (i<=10)
{
    if(i==5)
    {
        break;
        printf("%d",i); // 5 times only
    }
}
```

◆ continue Statement:

- The continue statement is used to pass the control to next iteration(loop).
- It is also unconditional statement.
- Continue does not terminate the loop but only skip the statement followed by if.

Syntax:

Continue;

Example:

```
Int I;
Clrscr();
For(i=1;i<=10;i++)
{
    If(i>4)
```

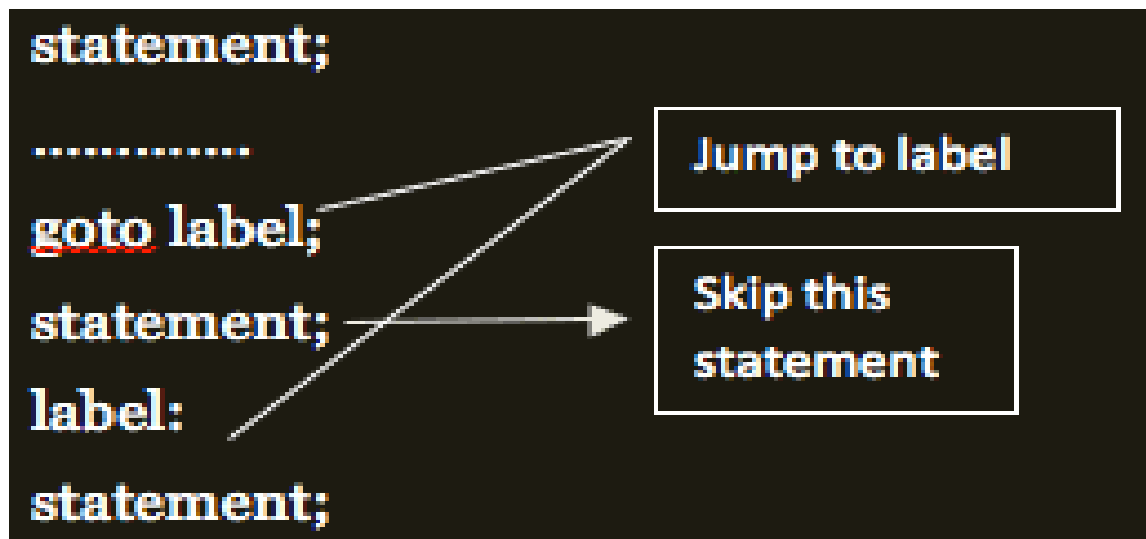
SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
    {  
        Continue;  
    }  
Else  
{  
    Printf("*\n");  
}  
}
```

◆ Goto Statement:

- Goto statement is also unconditional statement.
- Goto is used to transfer the control (cursor) from one point to another point in the program.
- The goto statement is marked by label.
- The label is variable name followed by colon (:).



Syntax:

Goto <variable name>;

Example:

```
Int i=10,j=20;  
Clrscr();  
If(i>j)
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
{
    goto l;
}
Else
{
    goto j;
}
l: printf("l is max");
j: printf("j is max");

    Continue;
}
Else
{
    Printf("*\n");
}
}
```

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	How many jumping statements available in c?	3(Break ,Continue,Goto)
2	_____ statement is used to terminate from the loop.	Break
3	_____ statement is used to pass control to the next iteration.	Continue
4	_____ statement is used to jump from one point to another point.	Goto

Q-8 Give difference between While loop & Do .. While loop.

While Loop	Do..While Loop
1)While loop is known as Entry Controlled Loop.	1)Do..While loop is known as Exit controlled Loop.
2) In While loop if condition become false then no any output will be given.	2)n Do..While loop if the condition become true than atleast one loop will

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

	be execute & output will be display.
3)There is no terminating semicolon(;) at the end of loop.	3)There is terminating semicolon(;) at the end of do..while loop.
4)Syntax : While (<condition> { <Statements> }	4)Syntax : do { <Statements> } While (<condition>;

Q-9 Give difference between While loop & For loop.

While Loop	For Loop
1)While loop is bit slower than for loop.	1)For loop is faster than while loop.
2) While loop is not suitable for simple initialization.	2)For loop is more suitable when there is simple initialization.
3)Syntax : While (<condition> { <Statements> }	3)Syntax : For(<Initialization>;<Condition>; <Inc./Dec.> { <Statements> }

Q-10 Give difference between Break & Continue.

While Loop	For Loop
1)Break is used to terminate the block & get the control out of the loop.	1)Continue is used to get the control to the next iteration of the loop.
2) Break statement can be used in both switch case & loop.	2)Continue statement can be only used with looping statement.
3)Syntax : <break>;	3)Syntax : <continue>;

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Looping statements are also known as _____ statements	Iterative
2	_____ Loop is known as Entry	While

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

	Control Loop.	
3	_____ Loop is known as Exit control Loop.	Do While
4	In Do while Loop condition will be terminated with ____.	Semi Colon(;
5	_____ Loop is known as faster loop in c.	For

SUMMARY QUESTIONS OF CHAPTER -2

ONE - MARK QUESTIONS

- ❖ What is Loop?
- ❖ What is Conditional Statement?
- ❖ Which Loop is known as Entry controlled Loop?

First PROGRAMMING IN C

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- ❖ Which Loop is known as Exit controlled Loop?
- ❖ Which Statement is used to Stop the execution of Program or Loop from middle of Iteration?
- ❖ If no any case value match with given condition then the statement following _____ will be execute automatically in Switch Case.
- ❖ Which Statement is also Known as Jumping Statement?
- ❖ How many cases you can create within single Switch Statement?
- ❖ List out different UnConditional Statements in C Language.
- ❖ Any Lable of GoTo Statement can be followed by _____.

TWO - MARK QUESTIONS

- ❖ Explain If.....Else Statement with example.
- ❖ Write down Syntax for ifElse if...Else Statement.
- ❖ Write down Syntax of For Loop.
- ❖ Write down Syntax for Switch Case Statement.
- ❖ Explain GoTo Statement in brief.
- ❖ Write note on Conditional(Ternary) Operator.

THREE - MARK QUESTIONS

- ❖ Explain Nested If with Suitable Example.
- ❖ Give difference between While and Do While.
- ❖ Give difference between Break and Continue.
- ❖ Explain If...Else if Ladders with Example.

FIVE - MARK QUESTIONS

- ❖ List out different Looping Statements(Iterative Statements) and Explain any two with Example.
- ❖ Explain Switch Case with Example.
- ❖ List out different Conditional Statements(Control Structure) and Explain any one with Example.
- ❖ Write note on Break , continue and GoTo.

CHAPTER-3

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

➤ FUNCTIONS

3.1 Types of functions

3.2 Types of library functions

- ◆ string functions

strcpy(), strncpy(), strcat(), strncat(), strchr(), strrchr(), strcmp(), strncmp(), strstr(), strspn(), strcspn(), strlen(), strpbrk(), strstr(), strtok()

- ◆ Mathematical Functions

Acosh(), asinh(), atanh(), ceil(), cosh(), div(), exp(), fabs(), floor(), fmod(), log(), modf(), pow(), sinh(), sqrt()

- ◆ Date and time functions

clock(), difftime(), mktime(), time(), asctime(), ctime(), gmtime(), localtime(), strftime()

- ◆ I/O Formatting Functions

printf(), scanf(), getc(), getchar(), gets(), putchar(), puts(), ungetc()

- ◆ Miscellaneous Functions

delay(), clrscr(), clrerr(), errno(), isalnum(), isalpha(), iscntrl(), isdigit(), isgraph(), islower(), isprint(), isspace(), isupper(), isxdigit(), toupper(), tolower().

- ◆ Standard Library functions

abs(), atof(), atol(), exit(), free(), labs(), qsort(), rand(), strtoul(), srand()

- ◆ Memory Allocation functions

malloc(), realloc(), calloc()

3.3 Types of user defined functions

3.4 Pointers

3.5 Function call by value

3.6 Function call by reference

3.7 Recursion

3.8 Storage class

3.9 Passing and returning values.

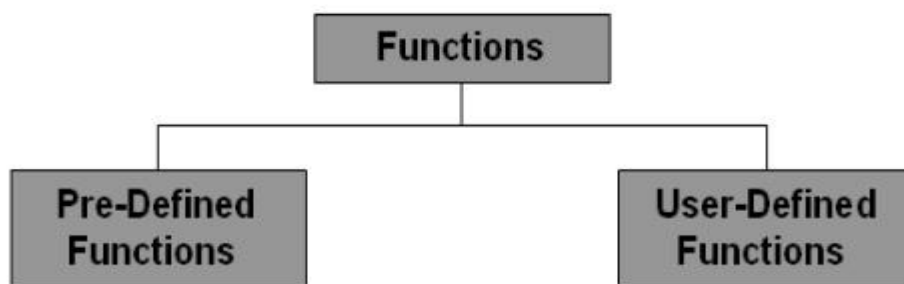
SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-1 Explain Types of functions in C.

3.Types of functions

- ▶ C functions can be classified into two categories
 - Library functions
 - User-defined functions



Detail :-

A function is a self-contained block of statements that perform a task of some kind. C functions can be classified into two categories:

1. Library Functions.
2. User-defined functions

1. Library Functions:

C provides library functions for performing some operations. These functions are present in the c library and they are predefined for example. Sqrt() is a mathematical library function which is used for finding the square root of any number The function scanf and printf() are input and output library function similarly we have strcmp() and strlen() for string manipulations.

To use a library function we have to include some header file using the preprocessor directive #include. Example: to use input and output function like printf() and scanf() we have to include stdio.h, for math

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

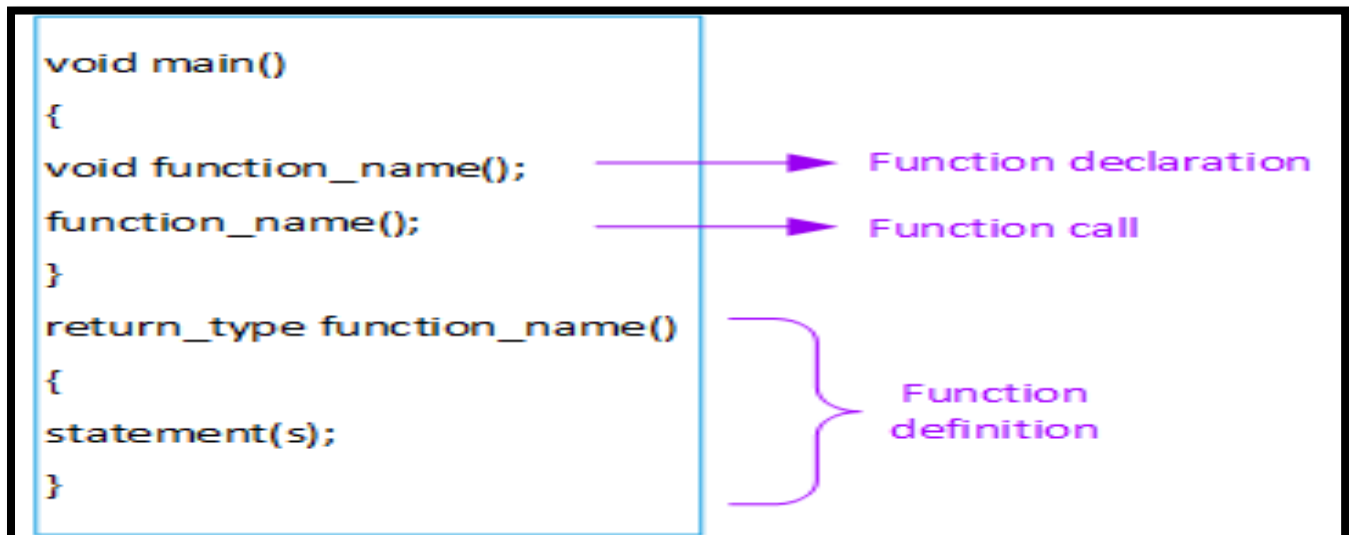
library function we have to include math.h for string library string.h should be included.

2. User-defined Functions:

User defined function (UDF) is an independent program module with group of statements and that function can be called and use whenever it is required. To make programming easier, we break larger program into smaller sub-programs to perform a specific well defined task. These sub-programs are called functions. To create and use UDF following 3 things should be kept in mind:

- ◆ Function Declaration
- ◆ Function Definition
- ◆ Function Call

Detail :-



SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
1 int a = 10, b = 5, c;  
2  
3 int product(int x, int y);  
4  
5 int main(void)  
6 {  
7     c = product(a,b);  
8     printf("%i\n",c);  
9     return 0;  
10 }  
11  
12  
13  
14 int product(int x, int y)  
15 {  
16     return (x * y);  
17 }
```

Function Prototype - int is the return type and int x and int y are the function arguments

Main Function - int is always the return type and there are no arguments, hence the (void). Curly braces { } mark the start and end of the main function

Function call - product(a,b); a and b are global variables the function is passed. Here the values returned by the function are assigned to the variable c

Function Definition - contains the function statement return(x * y); the function returns x times y to the main function where it was called. Curly braces { } mark the start and end of the function

▪ Function Declaration (Function Prototype):

The calling program should declare the function before it is used. This is known as function declaration or function prototype. A function declaration contains four elements

1. Function type
2. Function name
3. parameters list
4. terminating semicolon

Syntax:

function_type function_name(parameter list);

Example :

int add(int, int);

Function declaration is similar to function header except the terminating semicolon

▪ Function Definition:

The function definition consists of the whole description and code of a function. It tells that what the function is doing and what is

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

the input output for that. A function is called by simply writing the name of the function followed by the argument list inside the parenthesis. There are six elements of function definition:

1. Function type
2. Function name
3. Parameters list (arguments)
4. Local variable declaration
5. Function statements
6. return statement

Syntax:

```
function_type function_name(parameter list)
{
    local variable declaration;
    function statement 1;
    function statement 2;
    .
    .
    .
    return statement;
}
```

Example :

```
int add(int x, int y)
{
    int z;
    z = x + y;
    return(z);
}
```

The elements of function definition are categorized in two parts:

1. Function header
2. Function body

1. Function header: A function header contains three elements of function definition with function type, function name and parameter list. In the above example:

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

`int add(int x,int y)` is known as function header.

2. Function body: A function body contains last three elements of function definition with local variable declaration, function statements and a return statement. In the above example:

```
int z;
z=x+y;
return z; } Function body
```

- **Function call:**

A function can be called by using the function name and list of actual parameters.

Example :

```
main()
{
    c=add(10,5);//function call
    printf("%d",c);
}
```

In the above example, when a function calls statement encounter in program then the control is transferred to the function `add()`. After this `add` function is executed line by line and the answer is return to the function call using the return statement.

NOTE:

Actual Parameters: The parameters which are used in function call is known as actual parameter. Actual parameters send the copy of the data of called function.

Formal Parameters: The parameters which are used in function definition is known as formal parameters. Formal parameters will receive the data, sent by calling function.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-2 Explain Types of Library functions:

- String Functions:

Detail :-

- C supports a wide range of functions that manipulate null-terminated strings:

S.no	String functions	Description
1	<code>strcat ()</code>	Concatenates str2 at the end of str1.
2	<code>strncat ()</code>	appends a portion of string to another
3	<code>strcpy ()</code>	Copies str2 into str1
4	<code>strncpy ()</code>	copies given number of characters of one string to another
5	<code>strlen ()</code>	gives the length of str1.
6	<code>strcmp ()</code>	Returns 0 if str1 is same as str2. Returns <0 if str1 < str2. Returns >0 if str1 > str2.
7	<code>strcmpi ()</code>	Same as strcmp() function. But, this function negotiates case. "A" and "a" are treated as same.
8	<code>strchr ()</code>	Returns pointer to first occurrence of char in str1.
9	<code>strrchr ()</code>	last occurrence of given character in a string is found
10	<code>strstr ()</code>	Returns pointer to first occurrence of str2 in str1.
11	<code>strrstr ()</code>	Returns pointer to last occurrence of str2 in str1.
12	<code>strdup ()</code>	duplicates the string
13	<code>strlwr ()</code>	converts string to lowercase
14	<code>strupr ()</code>	converts string to uppercase
15	<code>strrev ()</code>	reverses the given string
16	<code>strset ()</code>	sets all character in a string to given character
17	<code>strnset ()</code>	It sets the portion of characters in a string to given character
18	<code>strtok ()</code>	tokenizing given string using delimiter

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



▪ STRING FUNCTIONS :-

Detail :-

(1) Strlen() :-

- This function is used to return the length of any given string.
- The length of the string is decided by the terminating NULL character ('\0').

Syntax :- Strlen (<string>)

Example :- Char str [10] = "hello hi";
Clrscr();
Printf("%d",strlen(str));

(2) Strcpy() :-

- This function is used to copy the content of one string into another string.
- **Syntax :- Strcpy (<destination str> , <source str>)**

Example :- Char str [] = "hello friends";
Char str2[40];
Clrscr();
Strcpy(str2,str1);
Printf("%s", str2);

(3) Strncpy() :-

- This function is used to copy the first n characters of a given string into another string.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- **Syntax :- Strcpy (<destination str> , <source str>,<size upto N>)**

Example :- Char str [] = "hello friends";

```
Char str2[40];  
Clrscr();  
Strncpy(str2,str1,5);  
Printf("%s", str2);
```

(4) Strchr() :-

- This function is used to search first occurrence of specified character in given string and return rest of the string.

- **Syntax :- Strchr (<string>,<Character to be search>)**

Example :- Char str [30] = "hello friends";

```
Clrscr();  
Printf("%s", strchr(str,'f');
```

(5) Strrchr() :-

- This function is used to search last occurrence of specified character in given string and return rest of the string.

- **Syntax :- Strrchr (<string>,<Character to be search>)**

Example :- Char str1 [30] = "This is just a string";

```
Clrscr();  
Printf("%s", strrchr(str1,'s');
```

(6) Strcmp() :-

- This function is used to compare two strings character by character using ASCII value and return integer value.
- If First string is greater than (>) the second string than it will return positive value.
- If first string is less than(<) the second string than it will return negative value.
- If both strings are equal than it will return zero (0).

- **Syntax :- Strcmp (<str 1>,<str 2>)**

Example :- Char str1 [20] = "ABC";

```
Char str2[20] = "abc";  
Int c;  
Clrscr();  
C = strcmp(str1,str2);  
Printf("%d",c);
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

(7)Strncmp() :-

- This function is used to compare number character of string one into number character of string two.
- This function start comparing first character of each string.
- **Syntax :-** int Strncmp (<str 1>,<str 2>,<size upto n>)

```
Example :- Char str1 [20] = "R2A6a";  
Char str2[20] = "R2A5a";  
Int r;  
Clrscr();  
r = strncmp(str1,str2,4);  
Printf("%d",r);
```

(8)Strcat() :-

- This function is used to append (join) one string with another string.
- The character of second string will be appended to the first string.
- **Syntax :-** strcat (<str 1>,<str 2>)

```
Example :- Char str1 [20] = "str";  
Char str2[20] = "cat";  
Clrscr();  
Printf("%s",strcat(a,b));
```

(9)Strncat() :-

- This function is used to append (join) specified given number of characters of destination string to source string.
- **Syntax :-** strncat (<destination string>,<source string>,<upto N>)

```
Example :- Char str1 [20] = "to be";  
Char str2[20] = "or not to be";  
Clrscr();  
Printf("%s",strncat(str1,str2,6));
```

(9)Strncat() :-

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- This function is used to append (join) specified given number of characters of destination string to source string.
- **Syntax :-** `strncat (<destination string>,<source string>,<upto N>)`
Example :- `Char str1 [20] = "to be";`
`Char str2[20] = "or not to be";`
`Clrscr();`
`Printf("%s",strncat(str1,str2,6));`

(10)Strspn() :-

- This function is used to get length of substring of string 1 that is made up of only those characters contained in string 2.
- **Syntax :-** `strspn (<string 1> , <string 2>)`
Example :- `Char str1 [20] = "I am 123";`
`Char str2[20] = "I am sorry123";`
`Int i;`
`Clrscr();`
`i=strspn(str1,str2);`
`Printf("length of matched character is =%d" , i);`

(11)Strcspn() :-

- This function is used to scan the main string for the given string and return number of characters from beginning till the first matched character is found.
- **Syntax :-** `strcspn (<string > , <string to be search>)`
Example :- `Char str1 [20] = "ABCDEFG0123";`
`Char str2[20] = "0123";`
`Int i;`
`Clrscr();`
`i=strcspn(str1,str2);`
`Printf("First Matched character is at =%d" , i);`

(12)Strstr() :-

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- This function is used to find out first occurrence of substring str 2 in str 1 and returns remaining string.
- **Syntax :-** strstr (<str 1> , <str 2>)
Example :- Char str1 [20] = "hi how r u";
Char str2[20] = "how";
Clrscr();
Printf("substring is = %s",strstr(str1 ,str2));

(12)Strtok() :-

- This function is used to break the string into series of tokens using given delimiter.
- **Syntax :-** strtok (<str 1> , <delimiter>)
Example :- Char str1 [20] = "www.hns.com";
Char str2[20] = ".";
Char *t;
T = strtok(str1,str2);
While (t!=NULL)
Clrscr();
{
Printf("%s\n",t);
T = strtok(str1,str2);
}

(12)Strpbrk() :-

- This function is used to find out first character in first string that match any character in second string.
- **Syntax :-** strpbrk (<str 1> , <delimiter>)
Example :- Char str1 [20] = " I love kite";
Char str2[20] = "kite";
Char *t;
T = strpbrk(str1,str2);
If(t!=NULL)
{

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
Printf("matched character is =%c\n",*t);  
Else  
Printf("character not found");  
}
```

- **MATHS FUNCTIONS:**

Header File: :- **<math.h>**

S.No	Function & Purpose
1	double cos(double); This function takes an angle (as a double) and returns the cosine.
2	double sin(double); This function takes an angle (as a double) and returns the sine.
3	double tan(double); This function takes an angle (as a double) and returns the tangent.
4	double log(double); This function takes a number and returns the natural log of that number.
5	double pow(double, double); The first is a number you wish to raise and the second is the power you wish to raise it t
6	double hypot(double, double); If you pass this function the length of two sides of a right triangle, it will return you the length of the hypotenuse.
7	double sqrt(double); You pass this function a number and it gives you the square root.
8	int abs(int); This function returns the absolute value of an integer that is passed to it.
9	double fabs(double); This function returns the absolute value of any decimal number passed to it.
	double floor(double); Finds the integer which is less than or equal to the argument passed to it.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

▪ MATHS FUNCTIONS :-

Detail :-

(1) abs() :-

▪ This functions is used to return absolute value of any given number.

Syntax :- abs (<number>)

Example :- abs (-90)

(2) sqrt() :-

▪ This functions is used to return square root of any given number.

Syntax :- sqrt (<number>)

Example :- sqrt (100)

(3) ceil() :-

▪ This functions is used to return or round up given value upto near by maximum value of given number.

Syntax :- ceil (<number>)

Example :- ceil(-14.55)

(4) floor() :-

▪ This functions is used to return or round up given value upto near by minimum value fo given number.

Syntax :- floor (<number>)

Example :- floor(-14.55)

(5) div() :-

▪ This functions is used to return reminder of any given number.

Syntax :- div (<number>,<divisible by>)

Example :- div(10,3)

(6) exp() :-

▪ This functions is used to return exponent value of any given number in which e raised to value of n.

Syntax :- exp (<number>)

Example :- exp(5)

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

(7)pow() :-

▪ This functions is used to return the value calculated by given number and its power.

Syntax :- pow (<number>,<raised to>)

Example :- pow(2,3)

(8)log() :-

▪ This functions is used to return natural logarithm of any given number.

Syntax :- log(<number>)

Example :- log(3.123)

(9)modf() :-

▪ This functions is used to return integer part and floating point part of any given number.

Syntax :- modf(<number>,<int.part>)

Example :- modf(3.00000,3)

(10)fabs() :-

▪ This functions is used to return absolute value of floating point number.

Syntax :- fabs(<number>)

Example :- fabs(-23.40)

(11)sin() :-

▪ This functions is used to calculate sine value of given number.

Syntax :- sin(<number>)

Example :- result = sine(param number * pi)/180;

(12)cos() :-

▪ This functions is used to calculate cosine value of any given number.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- It return the value in the range from -1 to +1;

Syntax :- `cos(<number>)`

Example :- `result = cos(number * pi)/180;`

(13)asin() :- [arc = curve]

- Asin() is known as arc sin function.
- It used to calculate value of arc sine in the range from -1 to +1.

Syntax :- `asin(<number>)`

Example :- `result = asin(number)*180.0/PI;`

(14)acos() :-

- This function is used to calculate principle value of any number in the range from 0 to π (3.14).

Syntax :- `acos(<number>)`

Example :- `acos(number)*180.0/1;`

(15)atan() :-

- This function is known as arc tangent function.
- It is used to calculate principal value of any given number in the range $[-\pi/2, +\pi/2]$.

Syntax :- `atan(<number>)`

Example :- `acos(number)*180.0/1;`

(16)fmod() :-

- This function is used to calculate floating point reminder.
- You can calculate value like x/y and it give you reminder.

Syntax :- `fmod(<number>,<divisible by>)`

Example :- `fmod(5.3,2);`

- DATE & TIME FUNCTIONS :-

Header File: :- `<time.h>`

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

<time.h>	
clock_t clock()	Returns clock ticks since program starts.
char *asctime(struct tm)	Converts date and time into ascii.
int stime(time_t *tp)	Sets time.
time_t time(time_t *timer)	Gets time of day.
double difftime(t1,t2)	Returns difference time between two times t1 and t2.

Detail :-

1. clock():

- clock() determines the amount of processor time used since the invocation of the calling process, measured in clocks_per_sec of a second.

Syntax:

clock_t clock(void)

2. mktime():

- mktime() transforms binary date and time values.

Syntax:

time_t mktime (struct tm *timeptr);

3. time():

- time() is used to get current system time.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Syntax:

```
time();
```

4. difftime():

- difftime() is used to get difference between two given times.

Syntax:

```
Difftime(endtime , starttime)
```

5. ctime():

- ctime() converts the time_t object pointed by timer to a C string containing human-readable version of the corresponding local time and date.
- The returned string has following format:

```
Www Mmm dd hh:mm:ss yyyy
```

- Where Www is the weekday, Mmm the month in letters, dd the day of the month, hh:mm:ss the time, and yyyy the year.

Syntax:

```
char * ctime ( const time_t * timer );
```

6. localtime():

- localtime() converts time_t to tm as localtime. localtime() uses the time pointed by timer to fill a tm structure with the values that that represent the corresponding local time.

Syntax:

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
struct tm * localtime ( const time_t * timer );
```

7. strftime():

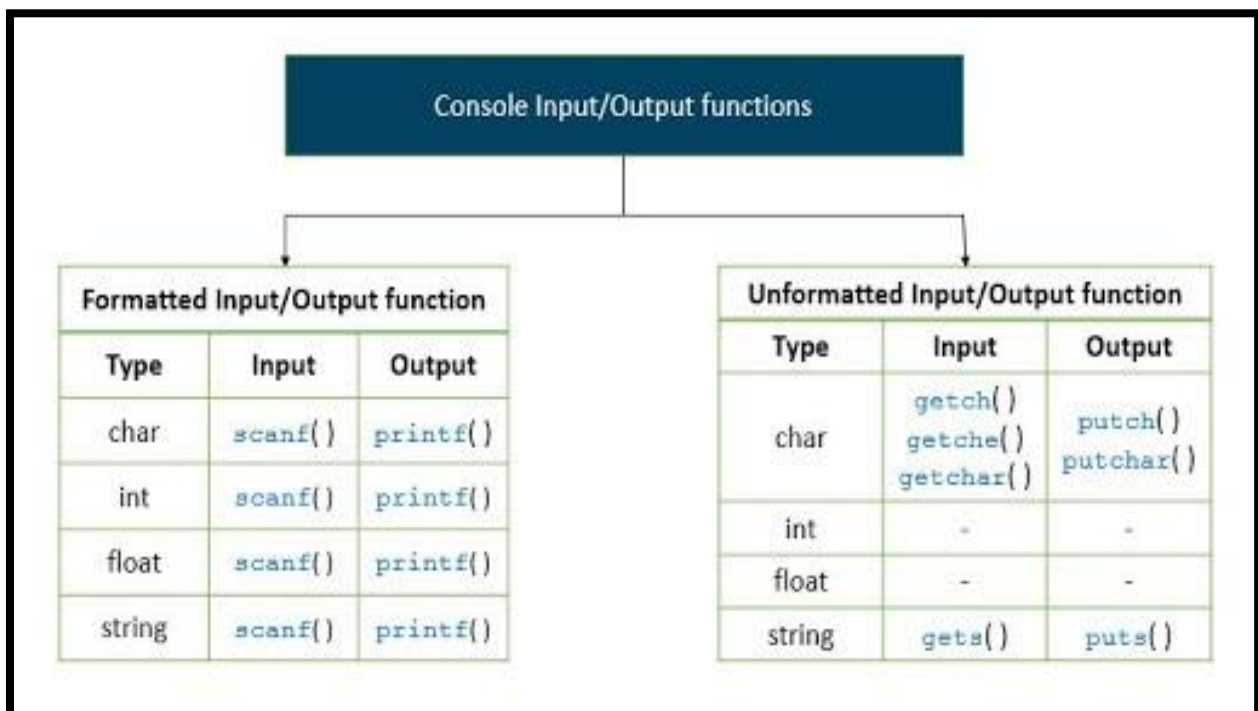
▪ strftime() format time to string. strftime() copies into a ptr the content of format, expanding its format tags into the corresponding values as specified by timeptr, with a limit of maxsize characters.

Syntax:

```
size_t strftime ( char * ptr, size_t maxsize, const char * format,  
const struct tm * timeptr );
```

▪ I/O FORMATTING FUNCTIONS :-

Header File: :- **<stdio.h>**



Detail :-

1.printf():

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- printf() is output function. printf() is used for printing formatted data to stdout. This function is generally used for printing text and numeric values.

Syntax:

```
printf("control string",arg1,arg2.....);
```

- In the above syntax, control string indicates how many arguments are there and what their types are. The control string consists of three types:
 1. Characters to be printed on screen.
 2. Format specifiers.
 3. Escape sequence characters. (\n, \t)

2.scanf():

- scanf() is an input function. It is used to read the data entered by the user. scanf() scans the formatted values.

Syntax:

```
scanf("control string",&arg1, &arg2.....);
```

- In the above syntax, control string specifies the field format in which the data is to be entered. The arguments &arg1, &arg2..... specifies the address of the locations where the data is stored
 - . scanf() is also required to read multiple values using the same scanf function. The control string consists of three types:
 1. Characters to be printed on screen.
 2. Format specifiers.
 3. Escape sequence characters. (\n, \t)

3.getc():

- This function is used to get a single character from input stream.

Syntax:

```
Getc (<input stream>)
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Example:

```
Ch = getc(stdin);
```

4.putc():

- This function is used to put/print a single character to output stream.

Syntax:

```
putc (<character> , <output stream>)
```

Example:

```
Ch = putc('a' , stdout);
```

5.gets():

- This function is used to get or read particular string with white space character.

Syntax:

```
Gets(<string>)
```

Example:

```
Gets("hello friends");
```

6.puts():

- This function is used to put or print particular string with white space character.

Syntax:

```
puts(<string>)
```

Example:

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
Void main()
{
    Char a[20];
    Printf("enter string");
    Gets(a);
    Puts(a);
    Getch();
}
```

7.getchar():

- This function is used to get a single character from standard input.

Syntax:

```
<variable> = getchar();
```

Example:

```
Void main()
{
    Char ch;
    Printf("enter character");
    Ch = getchar();
    Getch();
}
```

8.putchar():

- This function is used to print a single character to output stream.

Syntax:

```
Putchar(<character variable>)
```

Example:

```
Void main()
{
    Char ch;
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
Printf("enter character");  
Ch = getchar();  
Putchar(ch);  
Getch();  
}
```

- STANDARD LIBRARY FUNCTIONS :-

Header File: :- **<stdio.h>**

Standard C-Library Functions

<stdlib.h>

int atoi(s)	Converts string s to an integer
long atol(s)	Converts string s to a long integer.
float atof(s)	Converts string s to a double-precision quantity.
void* calloc(u1,u2)	Allocate memory to an array u1, each of length u2 bytes.
void exit(u)	Closes all files and buffers, and terminate the program.
void free (p)	Free block of memory.
void* malloc (u)	Allocate u bytes of memory.
int rand(void)	Return a random positive integer.
void* realloc(p,u)	Allocate u bytes of new memory to the pointer variable p.
void srand(u)	Initialize the random number generator.
void system(s)	Pass command string to the operating system.

Detail :-

1.atol():

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- This function is used to convert string argument to long integer value.

Syntax:

```
atol(<string>)
```

Example:

```
Void main()
{
    Char a[20] = "hi friends";
    long b;
    b= atol(a);
    printf("string=%s",longint = %ld",a,b);
    getch();
}
```

2.atof():

- This function is used to convert string argument to float number.

Syntax:

```
atof(<string>)
```

Example:

```
Void main()
{
    Char a[20] = "1234567";
    float b;
    clrscr();
    b= atof(a);
    printf("string=%s",float = %f",a,b);
    getch();
}
```

3.exit():

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- This function is used to terminate calling process immediately.

Syntax:

```
exit(<int>)
```

Example:

```
Void main()
{
    Clrscr();
    Printf("starting of program\n");
    Exit(0);
    Printf("exiting of program\n");
    getch();
}
```

4.labs():

- This function is used to return absolute value of long integer value.

Syntax:

```
labs(<long integer>)
```

Example:

```
Void main()
{
    long int a,b;
    a=labs(65987);
    Printf("%ld\n",a);
    B=labs(-100509);
    Printf("%ld\n",b);
    getch();
}
```

5.free():

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- This function is used to free or de-allocate the memory allocated by calloc , malloc and re-alloc.

Syntax:

Free(<pointer variable>)

Example:

```
Void main()
{
    Char *str;
    Str = (char *) malloc(15);
    Strcpy(str,"tutorial point");
    Printf("string=%s ,address=%u\n",str,str);
    Free(str);
    Getch();
}
```

6.rand():

- This function is used to generate random numbers in the range from 0 to n.

Syntax:

labs(<long integer>)

Example:

```
Void main()
{
    int i,n;
    clrscr();
    printf("enter n");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        Printf("%d",rand()%50);
    }
}
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
        getch());  
    }
```

▪ MISCELLANEOUS FUNCTIONS :-

Header File: :- **<stdio.h>**

Miscellaneous functions	Description
getenv()	This function gets the current value of the environment variable
setenv()	This function sets the value for environment variable
putenv()	This function modifies the value for environment variable
perror()	Displays most recent error that happened during library function call
rand()	Returns random integer number range from 0 to at least 32767
delay()	Suspends the execution of the program for particular time

Function	Return Type	Use
isalnum(c)	int	Determine if the argument is alphanumeric or not
isalpha(c)	int	Determine if the argument is alphabetic or not
isascii(c)	int	Determine if the argument is ASCII character or not
isdigit(c)	int	Determine if the argument is a decimal digit or not.
toascii(c)	int	Convert value of argument to ASCII
tolower(c)	int	Convert character to lower case
toupper(c)	int	Convert letter to uppercase

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :-

1. delay():<dos.h>

- This function is used to suspend execution of program for particular amount of time.

Syntax:

```
delay(<int>)
```

Example:

```
#include<dos.h>
```

```
Void main()
```

```
{
```

```
    printf("This c program will exit in 10 seconds.\n");
```

```
    delay(10000);
```

```
    getch();
```

```
}
```

- Above c program exits in ten seconds, after the printf function is executed the program waits for 10000 milliseconds or 10 seconds and then program termination occurs.

2. clrscr():

- This function is used to clear the output of the screen.

Syntax:

```
Clrscr();
```

Example:

```
Void main()
```

```
{
```

```
    Int i=10;
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
    clrscr();
    printf("%d",i);
    getch();
}
```

3. islower():<ctype.h>

- This function is used to check if entered character is in lowercase or not.

Syntax:

```
islower(character);
```

Example:

```
#include<ctype.h>

void main()
{
    char ch = 'A';
    clrscr();
    if(islower(ch))
        printf("lowercase");
    else
        printf("not in lowercase");

    getch();
}
```

4. isupper():<ctype.h>

- This function is used to check if entered character is in uppercase or not.

Syntax:

```
isupper(character);
```

Example:

```
#include<ctype.h>
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
Void main()
{
    Char ch = 'A';
    Clrscr();
    If(isupper(ch))
        Printf("uppercase");
    Else
        Printf("not in uppercase");

    getch();
}
```

5. isspace():<ctype.h>

- This function is used to check if entered character is whitespace character or not.

Syntax:

```
isspace(character);
```

Example:

```
#include<ctype.h>
```

```
Void main()
{
    Char ch = ' ';
    Clrscr();
    If(isspace(ch))
        Printf("whitespace ");
    Else
        Printf("not whitespace");

    getch();
}
```

6. isalpha():<ctype.h>

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- This function is used to check if entered character is alphabet or not.

Syntax:

```
Isalpha(character);
```

Example:

```
#include<ctype.h>
```

```
Void main()
```

```
{
```

```
    Char ch = 1;
```

```
    Clrscr();
```

```
    If(isalpha(ch))
```

```
        Printf("alphabet");
```

```
    Else
```

```
        Printf("not alphabet");
```

```
    getch();
```

```
}
```

7. isprint():<ctype.h>

- This function is used to check if entered character is printable or not.

Syntax:

```
Isalpha(character);
```

Example:

```
#include<ctype.h>
```

```
Void main()
```

```
{
```

```
    Char ch = 1;
```

```
    Clrscr();
```

```
    If(isprint(ch))
```

```
        Printf("printable");
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
        Else
            Printf("not printable");

        getch();
    }
```

8. isdigit():<ctype.h>

- This function is used to check if entered character is digit or not.

Syntax:

```
Isdigit(character);
```

Example:

```
#include<ctype.h>

Void main()
{
    Char ch = 'a';
    Clrscr();
    If(isdigit(ch))
        Printf("digit");
    Else
        Printf("not digit");

    getch();
}
```

9. isalnum():<ctype.h>

- This function is used to check if entered character is alphabet or number.

Syntax:

```
Isalnum(character);
```

Example:

```
#include<ctype.h>
```


SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
Void main()
{
    Char ch = 'a';
    Clrscr();
    If(isalnum(ch))
        Printf("alpha-numeric");
    Else
        Printf("not alpha-numeric");

    getch();
}
```

10. isctrl():<ctype.h>

- This function is used to check if entered character is control character or not.

Syntax:

```
Isdigit(character);
```

Example:

```
#include<ctype.h>
```

```
Void main()
{
    Char ch = '%';
    Clrscr();
    If(isctrl (ch))
        Printf("control character");
    Else
        Printf("not control character");

    getch();
}
```

11.errno():<ctype.h>

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- This function is used to return error number related to particular object.
- **Note :-** you have to include header file `<errno.h>`

Syntax:

```
errno( );
```

Example:

```
#include<errno.h>

Void main()
{
    File *fp;
    Fp=fopen("hello.txt","r");
    Printf("errno=%d\n",errno());
    getch();
}
```

12.toupper():<ctype.h>

- This function is used to convert lowercase character into uppercase.

Syntax:

```
toupper(<character>)
```

Example:

```
#include<ctype.h>

Void main()
{
    Char ch= 'a';
    Clrscr();
    Int c=toupper(ch);
    Printf("%c",c);
    getch();
}
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

}

13.tolower():<ctype.h>

- This function is used to convert uppercase character into lowercase.

Syntax:

tolower(<character>)

Example:

```
#include<errno.h>

Void main()
{
    Char ch= 'A';
    Clrscr();
    Int c=tolower(ch);
    Printf("%c",c);
    getch();
}
```

- MEMORY ALLOCATION FUNCTIONS :-

Header File: :- **<stdio.h>**

- In c language there are two types of memory allocation:
 - Static memory allocation -design-time
 - Dynamic memory allocation -runtime

1.malloc():

- This function is used to allocate single block of requested memory.
- It always return NULL if memory is not sufficient.

Syntax:

Ptr = (cast_type) malloc(byte_size)

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Example:

```
Ptr = (int) malloc(100*sizeof(int))
```

2 calloc():

- This function is used to allocate multiple block of requested memory.
- It always return 0(zero) if memory is not sufficient.

Syntax:

```
Ptr = (cast_type) calloc(number ,byte_size)
```

Example:

```
Ptr = (float) calloc(25,sizeof(float))
```

3 realloc():

- If memory is not sufficient for malloc & calloc at that time you can use realloc() to re – allocate the memory space.
- It always return 0(zero) if memory is not sufficient.

Syntax:

```
Ptr = realloc(ptr , newsize)
```

Example:

```
Ptr = realloc(p,n*sizeof(int))
```

4.free():

- The memory allocated by malloc and calloc will be released or free by calling free().
- It occupies memory until you exit your program

Syntax:

```
Free(<pointer variable>)
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

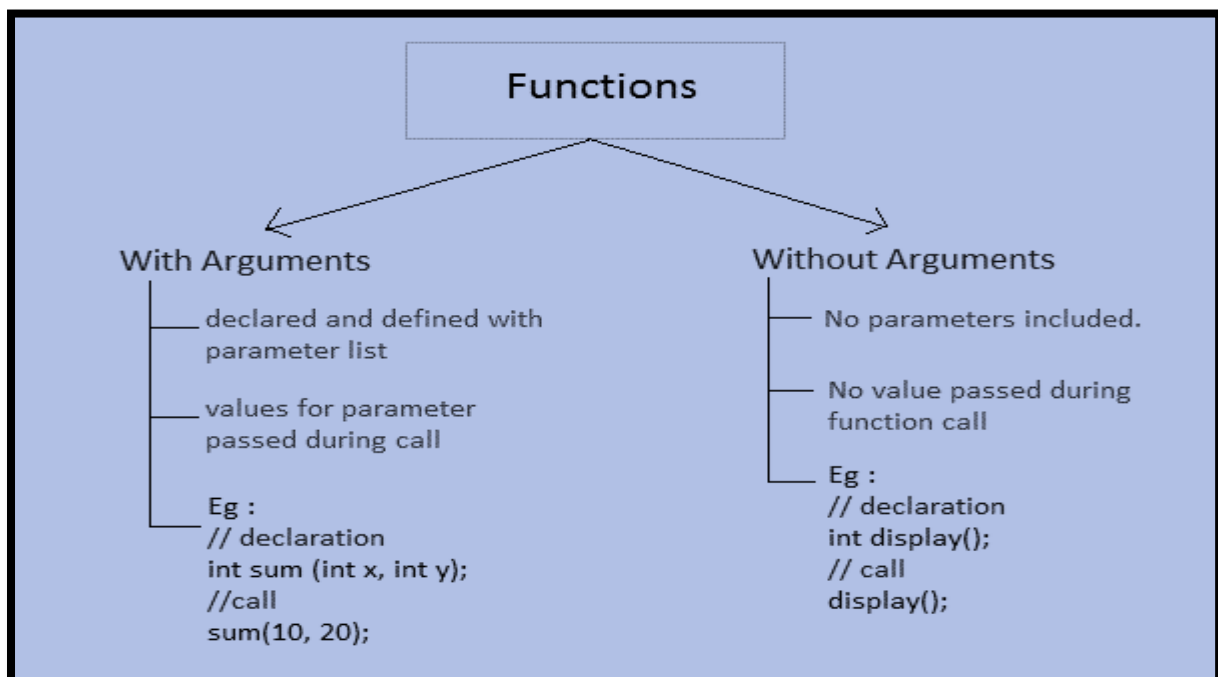
(Affiliated to Saurashtra University & Gujarat Technological University)

Example:

```
Void main()
{
    Char *str;
    Clrscr();
    Str = "tutorialpoint";
    Printf("string=%s,address=%u\n",str,str);
    Free(str);
    getch();
}
```

<i>malloc()</i>	<i>calloc()</i>
<ol style="list-style-type: none">1) malloc() has only one argument.2) malloc() allocates single block of storage space3) malloc() gives the garbage value of memory	<ol style="list-style-type: none">1) calloc() has two arguments.2) calloc() allocates multiple blocks of storage space.3) calloc() initializes the memory to zero.

Q-3 What is UDF? Explain with types.



Detail :

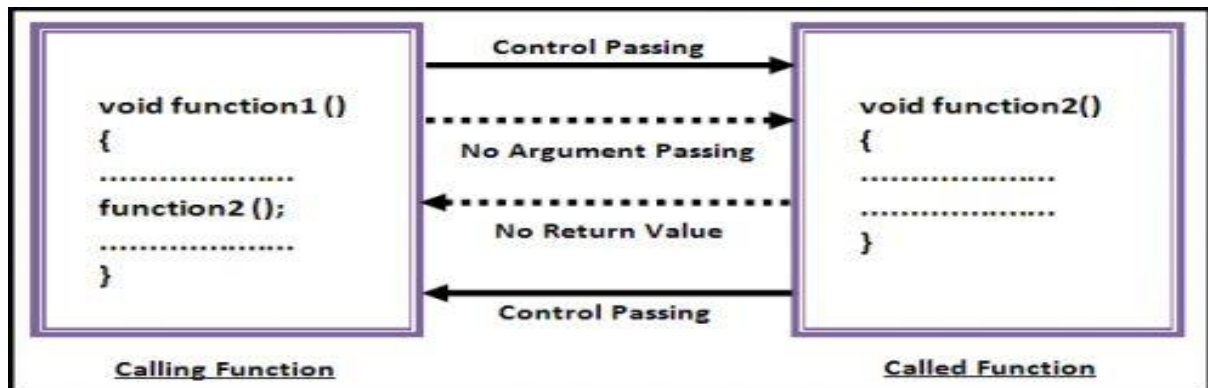
SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

There are 4 types of UDF:

1. Function with no arguments and with no return value
2. Function with no arguments and with return value
3. Function with arguments and no return value
4. Function with arguments and with return value

1. Function with no arguments and with no return value:



No Way Data Communication

This type of UDF does not receive any values from the calling function as well as it does not return any value. The calling function does not receive any values from the called function that means there is no transfer of data between the calling and called function. Only the control passes to the called function.

Example:

```
#include<stdio.h>
#include<conio.h>

void print(); // Function Declaration

void main()
{
    print(); // Function call
    getch();
}

void print() // Function Definition
{
```

Output:

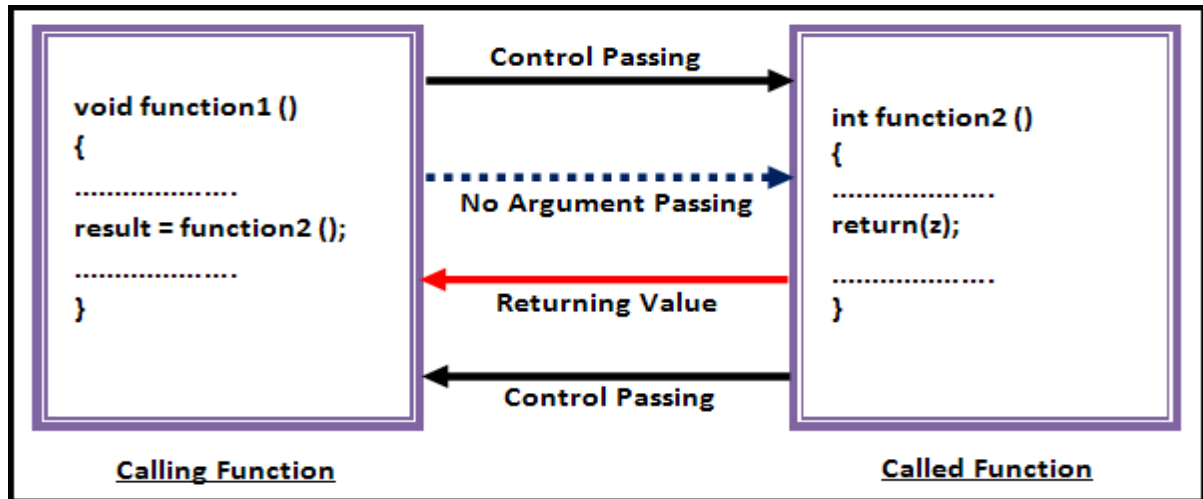
Hello

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
printf("\n Hello");  
}
```

2. Function with no arguments and with return value:



One Way Data Communication

This type of UDF does not have any arguments so called function does not receive any data from the calling function but it has return type so calling function receive the data from the called function. So, in this type of UDF, one way data communication takes place.

Example:

```
#include<stdio.h>  
#include<conio.h>  
  
int add(); // Function Declaratio  
  
void main()  
{  
    int x;  
    x=add(); // Function call
```

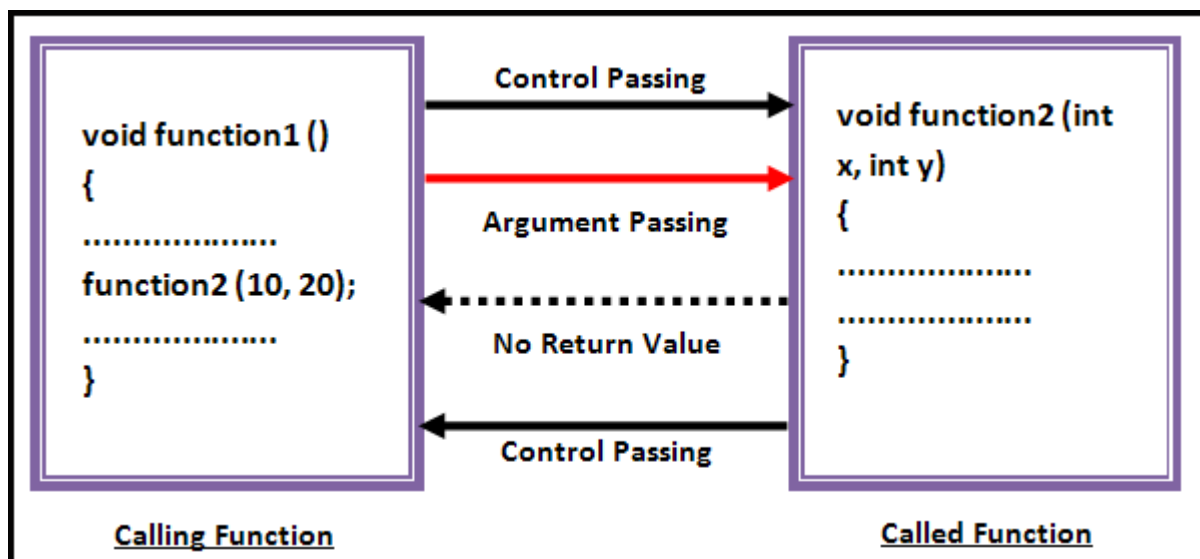
Output:
enter the first number 5
enter the second number 5
answer=10

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
        printf("\n answer=%d",x);
        getch();
    }
    int add() // Function Definition
    {
        int a,b,c;
        printf("\n enter the first number");
        scanf("%d",&a);
        printf("\n enter the second number");
        scanf("%d",&b);
        c=a+b;
        return c;
    }
```

3. Function with arguments and with no return value:



One Way Data Communication

This type of UDF have arguments so called function receives data from the calling function but it does not have return type so calling function does

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

not receive the data from the called function. So, in this type of UDF, one way data communication takes place.

Example:

```
#include<stdio.h>
#include<conio.h>

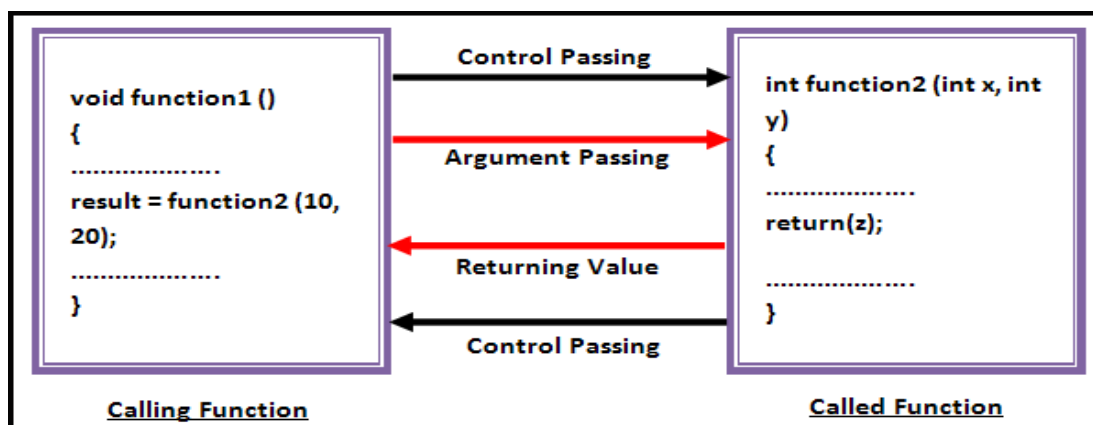
void add(int, int); // Function Declaration

void main()
{
    int x,y;
    printf("\n enter x:");
    scanf("%d",&x);
    printf("\n enter y:");
    scanf("%d",&y);
    add(x,y); // Function call
    getch();
}

void add(int a, int b) // Function Definition
{
    int c;
    c=a+b;
    printf("\n addition=%d",c);
}
```

Output:
enter x: 5
enter y: 6
addition=11

4. Function with arguments and with return value:



SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Two Way Data Communication

This type of UDF have both arguments and return value so called function receives the data from calling function and calling function also receives the data from the called function so in this type of UDF, two way communication takes place.

Example:

```
#include<stdio.h>
#include<conio.h>

int add(int, int); // Function Declaration

void main()
{
    int x,y,z;
    printf("\n enter x:");
    scanf("%d",&x);
    printf("\n enter y:");
    scanf("%d",&y);
    z=add(x,y); // Function call
    printf("\n addition=%d",z);
    getch();
}

int add(int a, int b) // Function Definition
{
    int c;
    c=a+b;
    return c;
}
```

Output:
enter x: 5
enter y: 6
addition=11

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-4 Explain Pointer in detail.

WHAT IS POINTER?

- A **pointer** is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before you can use it to store any variable address.
- The general form of a pointer variable declaration is:

```
dataType *var_name;
```

- Here,
 - **dataType** is the pointer's base type; it must be a valid C data type (i.e., int, float, char etc).
 - **var_name** is the name of the pointer variable.
 - The asterisk * you used to declare a pointer is the same asterisk that you use for multiplication. However, in this statement the asterisk is being used to designate a variable as a pointer.

Pointer Declaration

- **Type * Varname;**
declares **Varname** as a pointer to **Type**

- **Examples**

```
char* A;  
int* B;  
struct node * Root;
```

A is a pointer to a character

B is a pointer to an integer

Root is a pointer to a node structure

- *** associates with the variable name**

```
char *A, B;
```

A is a pointer to a character

B is a character !

Pg. 9

Detail :-

Pointer is a user defined data type which creates special types of variables which can hold the address of primitive data type like char, int, float, double or user defined data type like function, pointer etc. or derived data type like array, structure, union, enum. In C programming every variable keeps two types of value:

Value of variable

Address of variable where it has stored in the memory.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Example 1:

```
int a=5  
int *ptr;  
ptr=&a;
```

Explanation:

About variable a:

Name of the variable: a

Value of the variable which it keeps: 5

Address where it has stored in the memory : 1025 (assume)

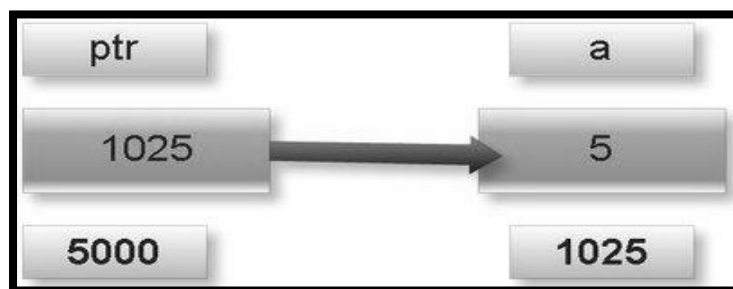
About variable a:

Name of the variable : ptr

Value of the variable which it keeps : 1025

Address where it has stored in the memory : 5000 (assume)

Pictorial Representation:



Note:

A variable where it will be stored in memory is decided by operating system. We cannot guess at which location a particular variable will be stored in memory.

Example 2:

```
int a=50;  
int *ptr1;  
int **ptr2;  
ptr1=&a;  
ptr2=&ptr1;
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Explanation:

About variable a:

Name of the variable: a

Value of variable which it keeps: 50

Address where it has stored in memory : 5000 (assume)

About variable ptr1:

Name of variable : ptr1

Value of variable which it keeps: 5000

Address where it has stored in memory : 9000 (assume)

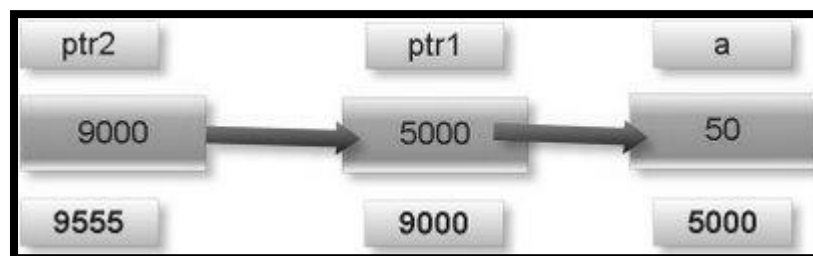
About variable ptr2:

Name of variable : ptr2

Value of the variable which it keeps: 9000

Address where it has stored in memory: 9555 (assume)

Pictorial Representation:



Note:

1. * is known as indirection operator which gives the content of any variable.
2. & is known as reference operator which gives address where variable has stored in memory.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-5 Explain Call by value and Call by reference with example.

Call by value

- Calling a function with parameters passed as values

```
int a=10;          void fun(int a)
fun(a);           {
                  {
                  defn;
                  }
                  }
```

Here fun(a) is a **call by value**.
Any modification done within the function is local to it and will not be effected outside the function

Function call by Value:

Detail :-

When a function is called by passing value of variables then that function is known as 'function call by passing values.' When the function is called by passing value of the variables, the changes made to the values of formal arguments (formal arguments-that are passed in function definition) are not reflected in actual arguments (actual arguments – that are passed in function call).

Advantage:-

- Call by value is very simple and it reduces confusion.

Disadvantage:-

- Any changes made in the formal arguments are not reflected to actual arguments in call by value.

Example:

First PROGRAMMING IN C

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
#include<stdio.h>
#include<conio.h>
void swap(int,int); //function declaration

void main()
{
    int a=10,b=5;
    clrscr();
    printf("\n value of a before swapping =%d",a);
    printf("\n value of b before swapping=%d",b);
    swap(a,b); //function call
    printf("\n value of a after swapping=%d",a);
    printf("\n value of b after swapping=%d",b);
    getch();
}

void swap(int x,int y) //function definition
{
    int temp;
    temp=x;
    x=y;
    y=temp;
    printf("\n value of x after swapping=%d",x);
    printf("\n value of y after swapping=%d",y);
}
```

Output:

value of a before swapping=10
value of b before swapping=5

value of x after swapping=5
value of y after swapping=10

value of a after swapping=10
value of b after swapping=5

Call By Reference

- Calling a function by passing pointers as parameters (address of variables is passed instead of variables)

```
int a=1;          void fun(int *x)
fun(&a);          {
                  defn;
                  }
```

Any modification done to variable a will effect outside the function also

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Function call by reference:

Detail :-

When a function is called by passing address of variables then that function is known as 'function call by reference.' When the function is called by passing value of the variables, the changes made to the values of formal arguments (formal arguments-that are passed in function definition) are reflected in actual arguments (actual arguments – that are passed in function call).

Advantage:-

- Any changes made in the formal arguments are reflected to actual arguments in call by reference.

Example:

```
#include<stdio.h>
#include<conio.h>
void swap(int,int); //function declaration

void main()
{
    int a=10,b=5;
    clrscr();
    printf("\n value of a before swapping =%d",a);
    printf("\n value of b before swapping=%d",b);
    swap(a,b); //function call
    printf("\n value of a after swapping=%d",a);
    printf("\n value of b after swapping=%d",b);
    getch();
}

void swap(int x,int y) //function definition
{
    int temp;
    temp=x;
```

Output:
value of a before swapping=10
value of b before swapping=5

value of x after swapping=5
value of y after swapping=10

value of a after swapping=5
value of b after swapping=10

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
x=y;  
y=temp;  
printf("\n value of x after swapping=%d",x);  
printf("\n value of y after swapping=%d",y);  
}
```

5.8 Calling Functions: Call by Value and Call by Reference

- Used when invoking functions
- Call by value
 - Copy of argument passed to function
 - Changes in function do not effect original
 - Use when function does not need to modify argument
 - Avoids accidental changes
- Call by reference
 - Passes original argument
 - Changes in function effect original
 - Only used with trusted functions
- For now, we focus on call by value

C Lecture Notes

Q-6 Explain Recursion with Example.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

RECURSION

When a called function in turn calls another function a process of chaining occurs. Recursion is a special case of this process, where a function calls itself.

Example :

```
void main ()
{
printf ( "\n This is an example of recursion");
main (); ← Recursive function call
}
```

Recursive Functions

```
int recursion ( x )
{
if ( x==0 ) ← Base case
return;
recursion ( x-1 ); ← Function being called again by itself
}
```

Recursive Definitions

- **Recursion**: solving a problem by reducing it to smaller versions of itself
- $0! = 1$ (1)
- $n! = n \times (n-1)!$ if $n > 0$ (2)
- The definition of factorial in equations (1) and (2) is called a recursive definition
- Equation (1) is called the base case
- Equation (2) is called the general case

C++ Programming: From Problem Analysis to Program Design, Third Edition

3

return 5 * factorial(4) = 120

└─ return 4 * factorial(3) = 24

└─ return 3 * factorial(2) = 6

└─ return 2 * factorial(1) = 2

└─ return 1 * factorial(0) = 1

javaTpoint.com

1 * 2 * 3 * 4 * 5 = 120

Fig: Recursion

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :-

When the function calls itself it is known as recursion.

Example:

```
#include <stdio.h>
#include <conio.h>
```

```
void main()
{
    clrscr();
    Recursion();
}
```

```
void Recursion()
{
    int no;
    printf("\nRecursion... ");
    printf("\n\n Enter Number : ");
    scanf("%d",&no);
    if (no==3)
        exit(0);
    else
        Recursion();
}
```

Output:

Recursion...

Enter Number : 2

Recursion...

Enter Number : 1

Recursion...

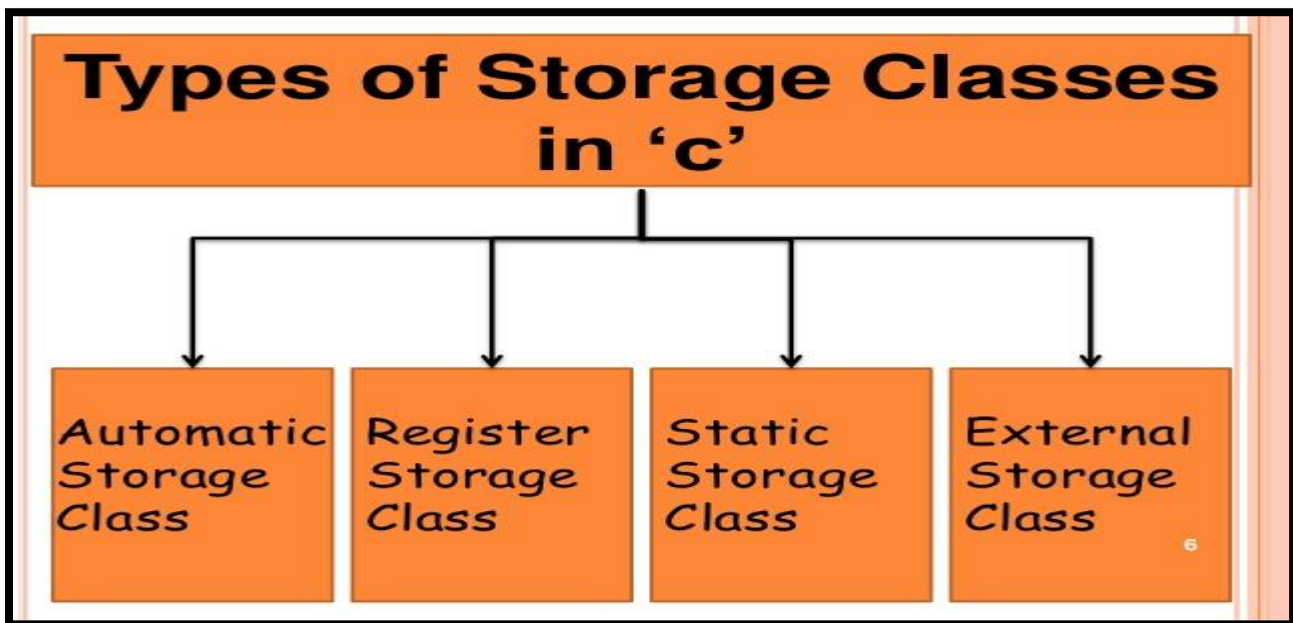
SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-7 Explain Storage Class in detail.

Detail :-

Storage class determines the scope (region of the program in which a variable is available for use.) and lifetime (duration in which a variable exist in the memory during execution of a program.). Storage class tells:



Storage Class	Declaration Location	Scope (Visibility)	Lifetime (Alive)
auto	Inside a function/block	Within the function/block	Until the function/block completes
register	Inside a function/block	Within the function/block	Until the function/block completes
extern	Outside all functions	Entire file plus other files where the variable is declared as extern	Until the program terminates
static (local)	Inside a function/block	Within the function/block	Until the program terminates
static (global)	Outside all functions	Entire file in which it is declared	Until the program terminates

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Storage classes in C Programming Language

Storage Class	Keyword used
Automatic	auto
External	extern
Static	static
Register	register

Sanyam.Khurana@TheGeekyWay.com

www.TheGeekyway.com

- Where the variable is stored.
- Initial value of the variable.
- Scope of the variable. Scope specifies the part of the program which a variable is accessed.
- Life of the variable.

There are four types of storage class:

1. Automatic Storage class
2. Register Storage class
3. Static Storage class
4. External Storage Class

1. Automatic Storage Class:

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

The variables that are local to the function are known as automatic variables i.e. declared within the function. It is the default storage class for the variables declared in a function.

- ◆ Keyword : auto
- ◆ Storage Location : Main memory
- ◆ Initial Value : Garbage Value
- ◆ Life : Local (function in which it is declared).
- ◆ Scope : Local to the block in which variable is declared.

Example:

```
#include <stdio.h>
#include <conio.h>

void main()
{
    auto int i=10;
    clrscr();
    {
        auto int i=20;
        printf("\n\t %d",i);
    }
    printf("\n\n\t %d",i);
    getch();
}
```

Output:

20

10

2. Register Storage Class:

Register variables are declared inside the function. When the variables are required very often during the program, they may be declared as register variable. Register access is faster than memory access. Generally, the register variables are used as loop counters which are used multiple times in the program. Register variables are not applicable for arrays, structures or pointers.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- ◆ Keyword : register
- ◆ Storage Location : CPU Register
- ◆ Initial Value : Garbage
- ◆ Life : Local to the block in which variable is declared.
- ◆ Scope : Local to the block.

Example:

```
#include <stdio.h>
#include <conio.h>

void main()
{
    register int i;
    for(i=1;i<10;i++)
        printf("%d",i);
    getch();
}
```

Output:

1 2 3 4 5 6 7 8 9 10

3. Static Storage Class:

Static storage class can be used only if we want the value of a variable to persist between different function calls. If the control comes back to the same function again, the static variables have the same values they had last time around. By default, global variables are static variables.

- ◆ Keyword : static
- ◆ Storage Location : Main memory
- ◆ Initial Value: Zero and can be initialize only once.
- ◆ Life : depends on function calls and the whole application or program.
- ◆ Scope : Local to the block.

Example:

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
#include<stdio.h>
#include<conio.h>

void print();

void main()
{
    clrscr();
    print();
    print();
    getch();
}

void print()
{
    static int k;
    printf("\n the value of static variable is %d",k);
    k++;
}
```

Output:

**value of static variable is 0
value of static variable is 1**

4. External Storage Class:

The variables of external storage class can be referred to as 'global or external variables.' They are declared outside the functions and can be invoked at anywhere in a program. The only drawback of declaring the variables as extern is it wastes lot of memory because these variables remain active throughout the life of the program.

- ◆ Keyword : extern
- ◆ Storage Location : Main memory
- ◆ Initial Value : Zero
- ◆ Life : Until the program ends.
- ◆ Scope : Global to the program.

Example:

```
#include <stdio.h>
```


SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
#include <conio.h>

extern int i=10;
void main()
{
    int i=20;
    void show(void);
    clrscr();
    printf("\n\t %d",i);
    show();
    getch();
}
void show(void)
{
    printf("\n\n\t %d",i);
}
```

```
Output:
20
10
```

Q-8 Explain Pass by value and Return value.

```
#include <stdio.h>
return_type func_name(arguments);
{
    .....
    .....
}
int main()
{
    .....
    func_name(arguments_value);
    .....
    return 0;
}
```

↑
formal arguments

↓
actual arguments

Return statement of a Function

```
#include <stdio.h>

int addNumbers(int a, int b);

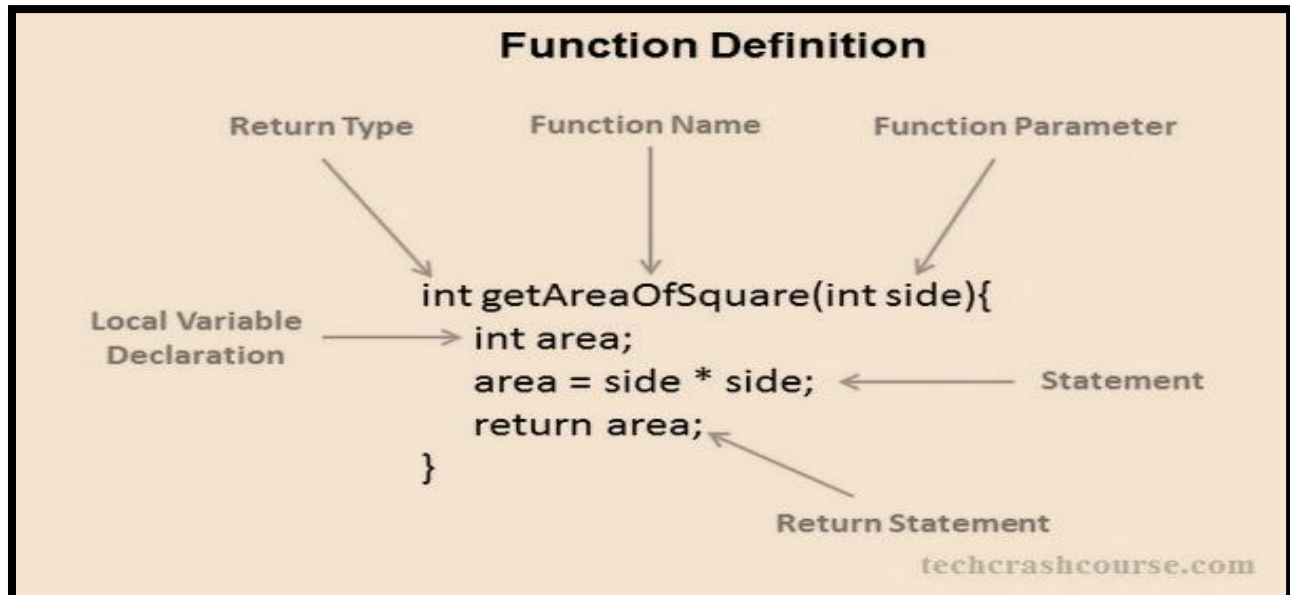
int main()
{
    ... ..
    sum = addNumbers(n1, n2);
    ... ..
}

int addNumbers(int a, int b)
{
    ... ..
    return result;
}
```

sum = result

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



Detail :-

When a function passes and returns value of variables then that function is known as 'function call by passing and returning values.'

Example:

```
#include <stdio.h>
#include <conio.h>
```

```
int number(int n)
{
    return n;
}
void main()
{
    int number(int);
    int a = number(4);
    clrscr();
    printf("\n Number is : %d",a);
    getch();
}
```

Output:

Number is: 4

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

SUMMARY QUESTIONS OF CHAPTER -3

ONE - MARK QUESTIONS

- ❖ What is Function?
- ❖ Give Full Form of UDF.
- ❖ List out Types of UDF.
- ❖ Which Operator is used to declare Pointer Variable?
- ❖ List out different Memory Allocation Functions.
- ❖ Write down Syntax for Realloc()

TWO - MARK QUESTIONS

- ❖ Explain any two String Functions with Example.
- ❖ Explain any two Maths Functions with Example.
- ❖ Explain any two Date & Time Functions with Example.

THREE - MARK QUESTIONS

- ❖ Explain Recursion with suitable Example.
- ❖ Explain any three I/O Formatting functions with Example.
- ❖ Explain Free() , Exit() ,atof() ,atol() with Example.
- ❖ Give difference between Malloc() and Calloc().

FIVE - MARK QUESTIONS

- ❖ Explain any six Miscellaneous Functions with Example.
- ❖ Explain Pointer with Suitable Example.
- ❖ Explain Call by Value and Call by Reference with Example.
- ❖ Write note on different Storage Classes.
- ❖ Explain User Define Function with Example.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

CHAPTER-4

➤ ARRAYS IN C

5.1 Types of arrays

5.1.1 Single Dimensional Array

5.1.2 Two Dimensional Array or Multi-Dimensional Array

5.1.3 String Arrays

5.2 Use of arrays in Programming

5.3 Arrays and matrices

SHREE H. N. SHUKLA GROUP OF COLLEGES

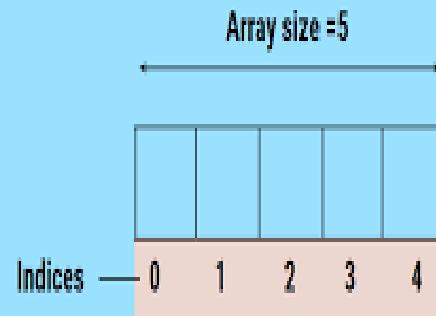
(Affiliated to Saurashtra University & Gujarat Technological University)

Q-1 What is Array ? Explain with types.

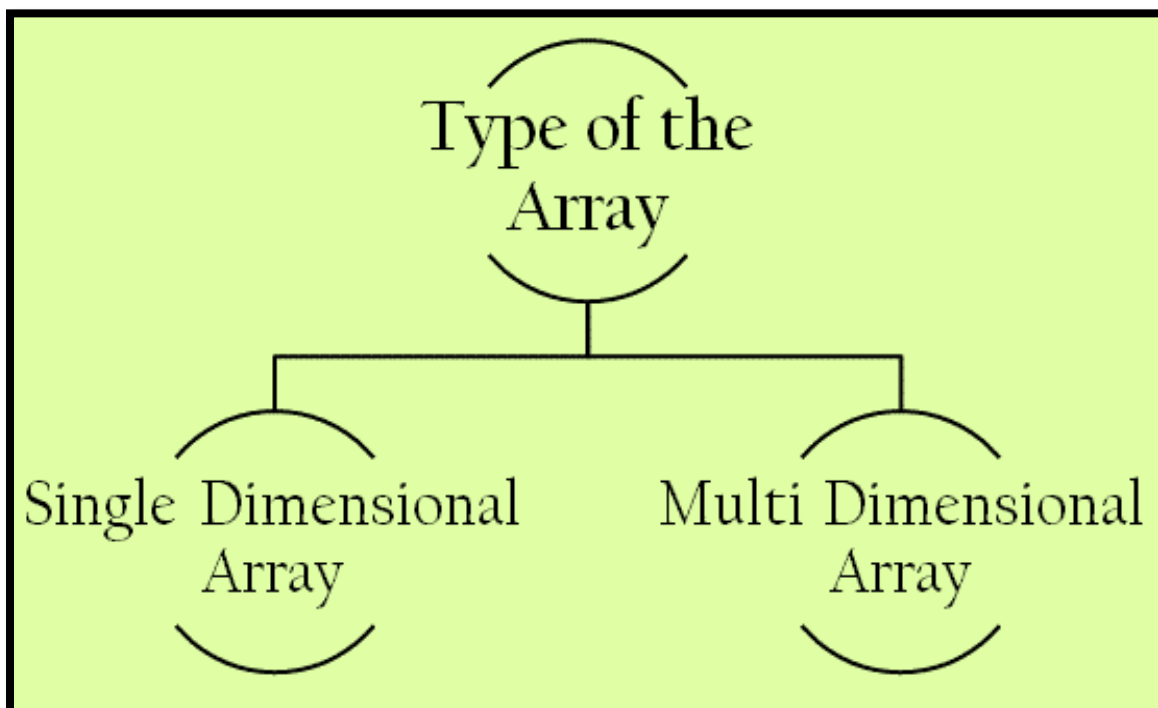
What is Array?

- An array is a fixed-size sequential collection of elements of same data types that share a common name.
- It is simply a group of data types.
- An array is a derived data type.
- An array is used to represent a list of numbers , or a list of names.

Arrays in C Programming



educba.com



Detail :-

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- ❖ Array is a group of elements that all have same data type.
- ❖ It is a data structure that can store multiple values at the same time in single variable.
- ❖ **How to declare array:- [1 mark]**
 - Array can be declare at design time and at run time.
 - To declare array , C language provide 3 criterias:
 - (i) Name of Array
 - (ii) Size of Array
 - (iii) Type of Array

Ex :- int a[3];

❖ **Types of Array :-**

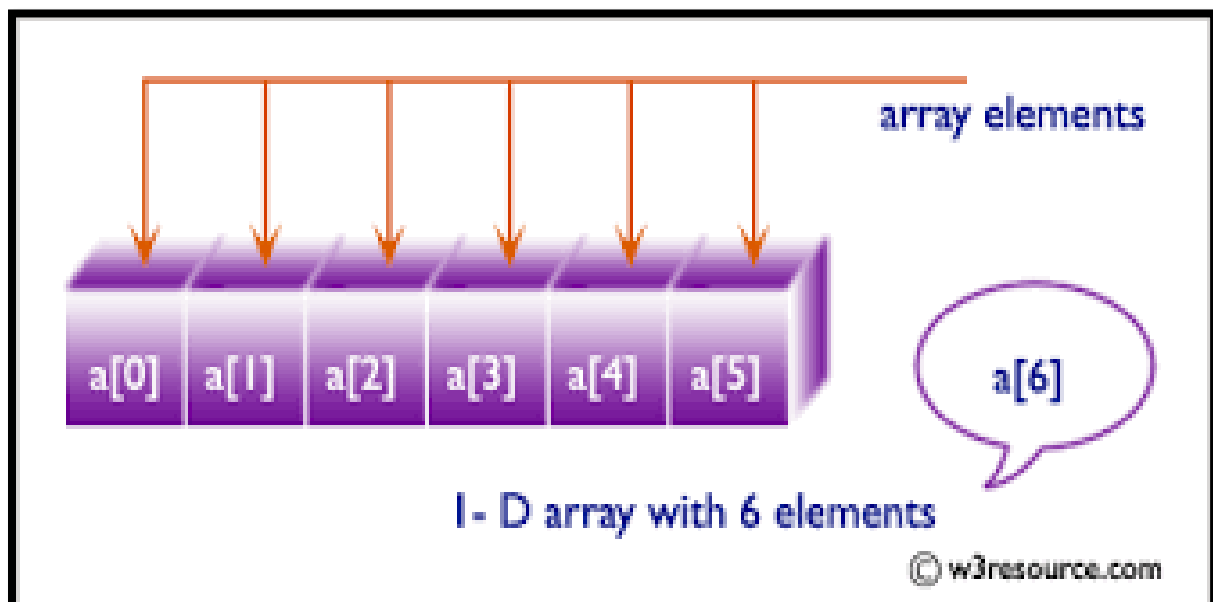
- Array can be of following types:
 - (i) One dimension array (1 - D)
 - (ii) Two dimension array (2 - D)
 - (iii) Multi dimension array (M – D)

❖ **One dimension array (1 - D) :-**

- 1 – D means if we provide only one dimension (one square bracket) with size.

Syntax :- <data type> <array name> [size of array]

Example :- int a[5];



SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Name	roll[0]	roll[1]	roll[2]	roll[3]	roll[4]	roll[5]	roll[6]	roll[7]
Values	12	45	32	23	17	49	5	11
Address	1000	1002	1004	1006	1008	1010	1012	1014

1-D Array memory arrangement

❖ Two dimension array (2 - D) :-

- In 2 – D array ,we have to provide two dimenstions (two square brackets) with size.
- One bracket for row & other bracket for column
- We can create table , matrix using 2- D array.

Syntax :- <data type> <array name> [size of row] [size of column]

Example :- int a[3][3]

TWO DIMENSIONAL ARRAY

Also referred to as a *table*, a two dimensional array has two indexes. The first index refers to the row, while the second refers to column. The syntax in declaring two-dimensional arrays in Visual C++ is:

<data type> <array name> [<row size>] [<column size>]

In a two-dimensional array, the array elements are arranged in row-major order.

int a [4] [3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12} } ;

	Column 0	Column 1	Column 2
Row 0	a[0] [0]	a[0] [1]	a[0] [2]
Row 1	a[1] [0]	a[1] [1]	a[1] [2]
Row 2	a[2] [0]	a[2] [1]	a[2] [2]
Row 3	a[3] [0]	a[3] [1]	a[3] [2]

MELJUN CORTES, MBA,MPA,BSCS

SHREE H. N. SHUKLA GROUP OF COLLEGES

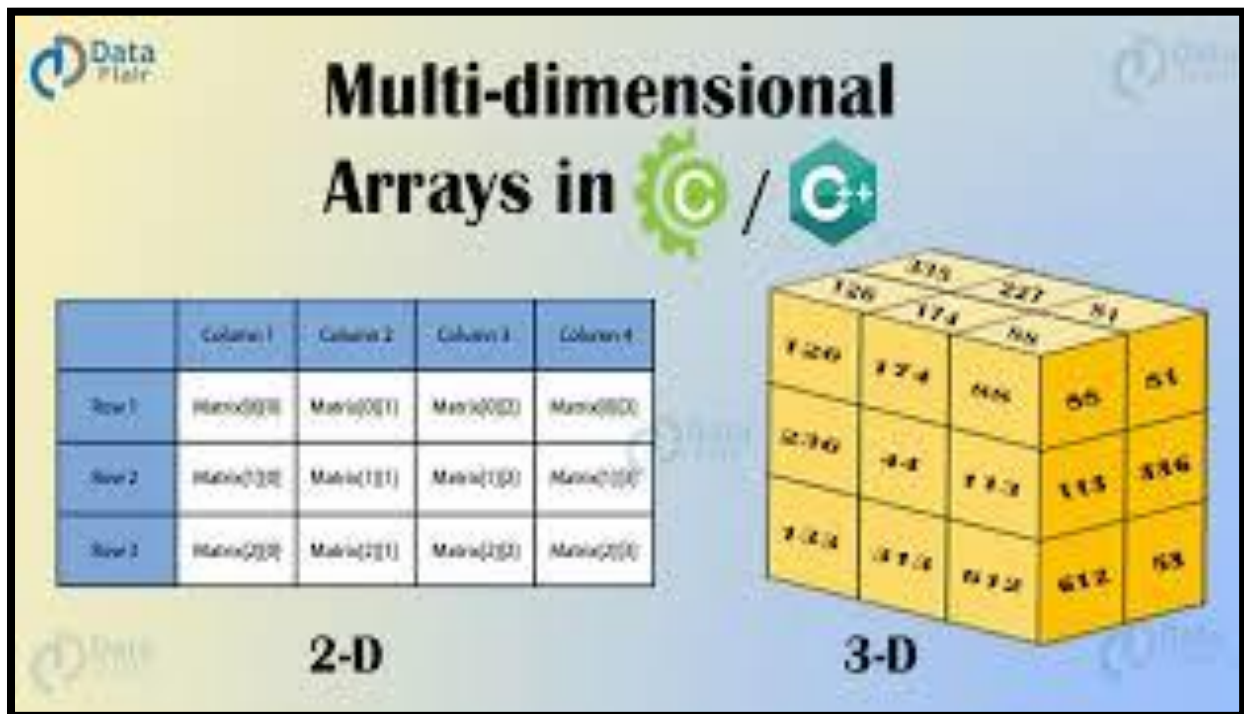
(Affiliated to Saurashtra University & Gujarat Technological University)

❖ Multi dimension array (M - D) :-

- M-D represented by more than 2 brackets at the same time with size.

Syntax :- <data type> <array name> (<size 1>) (<size 2>)....(size N)

Example :- int a[3][3][3]



1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	What is Array?	Group of Elements having same name and type.
2	Array is _____ datatype.	Derived
3	Array is used to represent _____	Collection
4	Types of array can be _____ & _____	Single/One dimension & Multi/Two dimension

Q-2 Explain Array Initialization in brief.

Detail :-

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

❖ Initialization of array elements:-

- ◆ After definition and declaration of array now array elements must be initialized.
- ◆ The two-dimensional array can be either initialized at :
 1. Compile time(at the time of declaration)
 2. Run time (using scanf statement or using assignment operator)



Compile time initialization

Example :

```
int number[ ] = {1,2,3,4};
```

- The character array can be initialized as follows :

```
char name[ ] = {'j','o','h','n','\0'};
```

- The character array can also be initialized as follows :

```
char name[ ] = "john";
```

12

1. Initialization of array at compile time:-

- ◆ In this type of initialization, the array elements can be initialized at the time of declaration.
- ◆ The values are assigned to each array elements enclosed within braces and separated by comma.

Example:-

```
data_type array_name[2] [3]={{1,2,3},{4,5,6}};
```

column1 column2 column3

1	2	3
---	---	---

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

row1 4 5 6

row 2

2. Initialization of array at run time:-



Run time initialization :

- In run time initialization, the array is explicitly initialize at run time.
- This concept generally used for initializing large arrays.
- Example:

```
for(i=0; i < 100; i++)
{
    if( i < 50)
        sum[i] = 0.0;
    else
        sum[i] = 1.0;
}
```

- Here first 50 elements of the array sum are initialized to 0 and the remaining 50 elements are initialized to 1 at run time.

13

- ◆ For initialization of two-dimensional array, we need nested for loops.
- ◆ The loop with counter i is used for row and the loop with counter j is used as columns.
- ◆ Initially, the value of i=0, which indicates the row 0 and the inner loop reads the values for the elements in each column in the row 0. It continues till the rows complete.
- ◆ Firstly declare an array `int a[3][3]` which will contain total (3*3=9) elements.. Now, using for loop we access the elements of this array as:

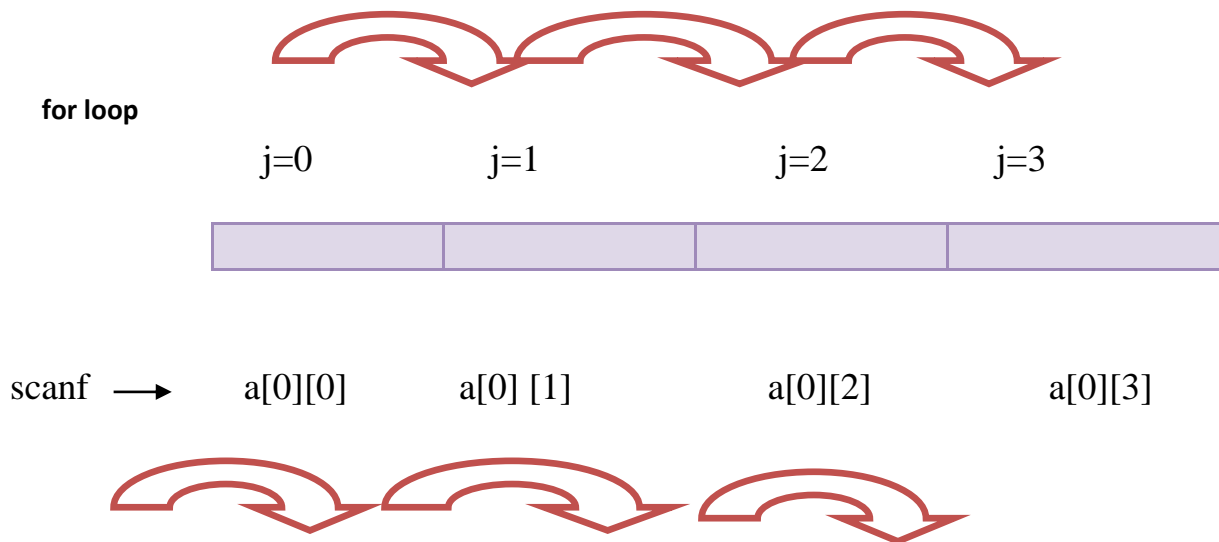
```
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        printf("\n enter elements a[%d][%d]",I,j);
        scanf("%d",&a[i][j]);
    }
}
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
}  
}
```

- ◆ The elements of array will ne initialized using the for loop under:-
- ◆ Initially the value of $i=0$, so the value of $a[0]$ will be scanned first using the scanf statement and likewise the loop executes for values upto $a[4]$.



1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	If array elements are initialized at the time of declaration then it is called _____ initialization.	Compile time
2	If array elements are initialized at the runtime then it is called _____ initialization.	Runtime
3	In _____ array initialization We have to assign fix value or size compulsory.	Compile time
4	Run time initialization of array is suitable for initializing _____ array.	Large

Q-3 Explain String Array in brief?How to declare String Array?

First PROGRAMMING IN C

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Arrays of Strings

- String is array of characters.
- Thus an array of string is 2-D array of characters.
- E.g.

```
char names[5][10];
```
- Here, names[5][10] means 5 names having 10 characters each.

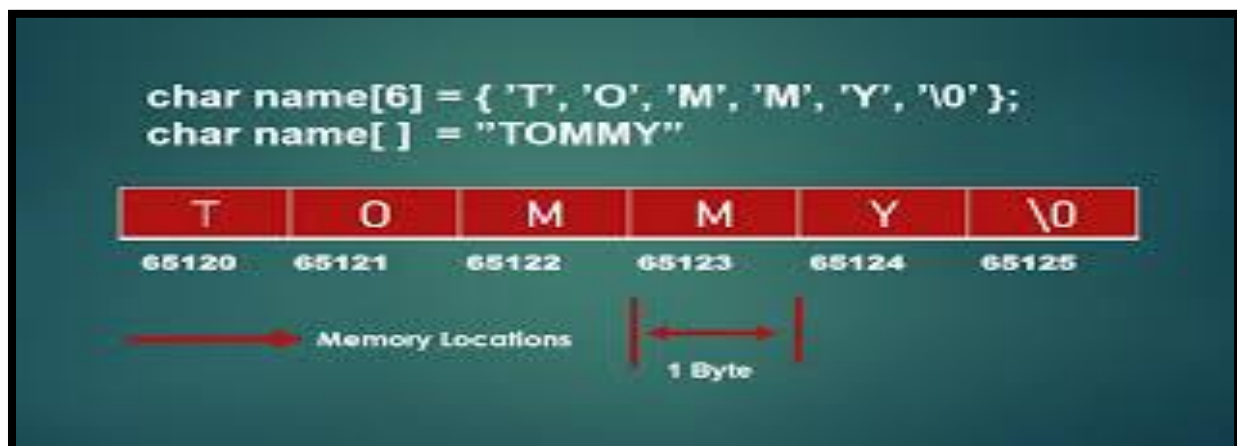
19

Detail :-

- ❖ The array of character is implemented in c which is known as string.
- ❖ The declaration of string array or character array can be like :

```
Char str[5] = { 'a','b','c','d','e','\0' }
```
- ❖ A NULL char '\0' must be added at the end of array string.
- ❖ The character array need one more space to store NULL char.
- ❖ C also provide facility to declare character array without specifying number of elements.
- ❖ In this case array size will be declare automatically.

❖ String Array Memory Allocation :-



SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

❖ Example:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    char arr[3][12]= { "Rose", "India", "technologies" };
    printf("Array of String is = %s,%s,%s\n", arr[0], arr[1], arr[2]);
    getch();
}
```

Output:

Array of String is=Rose, India, technologies

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	String array is _____ array of characters	2-D
2	String is array of _____ object.	Char
3	Write down statement to declare string array	Char arr[3][12]
4	String array represent _____ and _____.	Array size Total number of characters

Q-4 Explain Array with matrix.

Detail :-

- ❖ Array of Array is known as 2 – D array.
- ❖ 2-D array in C is also known as Matrix.
- ❖ Two dimensional array or multi-dimensional array are used to represent data in matrix form.
- ❖ Suppose , We declare array like a[3][3] then Matrix can be store as array like following:

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

	— i —————> row —————>			
j ↓ column	a [0][0]	a [0][1]	a [0][2]	a [i][j]
	a [1][0]	a [1][1]	a [1][2]	element
	a [2][0]	a [2][1]	a [2][2]	

Syntax:

`<data-type> <array_nm> [row_subscript][column-subscript];`

Example:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[3][3], i, j;
    clrscr();
    printf("\n\t Enter matrix of 3*3 : ");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            scanf("%d",&a[i][j]); //read 3*3 array
        }
    }
    printf("\n\t Matrix is : \n");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            printf("\t %d",a[i][j]); //print 3*3 array
        }
    }
    printf("\n");
}
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
}  
getch();  
}
```

Output:

Enter matrix of 3*3 : 3 4 5 6 7 2 1 2 3

Matrix is :

3	4	5
6	7	2
1	2	3

Matrix Addition

To add two or more matrices, first make sure they are the same size then add their corresponding elements

Matrix 1	Matrix 2	Matrix 1 + 2
$\begin{bmatrix} 10 & 0 \\ -4 & 5 \end{bmatrix}$	$\begin{bmatrix} -6 & 3 \\ 1 & -7 \end{bmatrix}$	$\begin{bmatrix} 4 & 3 \\ -3 & -2 \end{bmatrix}$
2 x 2	2 x 2	2 x 2

Subtraction of two Matrices

© w3resource.com

Matrix multiplication

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 0 & 7 \end{bmatrix} = \begin{bmatrix} 1*5 + 2*0 & 1*6 + 2*7 \\ 3*5 + 4*0 & 3*6 + 4*7 \end{bmatrix} = \begin{bmatrix} 5 & 20 \\ 15 & 46 \end{bmatrix}$$

How to multiply 2 matrices?

$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 5 \end{bmatrix}$	\times	$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$	$1 \times 1 + 2 \times 2 = 1 + 4 = 5$
$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 5 \end{bmatrix}$	\times	$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$	$1 \times 2 + 2 \times 4 = 2 + 8 = 10$
$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 5 \end{bmatrix}$	\times	$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$	$2 \times 1 + 4 \times 2 = 2 + 8 = 10$

& so on

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Matrix can be represented by _____ array.	2-D
2	In Array of Matrix elements can be represented by _____ and _____.	Rows Columns

➤ **STRUCTURES**

6.1 What is structure?

6.2 Memory Allocation functions

6.3 Pointers with structures

6.4 Array with structures

6.5 UDF with structures

6.6 Nested structures

6.7 Introduction to union

6.8 Difference between structure and union

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-5 What is structure? Explain initialization and declaration of structures.

Structure in C

Structure in c is a user-defined data type that enables us to store the collection of different data types. The ,struct keyword is used to define the structure.

```
struct structure_name
{
    data_type member1;
    data_type member2;
    .
    .
    data_type memeberN;
};
```

Sitesbay

STRUCTURE IN C

struct keyword tag or structure tag

```
struct Student
{
    char name[20];
    int roll number;
    float marks;
}
```

Members or fields of Structure

Detail :-

- ❖ Structure is group /collection of multiple variables tht all have different Data type.
- ❖ The variables inside the structure are called members of structure.
- ❖ To declare structure in C language ,”struct” keyword can be used.
- ✓ **How to declare(initialize) structure with members:**

Syntax:

Keyword **structure tag name**

↓ ↙

```
struct structure-tag
{
    datatype variable1;
    datatype variable2;
    datatype variable3;
}<Access Variable>;
```

↘ **Terminating semicolon**

Example:

First PROGRAMMING IN C

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
struct emp
{
    int empid;
    char empname[20];
    float salary;
};
```

- ❖ The members of the structure cannot be accessed directly as other variables, as they are associated with a structure.
- ❖ The structure elements or member variables associated with the structure are accessed using the structure **member operator (.)** **also called the dot operator** which is used between the structure name and the member name.

The members of the structure are accessed by two methods:

- 1) Accessing the values using member operator (.)
- 2) Accessing the values using scanf statement.

1. Accessing the values using member operator (.):-

Syntax:

```
Structure-variable . member-name=value;
                ↓
                dot operator
```

Example:

```
Emp01.salary=10000;
```

2. Accessing the values using scanf statement:-

Syntax:-

```
scanf("control string",&structure-variable . member-name);
```

Example:-

```
scanf("%f",&emp01.salary);
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Structure is _____ datatype.	User defined
2	Structure is a collection of _____ datatypes.	Different
3	_____ keyword is used to define structure.	Struct
4	The block of structure can be terminate with _____.	;(Semi colon)
5	To access member of the structure _____ opertator can be used.	.(dot)

Q-6 Explain Array and Structure.

Detail :

Array and structure are related in two ways:

1. Array of structure
2. Array within structure

Array of structure...

- Two ways to declare an array of structure:

```
struct student
{
  char name[20];
  int roll;
  char remarks;
  float marks;
  }st[100];
```

```
struct student
{
  char name[20];
  int roll;
  char remarks;
  float marks;
  };
struct student st[100];
```

20

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

1. Array of structure:

Detail :

Array of structure means creating the array of the type struct.

Example:. struct student s[5];

If we want to access the members of the structure, it can be done in following way:-

s[0].rollno	s[1].rollno
s[0].marks[0]	s[1].marks[0]
s[0].marks[1]	s[1].marks[1]
.	.
.	.
s[0].marks[4]	s[1].marks[4]

2. Array within structure:

Detail :

Array within structure means the array is used as member of the structure

Example:

```
struct student
{
    int roll;
    int marks[5]; // array within structure.
};
```

} members of structure

marks[0] → marks of subject 1

marks[1] → marks of subject

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-7 Explain Array within Structure with example.

ARRAYS WITHIN STRUCTURES

- ▶ C permits the use of array as a structure members.
- ▶ We can use single or multi-dimensional array of type **int** or **float**.
- ▶ i.e. Struct marks

```
{  
    int number;  
    float subject[3];  
} student[2];
```

- ❖ Array within structure is used to create multiple elements in the same time.
- ❖ If we want to store detail of students with multiple marks at that time array within structure can be used.
- ❖ Array within structure means we create array of particular member of structure.
- ❖ It is array within structure means array can be used as member of any structure.

Syntax: Struct <structure name>

```
{  
    <datatype>    <variable 1>;  
    <datatype>    <variable 2>[size];  
}<access variable>;
```

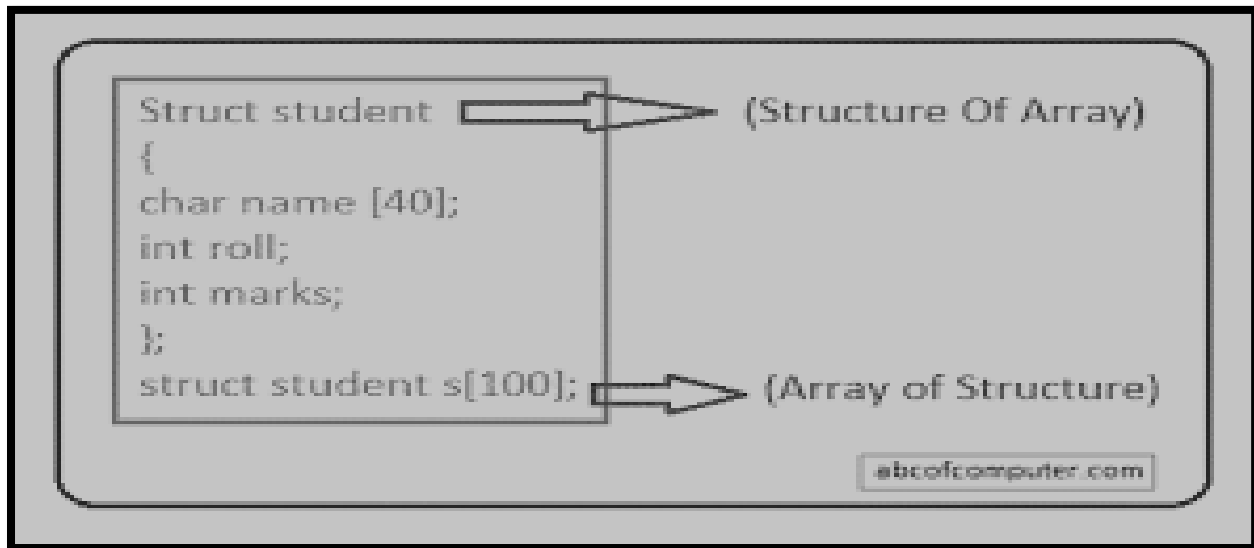
Example: Struct stud

```
{  
    Int rno;  
    Int marks[5]; //array within structure  
};
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Q-7 Explain Array of Structure with example.



- ❖ The array of structure means to work with more than one elements at the same time.
- ❖ If we want to store multiple records of multiple students at that time array of structure can be used
- ❖ Array of structure means creating array of every elements or members of structure.
- ❖ The array of structure can be represented by following manner:

Syntax:

```
Struct <structure name>
{
    <datatype>    <variable 1>;
    <datatype>    <variable 2>[size];
} <access variable>;
```

Example: Struct stud

```
{
    Int rno;
    Int marks;
}s[10]; //array of structure
```

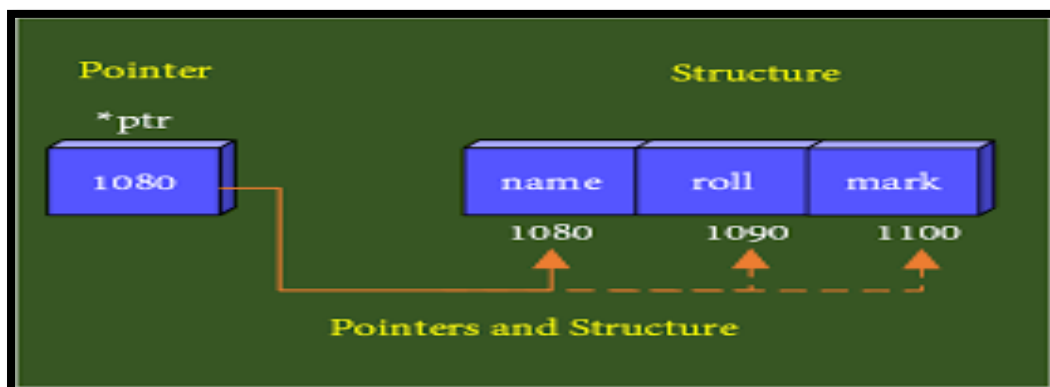
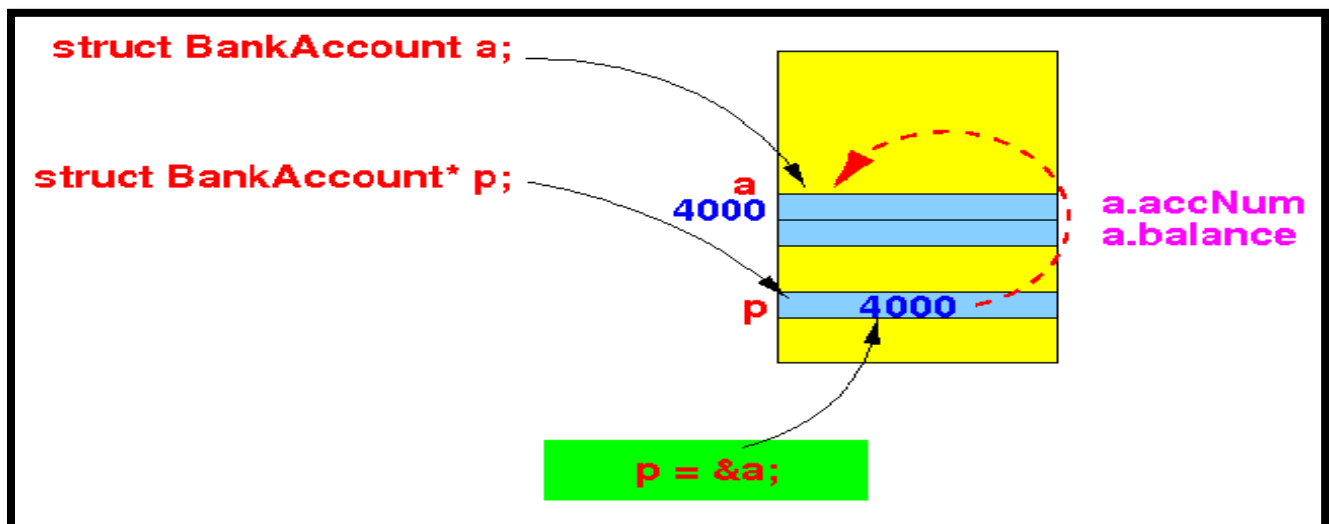
SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Array of structure means creating array of type _____	Struct
2	Array within structure means array is used as _____ of structure	Member
3	Write down statement to declare array within structure	Struct stud s { Int rno; Int marks[5]; }s;

Q-8 Explain Pointer with Structure with example.



Detail :

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- ◆ Structure is the data structure which consists of group of elements that may or may not have same data type.
- ◆ First we can define the structure like following;

Syntax:

```
struct student
{
    int roll;
    char name[20];
    char lname[20];
};
struct student s1;
```

- ◆ Defining the pointer to structure

```
struct student *p;
```

- ◆ It is already known that pointer must be initialized before it is used.
- ◆ Pointer to the structure can also be allocated memory dynamically.

```
p=(struct student *)malloc(sizeof(struct student));
```

Example:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

struct student
{
    int roll;
    char name[20];
    char lname[20];
}
```


SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
struct student s1;
void main()
{
    struct student *p;
    p=&s1;
    clrscr();
    s1.roll=1;
    strcpy(s1.name,"snehal");
    strcpy(s1.lname,"pandya");
    printf("\n rollno=%d",p->roll);
    printf("\n name=%s",p->name);
    printf("\n lname=%s",p->lname);
    getch();
}
```

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Pointer to structure can allocate memory _____.	Dynamically
2	In pointer to structure members can be access using _____ operator.	->(arrow operator)
3	Write down statement to declare pointer with structure.	Struct stud *p;
4	In pointer with structure _____ must be initialized before it is used.	pointer

Q-9 Explain UDF with Structure with example.

Detail :

- ❖ UDF stands for User Defined Function.
- ❖ UDF means the functions that are created by the user itself.
- ❖ It is also possible to pass structure as arguments to a function.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- ❖ When you create UDF with the help of structure then it is called UDF with structure.
- ❖ The following Example will display how to pass structure as argument in UDF.

Example:

```
#include<stdio.h>
#include<conio.h>      struct data
{
    float amount;
    char fname[30];
    char lname[30];
}rec;

void print_rec(struct data x)
{
    printf("\n %s %s %f",x.fname,x.lname,x.amount);
}

void main()
{
    clrscr();
    printf("\n Enter the donor's first and last names");
    scanf("%s %s", rec.fname, rec.lname);
    printf("\n Enter the donation amount");
    scanf("%f",&rec.amount);
    print_rec(rec);
    getch();
}
```

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
--------	----------	--------

First PROGRAMMING IN C

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

1	With UDF , you can pass structure as _____ of user defined function	argument
---	---	----------

Q-10 Explain Nested Structure with example.

Detail :

- ❖ Nested Structure means you can create one structure inside another structure.
- ❖ In simple words, “**Structure within a structure**” is called nested structure.
- ❖ When the structure is declared as member of another structure then it is called structure within structure.
- ❖ The process is known as nested structure.



Syntax:

```
struct structure_nm
{
    <data-type> element 1;
    <data-type> element 2;
    -----
    -----
    <data-type> element n;

    struct structure_nm
    {
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
<data-type> element 1;
<data-type> element 2;
-----
-----
<data-type> element n;
}inner_struct_var;
}outer_struct_var;
```

Example :

```
struct stud_Res
{
    int rno;
    char nm[50];
    char std[10];

    struct stud_subj
    {
        char subjnm[30];
        int marks;
    }subj;
}result;
```

Nested Structures

Structures can be nested within other structures in C programming.

```
struct complex
{
    int imag_value;
    float real_value;
};
struct number{
    struct complex c1;
    int real;
}n1,n2;
```

Suppose you want to access imag_value for n2 structure variable then, structure member n1.c1.imag_value is used.

1 Word Question – Answer

SHREE H. N. SHUKLA GROUP OF COLLEGES

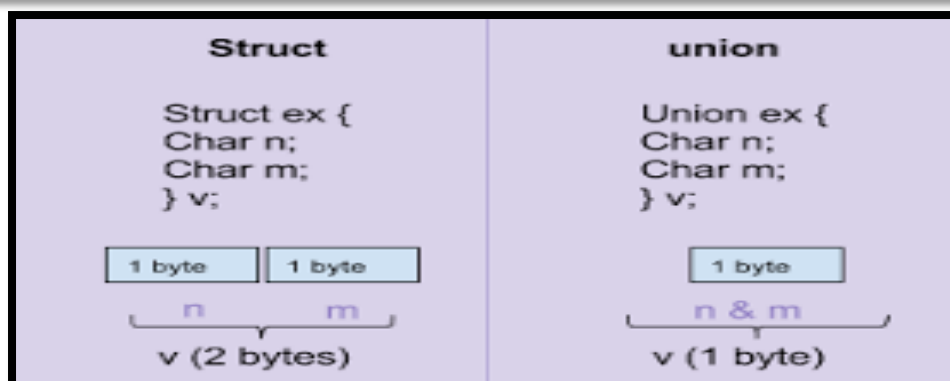
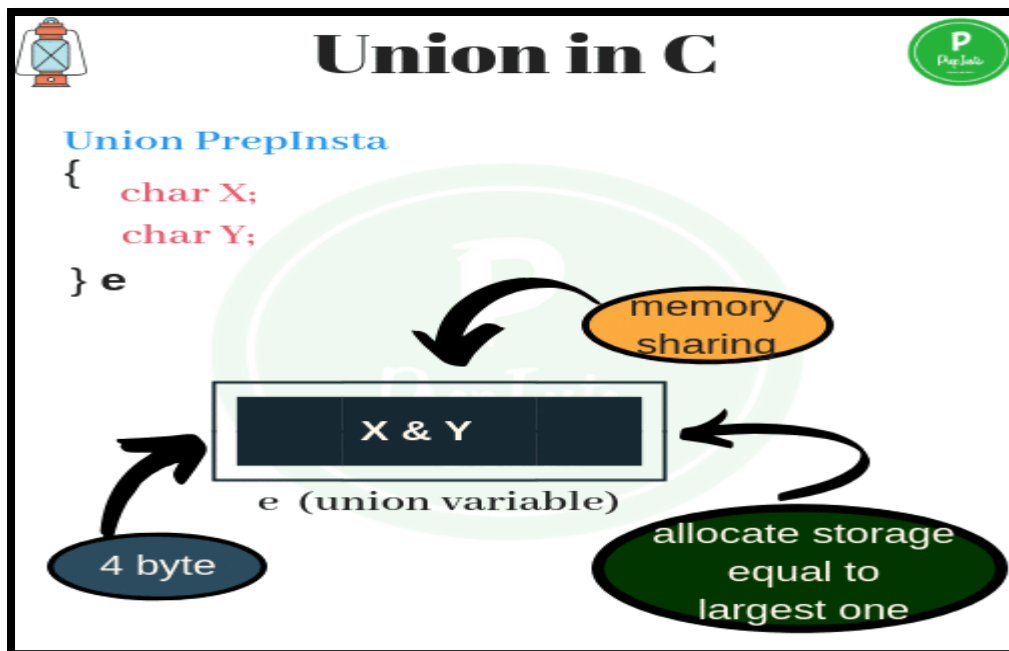
(Affiliated to Saurashtra University & Gujarat Technological University)

SR.NO.	QUESTION	ANSWER
1	Nested structure means _____.	Structure within structure
2	In nested structure , one structure can be declare as _____ of another structure.	Member

Q-11 Briefly Explain Union.

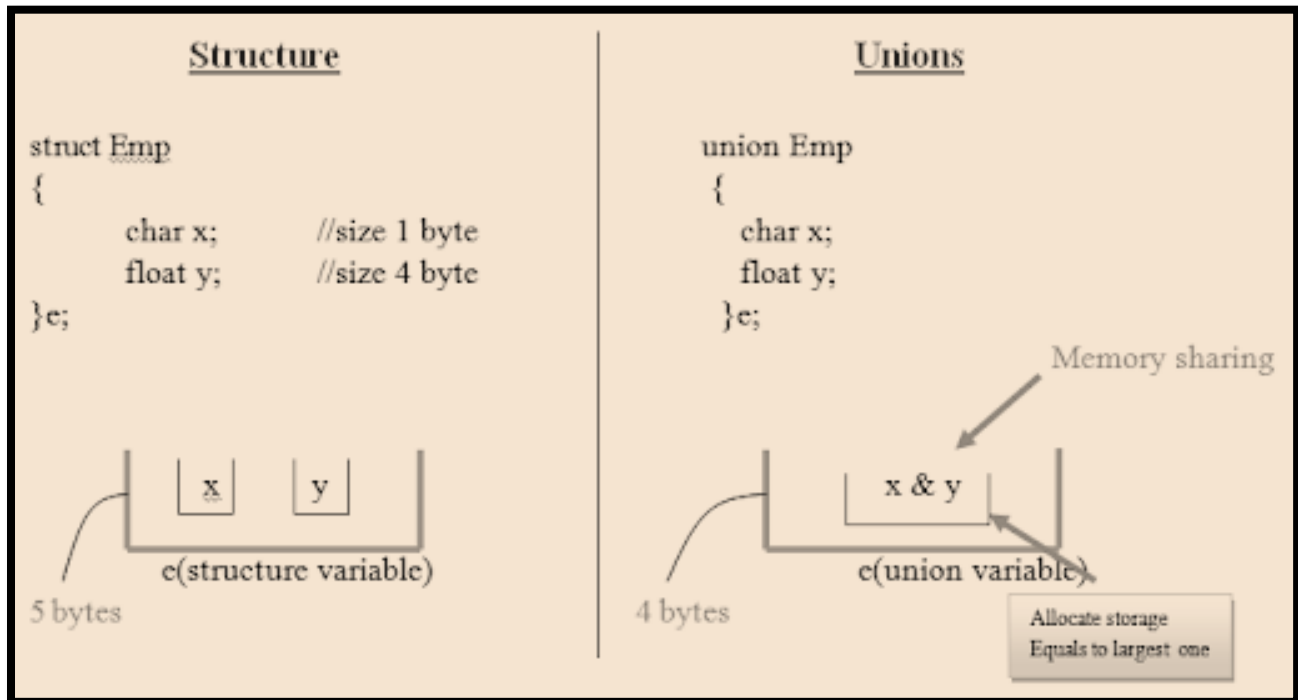
Detail :

- ❖ The concept of union is similar to that of structures but differs in terms of storage space.
- ❖ Union is declared and used in same way as the structures.



SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



Syntax:-

```
keyword      union tag name
↓            ↙
union union-tag
{
    datatype variable1;
    datatype variable2;
    datatype variable3;
};
```

terminating semicolon

Example:-

```
union emp
{
    char c;
    float x;
}b1;
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

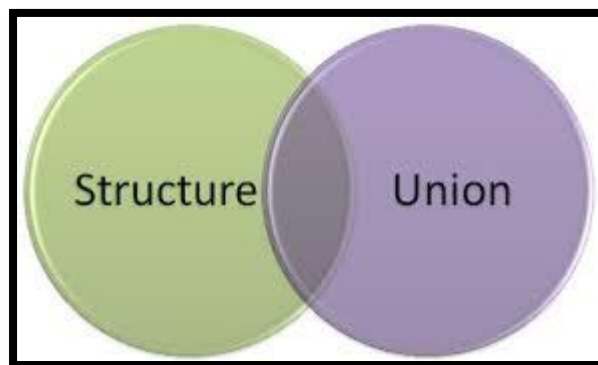
(Affiliated to Saurashtra University & Gujarat Technological University)

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	Union can be declare using _____ keyword.	union
2	All members of union occupies _____ storage location.	Same

Q-12 Give difference between Structure & Union.

Detail :



<u>Structure</u>	<u>Union</u>
1) In structure, each member have its own Storage Space.	1) In Union, All the members can share Same Memory Space.
2) Structure can be declared using "struct" keyword.	2) Union can be declared using "union" keyword.
3) The size of structure depends on length of its elements.	3) The size of union depends on maximum length of integer elements.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

SUMMARY QUESTIONS OF CHAPTER -4

ONE - MARK QUESTIONS

- ❖ What is Array?
- ❖ List out Types of Array.
- ❖ How to declare String Array in C?
- ❖ Which Keyword is used to declare Structure?
- ❖ What is Union?
- ❖ Write down Syntax for declaring Structure.
- ❖ Write down different uses of Array.(OR benefits)

TWO - MARK QUESTIONS

- ❖ Explain Two Dimension Array with Example.
- ❖ Write down syntax for Nested Structure.
- ❖ Explain Array with Structure by Example.

THREE - MARK QUESTIONS

- ❖ Explain Array and Matrices with suitable Example.
- ❖ Write note on Pointer and Structure with Example.
- ❖ Give difference between Structure and Union.

FIVE - MARK QUESTIONS

- ❖ Explain Structure with Example.
- ❖ Explain Union with Example.
- ❖ Write note on UDF with Structure by Example.
- ❖ Explain Array with different types
- ❖ Explain Nested Structure with Example.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

CHAPTER-5

➤ POINTERS

- 7.1 Introduction of Pointers
- 7.2 Use of pointers in dynamic programming
- 7.3 Pointers to variables
- 7.4 Pointer to Array
- 7.5 Pointer within Array
- 7.6 Pointer to structure
- 7.7 Pointer within structure
- 7.8 Pointer to pointer

Q-1 What is pointer ? How to create pointer?
First PROGRAMMING IN C

SHREE H. N. SHUKLA GROUP OF COLLEGES

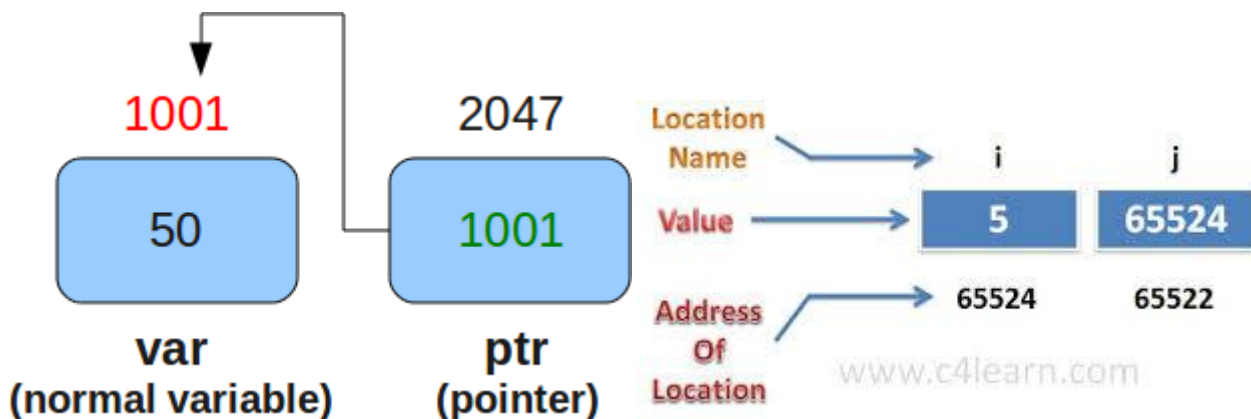
(Affiliated to Saurashtra University & Gujarat Technological University)

WHAT IS POINTER?

- A **pointer** is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before you can use it to store any variable address.
- The general form of a pointer variable declaration is:

dataType *var_name;

- Here,
 - **dataType** is the pointer's base type; it must be a valid C data type (i.e., int, float, char etc).
 - **var_name** is the name of the pointer variable.
 - The asterisk * you used to declare a pointer is the same asterisk that you use for multiplication. However, in this statement the asterisk is being used to designate a variable as a pointer.



Detail :-

- ❖ Pointer is the variable that stores the address of another variable.
- ❖ Pointer is used to point out memory address.
- ❖ The pointer has three main concepts:

1) Pointer constant:-

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- In computer memory, memory addresses are known as pointer constant. One can not change its value, it can only be used.

Example: *house number.*

2) Pointer value:-

- The pointer value means the *address of another variable.*
- We can not access the value of memory address directly. If we want to access then we have to use & operator and * operator

Example: `int a;`

`&a` is known as pointer value.

3) Pointer Variable:-

- The variable which stores the pointer value is known as pointer variable.

Example: `int a;`
`int *P;`
`P=&a;`

P is known as pointer variable.

Example:

```
#include<stdio.h>
#include<conio.h>

void main()
{
    int a;
    int *p;
    clrscr();
    p=&a;
    printf("\n %u",p); //will print address of a
    printf("\n %d",*p); // will print value of a
    getch();
}
```

1 Word Question – Answer

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

SR.NO.	QUESTION	ANSWER
1	_____ is a variable whose value is address of another variable.	pointer
2	Pointer variable can be declare using _____	* (asterisk)
3	The general form of declaring pointer variable is _____	Datatype *<variable name>
4	_____ operator is used to store address of any variable	&

Q-2 What is pointer to variable? Explain with Example.

What is pointer ?

The general form of a pointer variable declaration is

```
type *var_name;
```

Take a look at some of the valid pointer declarations -

```
int *ip; /* pointer to an integer */  
double *dp; /* pointer to a double */  
float *fp; /* pointer to a float */  
char *ch /* pointer to a character */
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Pointer Declaration

- **Type * Varname;**
declares **Varname** as a pointer to **Type**

- **Examples**

```
char* A;  
int* B;  
struct node * Root;
```

A is a pointer to a character

B is a pointer to an integer

Root is a pointer to a node structure

- *** associates with the variable name**

```
char *A, B;
```

A is a pointer to a character

B is a character !

Pg. 9

C Pointer Variables

To declare a pointer variable, we must do two things

- Use the “*” (star) character to indicate that the variable being defined is a pointer type.
- Indicate the type of variable to which the pointer will point (the pointee). This is necessary because C provides operations on pointers (e.g., *, ++, etc) whose meaning depends on the type of the pointee.

- **General declaration of a pointer**

```
type *nameOfPointer;
```

1/14/10

Detail :-

- ❖ Variable means you can store particular value.
- ❖ Pointer is also one type of variable but the main difference between simple variable and pointer variable is that ;
 - ✓ Simple variable can store only the value.
 - ✓ Pointer Variable can store address of another variable.
- ❖ In short, Pointer to variable means a variable that store pointer value.
- ❖ You can declare pointer variable using * (asterisk) and you can store address of pointer using &(ampersand).

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Example:

```
#include<stdio.h>
#include<conio.h>

void main()
{
    int a=10;
    int *p;
    clrscr();
    p=&a;
    printf("\n %u",p); //will print address of a
    printf("\n %d",*p); // will print value of a
    getch();
}
```

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	_____ is a variable whose value is address of another variable.	pointer
2	Pointer variable can be declare using _____	* (asterisk)
3	The general form of declaring pointer variable is_____	Datatype *<variable name>
4	_____ operator is used to store address of any variable	&

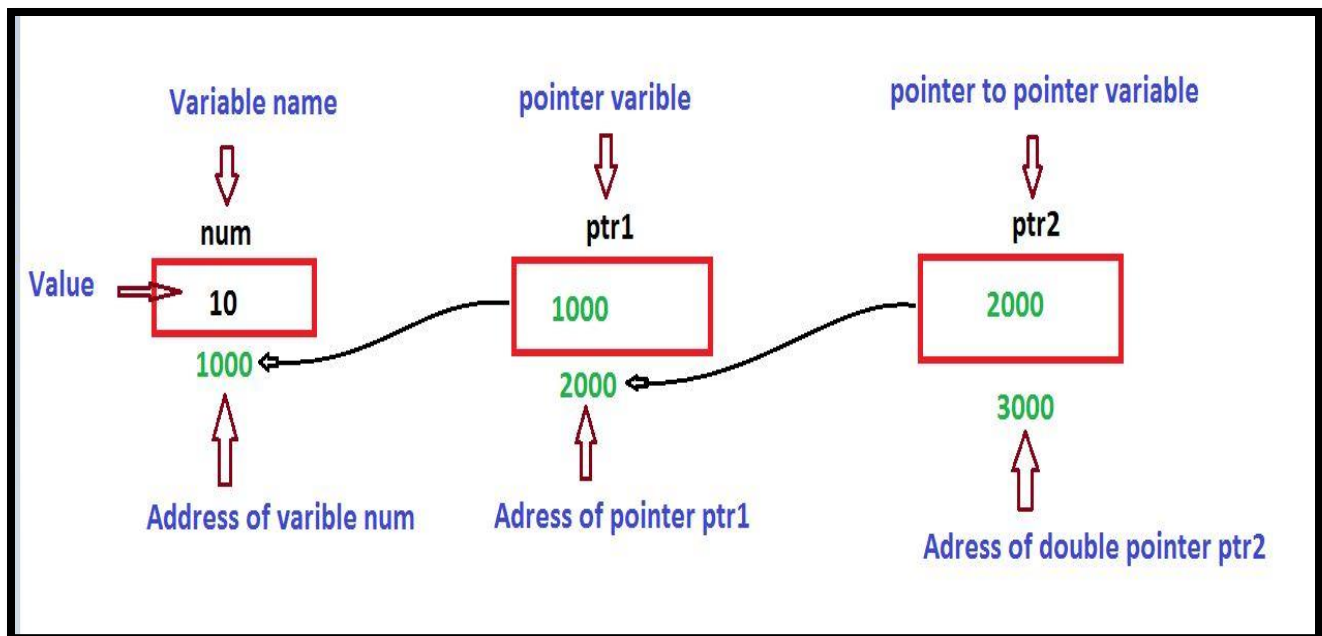
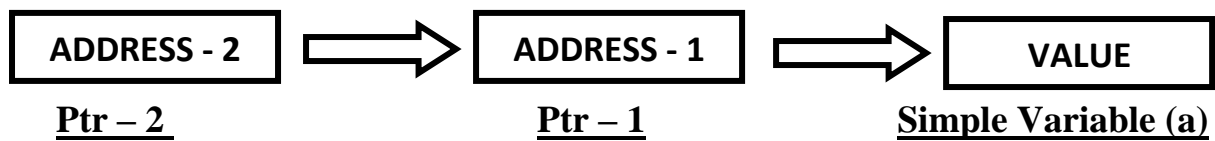
Q-3 What is Pointer to Pointer? Explain with Example.

Detail :-

- ❖ In C Programming it is also possible that pointer variable is pointed by another pointer variable.
- ❖ This process is called pointer to pointer.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



- ❖ Pointer variable ptr-2 having the address of ptr-1 ,same way ptr-1 having the address of simple variable(a).
- ❖ This process is known as multiple indirection.
- ❖ Pointer to pointer variable must be declare using additional indirection operation(**) in front of variable name like ;
Int **p;

Example:

```
#include<stdio.h>
#include<conio.h>

void main()
{
    int a 10;
    int *p;
    int **a;
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
clrscr();
p=&a;
a=&p;
clrscr();
printf("\n value of p = %d",*p); //will print address of a
printf("\n address of p=%d=u",p); // will print value of a
printf("\n value of a = %d",*p);
printf("\n value of **a=%d",**a);
getch();
}
```

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	If pointer variable is pointed by another pointer variable then it is called _____	Pointer to pointer
2	A variable that is pointer to pointer must be declared using _____.	additional indirection (**)
3	_____ operator can be used to access members of the structure using pointer.	-> (arrow)
4	The another pointer variable can contains _____ of first pointer variable.	address

Q-4 What is Pointer to Array? Explain with Example.

Array of Pointer

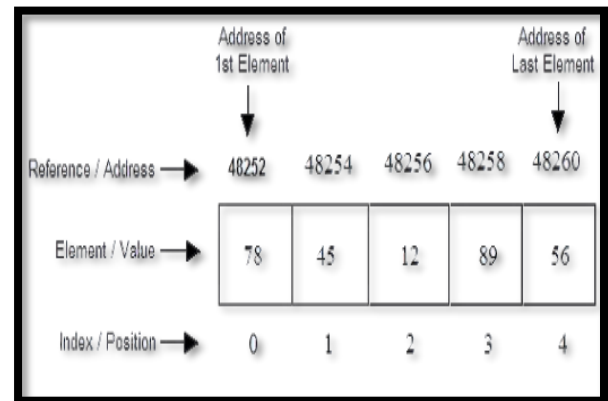
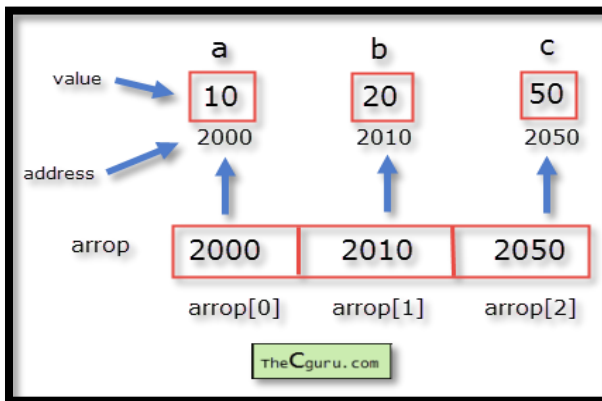
```
int a [ ] = {10,20,30,40};
```

```
int *p[4]; // *p[0], *p[1], *p[2], *p[3]
```

2886728	2886732	2886736	2886740
10	20	30	40
a[0]	a[1]	a[2]	a[3]

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



Arrays of pointers

- More commonly used (by experienced programmers) is an array of pointers.
- We can use an array of pointers in a similar way to a multi-dimensional array.
- We can declare an array of pointers like so:

```
char *name[] = {"Dave", "Bert", "Alf"};  
/* Creates and initialises 3 names */
```

We can now use `name[0]` anywhere we could use a string.



docsity.com

Detail :-

- ❖ When array is declared, compiler decides the memory to store array elements.
- ❖ Pointer to array is used to store whole array into single pointer variable.
- ❖ We can declare pointer to array like following :

```
int a[5];  
int *p;  
p=a;  
p=&a[0];
```

- ❖ For Example, you can store each element of array which can be pointed out by pointer.
- ❖ The array elements using pointer can be stored like following ;

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Elements	10	20	30	40	50
Array Index	a[0]	a[1]	a[2]	a[3]	a[4]
Pointer	p	p+1	p+2	p+3	p+4
Address	1000	1002	1004	1006	1008

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	_____ is used to point to zero element of the array using pointer.	P=&a[0]
2	If p point to particular element of array then _____ represent next element of array.	P+1

Q-5 What is Pointer to Structure? Explain with Example.

Pointer to Structure

- A structure type pointer variable can be declared as:

```

struct book
{
    char name[20];
    int pages;
    float price;
};
struct book *bptr;
    
```

- However, this declaration for a pointer to structure does not allocate any memory for a structure but allocates only for a pointer, so that to access structure's members through pointer **bptr**, we must allocate the memory using **malloc()** function.
- Now, individual structure members are accessed as:

```

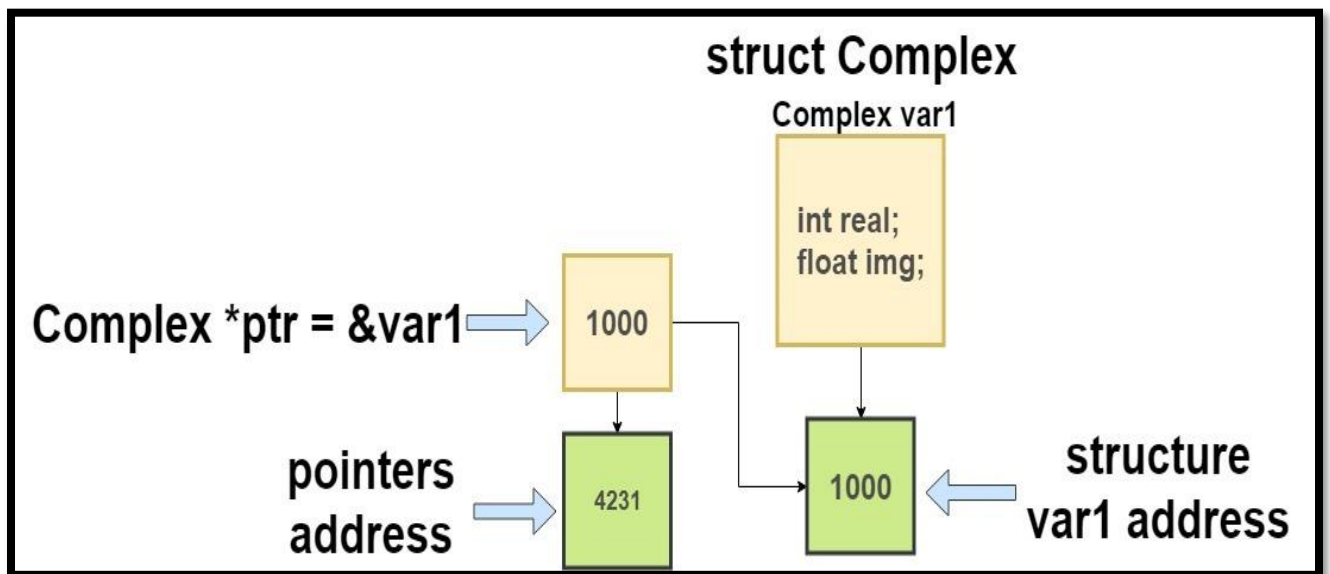
bptr->name      bptr->pages      bptr->price
OR
(*bptr).name    (*bptr).pages    (*bptr).price
    
```

- Here, -> is called arrow operator and there must be a pointer to the structure on the left side of this operator.

35

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



Detail :-

- ❖ Structure is a data structure that support group of elements that can be of different data types.
- ❖ We can easily use pointer with the structure.
- ❖ To access elements of structure (members) ->(arrow) operator can be used.
- ❖ Pointer to structure can allocate memory dynamically.
- ❖ Pointer must be declare before it is used.

Example:

```
#include<stdio.h>
#include<conio.h>
Struct stud
{
    Int rno;
    Char nm[20];
};

void main()
{
    Struct stud *p;
    P=&s;
```

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

```
Clrscr();  
s.rno=1;  
printf("rno=%d",p->rno);  
getch();  
}
```

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	The general form of defining structure to pointer is _____	Struct stud *p
2	Pointer to structure can also allocate memory _____	Dynamically
3	_____ operator can be used to access members of the structure using pointer.	-> (arrow)

Q-6 Explain Memory Allocation of Structure.

Detail :-

- ❖ The total memory occupied by structure can be the total size of each member.
- ❖ We can also use size of operator to get memory allocation of structure.
- ❖ For example : consider that we have following structure:

Struct student

```
{  
  
    Int rno;  
  
    Float salary;  
  
    Char nm[20];  
  
};
```

- ❖ In above structure it have 3 members:
Int rno ; - 2 bytes
Float salary ; - 4 bytes
Char nm[20] ; - 20 bytes

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- ❖ Total 26 bytes will be occupied by structure student in the memory.

Q-7 What is Data file? [1 mark]

Data File

- ▶ A **data file** is a computer file which stores data for use by a computer application or system.
- ▶ It generally does *not* refer to files that contain instructions or code to be executed (typically called program files), or to files which define the operation or structure of an application or system (which include configuration files, directory files, etc.);
- ▶ but to files that specifically contain information used as input, and/or written as output by some other software program.

Files Streams in C programming

Data files

- Can be created, updated, and processed by C programs
- Storage of data in variables and arrays is only temporary

Data files are of two types

TEXT FILES	BINARY FILES
Saves the data in the form of ASCII codes It consist of sequential characters divided into lines. Each line terminates with the newline character (<code>\n</code>).	Saves the data in Binary codes It consist of data values such as integers, floats or complex data types, "using their memory representation."

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :-

- ❖ Data file is used to store amount of informations permanently.
- ❖ Data file can be of 2 types:
 - (1) Text file (.txt)
 - (2) Binary file(.obj , .exe)

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	The types of Data file can be of _____ & _____ .	Text file & Binary file
2	The extension of binary file may be _____	.exe or .obj
3	Textfile saves data in the form of _____	ASCII Code
4	The End of the file can be indicate using _____ character.	NULL ('\0')

Q-8 What is File Handling?[1 mark]

Detail :-

- ❖ The process of accessing file from particular program is known as file handling.
- ❖ File handling also known as I/O (Input / Output).
- ❖ File handling include following operations :-
 - Naming the file
 - Opening the file
 - Reading from the file
 - Writing to the file
 - Closing file

Files

- File – place on disk where group of related data is stored
 - E.g. your C programs, executables

- High-level programming languages support file operations
 - Naming
 - Opening
 - Reading
 - Writing
 - Closing

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

C File Handling

1. Opening – FOPEN("filename", "mode");
2. Closing – FLCLOSE(FP);
3. File Creation – Use File Modes "w", "w+", "a" and "a+"
4. File Reading – FGETC, FGETS, FSCANF
5. File Writing – FPUTC, FPUTS, FPRINTF
6. File Seeking – FSEEK

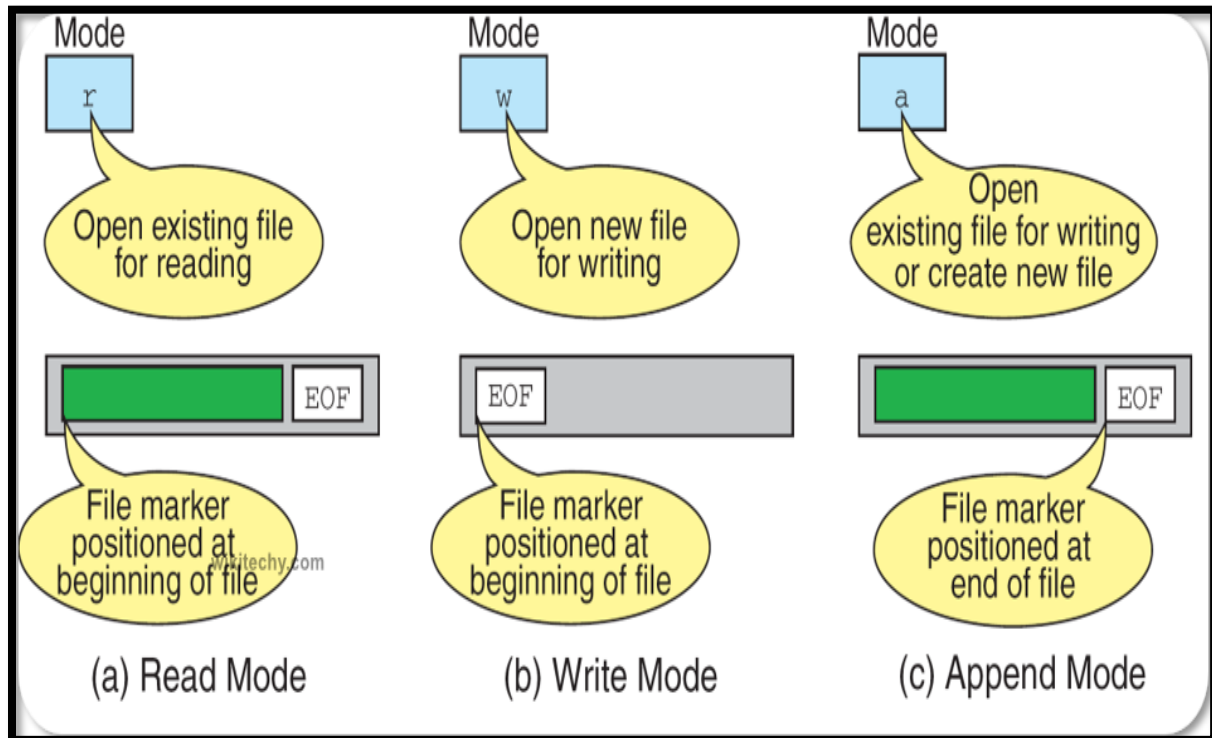
1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1		Text file & Binary file
2	To write data to the particular file , the file must be open in _____ mode.	Write (w)
3	_____ is used to indicate end of the file	feof()
4	_____ is used to set the file position to the beginning of the file.	Rewind()
5	_____ function is used to return current file position.	Ftell()
6	_____ is used to return file position to given location of the file.	Fseek()
7	_____ is used to flushes all the file streams.	Fflush()

Q-9 Explain different file modes.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)



File Opening Modes

Mode	Meaning	Description
r	Read	Only reading possible. Not create file if not exist
w	Write	Only writing possible. Create file if not exist otherwise erase the old content of file and open as a blank file
a	Append	Only writing possible. Create file if not exist, otherwise open file and write from the end of file (do not erase the old content)
r+	Reading + Writing	R & W possible. Create file if not exist. Overwriting existing data. Used for modifying content
w+	Reading + Writing	R & W possible. Create file if not exist. Erase old content.
a+	Reading + Appending	R & W possible. Create file if not exist. Append content at the end of file

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Detail :-

- ❖ When user want to open any file , at that time following file opening modes are available for different purpose.

(1) “r” (read mode) :-

- This mode search particular file.
- If the file exist then load into the memory and read data from it.
- If file does not exist then it will return “null”.

(2) “w” (write mode) :-

- This mode search particular file.
- If the file exist then load into the memory and write data to it.
- If file does exist then it will create new file automatically.

(3) “a” (append mode) :-

- This mode search particular file.
- If file exist then load into the memory and append it.
- Otherwise it will create new file automatically.

(4) “r+” (Read & Write mode) :-

- This mode search particular file.
- If file exist then load into memory and allow you to read data first and write new data.
- Otherwise it will create new file automatically.

(5) “w+” (Write & Read mode) :-

- This mode search particular file.
- If file exist then load into the memory and allow you to write new data first and then read it.
- Otherwise it will create new file automatically.

(6) “a+” (Read , Write ,append & modify) :-

- This mode search particular file.
- If file exist then load into the memory and allow you to read , write ,append and modify.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- Otherwise it will create new file automatically.
- **1 Word Question – Answer**

SR.NO.	QUESTION	ANSWER
1		Text file & Binary file
2	To write data to the particular file , the file must be open in _____ mode.	Write (w)
3	_____ is used to indicate end of the file	feof()
4	_____ is used to set the file position to the beginning of the file.	Rewind()
5	_____ function is used to return current file position.	Ftell()
6	_____ is used to return file position to given location of the file.	Fseek()
7	_____ is used to flushes all the file streams.	Fflush()

Q-10 Explain different file handling functions.

Detail :-

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

File operation functions in C:

Function Name	Operation
fopen()	Creates a new file for use Opens a new existing file for use
fclose	Closes a file which has been opened for use
getc()	Reads a character from a file
putc()	Writes a character to a file
fprintf()	Writes a set of data values to a file
fscanf()	Reads a set of data values from a file
getw()	Reads a integer from a file
putw()	Writes an integer to the file
fseek()	Sets the position to a desired point in the file
ftell()	Gives the current position in the file
rewind()	Sets the position to the beginning of the file

❖ File handling support mainly following functions.

(1) Fopen() :-

- It is used to open any particular file.
Syntax :- fopen ("filename" , "filemode")
- In first argument , we have to provide file name.
- In second argument , we have to provide file modes like read , write and append.

Example :- fopen("c:\\hi.txt","r")

(2) Fclose() :-

- It is used to close currently opened file.

Syntax :- fclose(<file pointer>)

Example :- fclose(fp)

(3) Fprintf() :-

- It is used to print particular value to the file.
- It have three arguments :

File pointer

Control String

variable

Syntax :- fprintf ("filepointer" , "control string",variable)

Example :- fprintf(fp,"%c",a)

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

(4) Fscanf() :-

- It is used to read value for standard input.
- It have three arguments :
File pointer
Control String
variable
Syntax :- fscanf ("filepointer" , "control string",variable)
Example :- fscanf(fp,"%c",a)

(5) getw() :-

- It is used to get or read an integer value. [only work with integer data]
Syntax :- getw(<file pointer>)
Example :- getw(fp)

(6) putw() :-

- It is used to write an integer value to the file. [only work with integer data]
Syntax :- putw(<int>,<file pointer>)
Example :- putw(num,fp)

(7) fseek() :-

- It is used to set file position means that it is used to move file position to given location in the file.
Syntax :- fseek(<file pointer> , <off set> , <position>)
Example :- fseek(fp , s,seek_set)

0- SEEK_SET -> beginning of the file

1- SEET_CUR -> Current position

2- SEEK_END -> end of the file

(8) Rewind() :-

- It is used to set file position to beginning of the file or starting of the file.
Syntax :- rewind(<file pointer>)
Example :- rewind(fp)

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

(9) Ftell() :-

- It is used to return current file position of given file.
Syntax :- ftell(<file pointer>)
Example :- ftell(fp)

(10)Remove() :-

- It is used to remove particular file by providing file name.
Syntax :- remove(<file name>)
Example :- remove("myfile.txt")

(11)Feof() :-

- It is used to indicate end of the file.
Syntax :- feof(<file pointer>)
Example :- feof(fp)

(12)Rename() :-

- It is used to change from old file name to new file name.
Syntax :- int rename (char * old file name ,char * new file name)
Example :- rename("abc.txt" , "jkl.txt")

(13)Ferror() :-

- It is used to check if the file opened successfully or not , otherwise it returns error.
Syntax :- ferror(< file pointer>)
Example :- ferror(fp)

(14)Fflush() :-

- It is used to flush the stream,It is used to clean input buffer as well as output buffer.
Syntax :- fflush(< file pointer>)
Example :- fflush(fp)

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
--------	----------	--------

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

1		Text file & Binary file
2	To write data to the particular file , the file must be open in _____ mode.	Write (w)
3	_____ is used to indicate end of the file	feof()
4	_____ is used to set the file position to the beginning of the file.	Rewind()
5	_____ function is used to return current file position.	Ftell()
6	_____ is used to return file position to given location of the file.	Fseek()
7	_____ is used to flushes all the file streams.	Fflush()

Q-11 Explain different I/O operations.

Detail :-

- ❖ I/O stands for input and output.
- ❖ It support following functions for input(read) & output(write).

INPUT	OUTPUT
Fread()	Fwrite()
Fgets()	Fputs()
Fgetc()	Fputc()
Fscanf()	Fprintf()

❖ INPUT FUNCTIONS :-

(1)Fgets() :-

- It is used to get a string.
- It have three arguments

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Address of String

Maximum length

File pointer

Syntax :- fgets(char *s , int n , <file pointer>)

Example :- fgets(s,80,fp)

(2)Fgetc() :-

- It is used to get a single character from the file.
- It have two arguments

Character variable

File pointer

Syntax :- fgetc(<char variable> , <file pointer>)

Example :- fgetc(ch , fp)

(3)Fread() :-

- It is same as fscanf() , but fread() also read data from binary file.
- It have four arguments

Address of variable

Size of data

Number of values to be read

File pointer

Syntax :- fread(v,s,n,fp)

Reading data from a File

There are three different functions dedicated to reading data from a file

- **fgetc(file_pointer):** It returns the next character from the file pointed to by the file pointer. When the end of the file has been reached, the EOF is sent back.
- **fgets(buffer, n, file_pointer):** It reads n-1 characters from the file and stores the string in a buffer in which the NULL character '\0' is appended as the last character.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

- **fscanf(file_pointer, conversion_specifiers, variable_addresses):** It is used to parse and analyze data. It reads characters from the file and assigns the input to a list of variable pointers variable_addresses using conversion specifiers. Keep in mind that as with scanf, fscanf stops reading a string when space or newline is encountered.

❖ OUTPUT FUNCTIONS :-

(1) fputs() :-

- It is used to print a string to the file.
- It has two arguments

String variable

File pointer

Syntax :- fputs(<string variable> , <file pointer>)

Example :- fputs(str,fp)

(2) fputc() :-

- It is used to print a single character to the file.
- It has two arguments

Character variable

File pointer

Syntax :- fputc(<character variable> , <file pointer>)

Example :- fputc(ch,fp)

(3) fwrite() :-

- It is same as fprintf() , but the difference is fwrite() can also write binary data.
- It has four arguments
 - Address of variable
 - Size of data
 - Number of values to be read
 - File pointer

Syntax :- fwrite(v,s,n,fp)

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Writing to a File

In C, when you write to a file, newline characters '\n' must be explicitly added.

The stdio library offers the necessary functions to write to a file:

- **fputc(char, file_pointer)**: It writes a character to the file pointed to by file_pointer.
- **fputs(str, file_pointer)**: It writes a string to the file pointed to by file_pointer.
- **fprintf(file_pointer, str, variable_lists)**: It prints a string to the file pointed to by file_pointer. The string can optionally include format specifiers and a list of variables variable_lists.

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	_____ is used to write character to the file pointed by file pointer	Fputc()
2	_____ is used to read whole string from particular file.	Fgets()
3	_____ is used to print different values to the file.	Fprintf()
4	_____ is used to perform I/O operations for binary file.	Fwrite()

Q-12 Explain Command Line Arguments with suitable example.

Command Line Arguments (Cont...)

- It should be noted that **argv[0]** holds the name of the program itself and **argv[1]** is a pointer to the first command line argument supplied, and *argv[n] is the last argument.
- If no arguments are supplied, argc will be one, and if you pass one argument then **argc** is set at 2.
- E.g. F:\C\L04>CommandLineArguments.exe test

```
argv  CommandLineArguments.exe  test
```

argc is 2 here

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

Command line arguments

argument count

array of string arguments

```
#include <stdio.h>
int main(int argc, char *argv[]) {
}
```

argv[0] is the program name with full path

argv[1] is the first argument

argv[2] is the second argument, etc.

Detail :-

- ❖ Command line arguments are the argument that are pass when the program is execute.
- ❖ In command line argument , main() also take argument from the user.
- ❖ Main() take two arguments
 - (i) Argc – It indicate number of arguments to be passed
 - (ii) Argv[] – It is pointer array which point to each argument that is passed with the main().

(iii) main() takes two arguments:

- i. **argc** → Number of arguments passed
- ii. **argv[]** → pointer array which points to each argument which is passed to main().

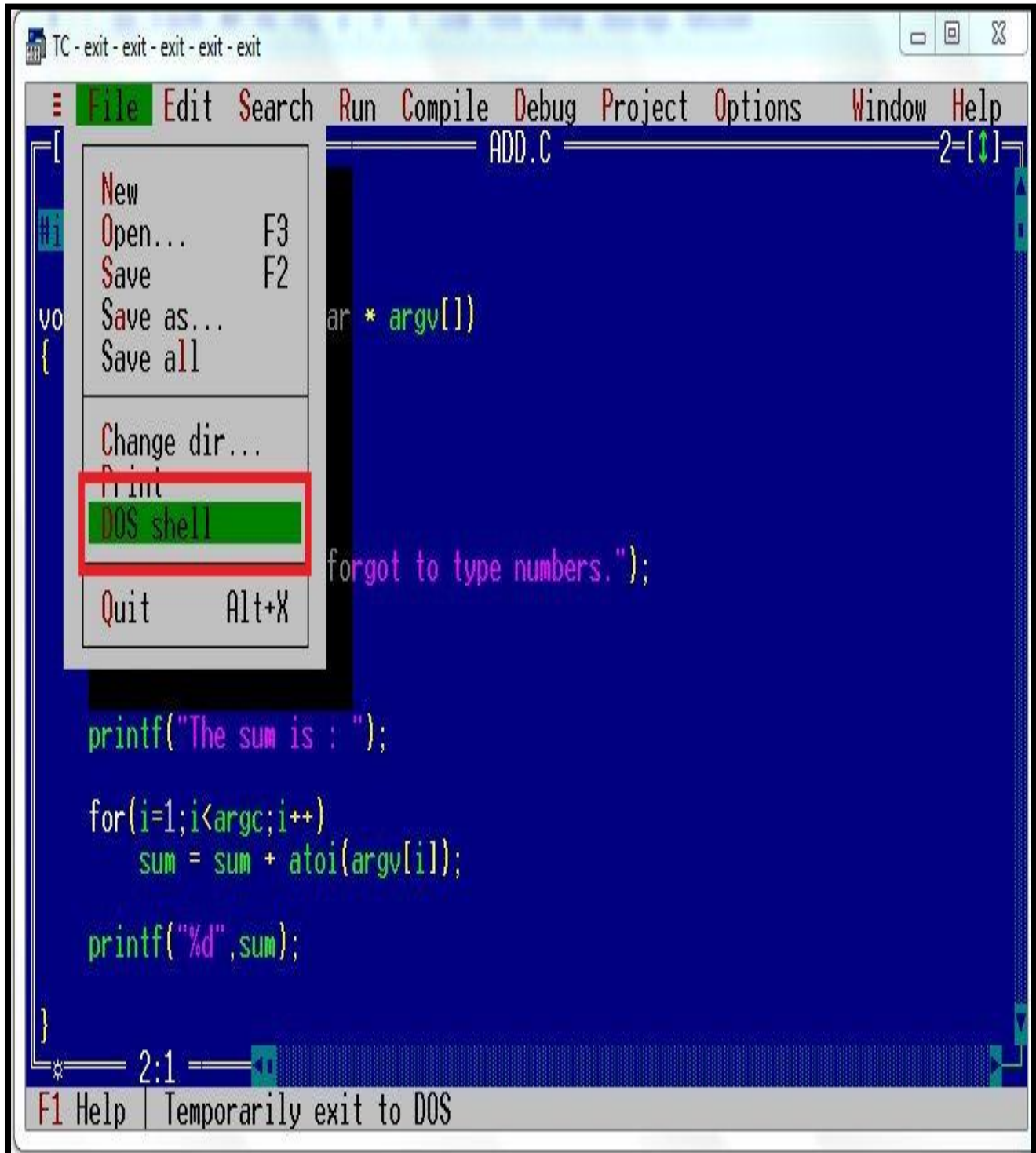
Note:- argv[0] → name of the program that is exevuted.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

argv[1] → First argument

argv[2] → Second argument



The screenshot shows the Turbo C++ IDE window titled "TC - exit - exit - exit - exit - exit". The menu bar includes File, Edit, Search, Run, Compile, Debug, Project, Options, Window, and Help. The "File" menu is open, showing options like New, Open..., Save, Save as..., Save all, Change dir..., Print, DOS shell (highlighted with a red box), and Quit. The main editor area displays the following C code:

```
ADD.C
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char * argv[])
{
    int sum = 0;
    if(argc < 2)
        printf("You forgot to type numbers.");

    printf("The sum is : ");

    for(i=1; i<argc; i++)
        sum = sum + atoi(argv[i]);

    printf("%d", sum);
}
```

The status bar at the bottom shows "F1 Help" and "Temporarily exit to DOS".

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

1 Word Question – Answer

SR.NO.	QUESTION	ANSWER
1	In Command Line Arguments main () takes _____ arguments.	2(Two)
2	argc stands for _____	Argument count
3	_____ is pointer array which points to each argument which is passed to main().	Argv[]
4	_____ holds name of the program	Argv[0]
5	To execute command line program , go to file menu and select _____	Dos Shell

Q-13 Give difference between Text file and Binary file.

Detail :-

<u>Text File</u>	<u>Binary File</u>
<ol style="list-style-type: none">1) Text file is the file created by user.2) The extension of text file may be .txt, .c etc.3) In text file, there is special character ('\0') which indicates end of file.4) In text file, data is stored in text format, so if we have integer value 1234 then it will require 5 bytes (1234\0)	<ol style="list-style-type: none">1) Binary file is the file created by the computer.2) The extension of binary file may be .exe, .obj etc.3) In binary file there is no such special character that indicates end of file.4) In binary file, data is stored in the format of main memory. So if we have integer value 1234 then it will require only 2 bytes.

SHREE H. N. SHUKLA GROUP OF COLLEGES

(Affiliated to Saurashtra University & Gujarat Technological University)

SUMMARY QUESTIONS OF CHAPTER -5

ONE - MARK QUESTIONS

- ❖ What is Pointer?
- ❖ List out uses of Pointer.(Advantages)
- ❖ Which operator is used to represent Pointer to Pointer.
- ❖ How to declare Pointer to Structure?
- ❖ What is Data File?
- ❖ How to Declare Pointer Variable?

TWO - MARK QUESTIONS

- ❖ Explain Pointer to Structure with Example.
- ❖ Write note on Pointer to Array.
- ❖ Explain Pointer within Structure by Example.
- ❖ Explain Pointer to Pointer in brief.

THREE - MARK QUESTIONS

- ❖ Explain Command Line Argument with Example.
- ❖ Explain Fopen() , Fclose() and Fseek() with Example.
- ❖ Explain any three I/O operations functions with Example.

FIVE - MARK QUESTIONS

- ❖ Explain File Handling in Detail.
- ❖ List out different File Handling Functions and explain any five with Example.
- ❖ Explain Pointer with reference to Dynamic Programming.
- ❖ Explain Pointer within Array with Example.