

The code is life. Never mess with the code.



Chirag Ferwani

## PREFACE

**C is a general-purpose computer programming language. It was created in the 1970s by Dennis Ritchie and remains very widely used and influential. By design, C's features cleanly reflect the capabilities of the targeted CPUs.**

**C is an imperative procedural language supporting structured programming, lexical variable scope, and recursion, with a static type system. It was designed to be compiled to provide low-level access to memory and language constructs that map efficiently to machine instructions, all with minimal runtime support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant C program written with portability in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code. This book written by Chirag Ferwani, states the importance of programming in C in terms of both theory and programming.**

## ORGANISATION

**Error by Night is a pace where technophiles turn to get higher rankings and more traffic. It is all about teaching you about coding in Python | Java | C, web development, web hosting, and a lot more tech solutions.**

**For more information regarding programming languages | software | web development | projects visit the website: <https://chiragf27.github.io/errorbynigh.org/>**

# **THEORY**

## INDEX

### THEORY

- |                                          |      |
|------------------------------------------|------|
| 1. CHAPTER:1 - PROGRAM LOGIC DEVELOPMENT | [1]  |
| 2. CHAPTER:2 - BASIC CONCEPTS OF C       | [5]  |
| 3. CHAPTER:3 - CONTROL STRUCTURE         | [10] |
| 4. CHAPTER:4 - ARRAY AND STRUCTURE       | [14] |
| 5. CHAPTER:5 - FUNCTIONS                 | [18] |
| 6. CHAPTER:6 - POINTERS                  | [21] |

### PROGRAMMING

- |                       |      |
|-----------------------|------|
| 1. CH:1 - CH:3        | [23] |
| 2. ARRAY              | [40] |
| 3. STRING             | [50] |
| 4. STRUCTURE          | [57] |
| 5. ARRAY OF STRUCTURE | [60] |
| 6. NESTED STRUCTURE   | [62] |
| 7. FUNCTIONS          | [64] |
| 8. RECURSION          | [72] |
| 9. POINTERS           | [75] |

EXTRA QUESTIONS [82]

## CHAPTER: 1 – PROGRAM LOGIC DEVELOPMENT

### **1. What is a problem?**

**A problem is a situation or condition which needs to solve to achieve the goal.**

### **2. What are the steps in problem solving?**

- Define the problem
- Data gathering
- Decide the effective solution
- Implement and evaluate the solution
- Review the results

### **3. What are the two problem solving techniques?**

- Algorithm
- Flowchart

### **4. What is an Algorithm? ®**

**An algorithm is a set of ordered instructions written in simple English Language.**

**It defines step by step logic for a program to solve the problem.**

### **5. What are the characteristics of Algorithm?**

- Input – an algorithm should accept one or more input
- Output – an algorithm should generate one or more output
- Definiteness – each instruction should be clear
- Effectiveness – each instruction should have a proper meaning
- Finiteness – the algorithm must terminate after fixed number of steps

### **6. What are the conventions used in algorithm?**

- 1) Algorithm name
- 2) Comments
- 3) Input & Output
- 4) Steps
- 5) Variables
- 6) Assignment statement
- 7) If statement
- 8) Repeat statement
- 9) End of algorithm

### **7. What is a Flowchart?**

**Flowchart is a diagram created by different shapes to show the flow of data. It is the graphical representation of algorithm.**

### **8. Difference between Algorithm and Flowchart?**

Algorithm	Flowchart
1) Algorithm is a step by step procedure to solve the problem	1) Flowchart is a diagram created by different shapes to show the flow of data.
2) Algorithm is complex to understand	2) Flowchart is easy to understand.

3) Algorithm uses plain text	3) Flowchart uses symbols
4) Algorithm is a pseudo code for the program	4) Flowchart is a graphical representation of that logic
5) Algorithm is easy to debug	5) Flowchart is hard to debug

9. State the symbols used in Flowchart? ® ® ®

Symbol	Operation	Meaning
	Start/Stop	Represents the start/stop of algorithm.
	Input/output	Represents value to be given by the end user and the values to be displayed.
	Processing	Represents arithmetic operations.
	Decision making	Represents logical checking to decide flow of sequence.
	Off page connection	Represents continuing of flowchart to another page
	Looping	Represents looping which is based on condition or value of variable.
	Connection (on page connection)	Represents continuing of flowchart to another place on the same page
	Flow lines	Represents the direction of flow.

10. Write the advantages of Algorithm and Flowchart? ®

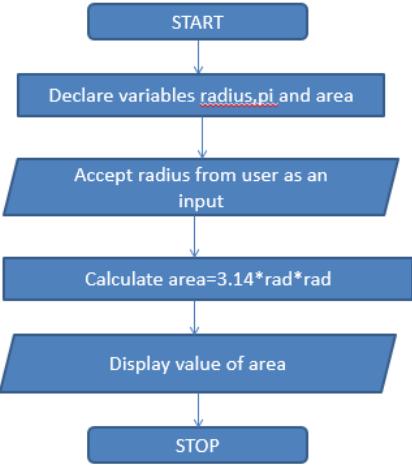
**Algorithm:-**

- 1) It defines step by step logic for the program.
- 2) By using algorithm decision making becomes a more rational process.
- 3) Algorithm allows separation of procedure which increases efficiency.
- 4) Algorithms are often reusable.

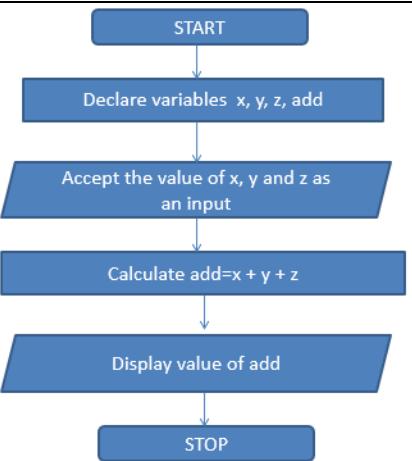
**Flowchart:-**

- 1) Flowcharts are the graphical representation of algorithm.
- 2) The problem can be analyzed in a more effective way.
- 3) It helps in debugging process
- 4) Maintenance of program is easy with the help of flowchart.

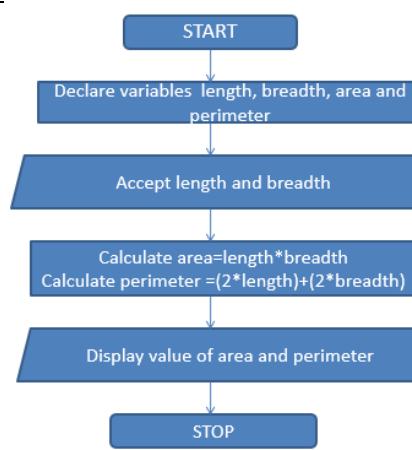
**11. Write an algorithm and draw the flowchart to calculate area of a circle**

Algorithm	Flowchart
<b>Name : calculate area of circle</b> <b>Input: Radius of circle</b> <b>Output: Area of circle</b> <b>Step1: Start</b> <b>Step2: Declare variables pi=3.14, radius and area</b> <b>Step3: Read values for radius //accept the radius value</b> <b>Step4: Calculate area of circle and assign the result to area</b> <b>Step 5: Display value of area</b> <b>Step6: Stop</b>	 <pre> graph TD     START([START]) --&gt; Declare[Declare variables radius,pi and area]     Declare --&gt; Accept[Accept radius from user as an input]     Accept --&gt; Calculate[Calculate area=3.14*rad*rad]     Calculate --&gt; Display[Display value of area]     Display --&gt; STOP([STOP]) </pre>

**12. Write an algorithm to find addition of 3 numbers®**

Algorithm	Flowchart
<b>Name Algorithm: addition of 3 numbers</b> <b>Input: 3 numbers</b> <b>Output: addition</b> <b>Step1: Start</b> <b>Step2: Declare variables x, y, z, add</b> <b>Step3: Read values for x, y, z</b> <b>Step4: Calculate addition of three numbers and assign the result to add. Or add= x+y+z</b> <b>Step 5: Display value of add</b> <b>Step6: Stop</b>	 <pre> graph TD     START([START]) --&gt; Declare[Declare variables x, y, z, add]     Declare --&gt; Accept[Accept the value of x, y and z as an input]     Accept --&gt; Calculate[Calculate add=x + y + z]     Calculate --&gt; Display[Display value of add]     Display --&gt; STOP([STOP]) </pre>

**13. Write an algorithm and draw the flowchart to calculate and display the area and perimeter of a rectangle**

Algorithm	Flowchart
<b>Name: area and perimeter of rectangle</b> <b>Input: length and breadth</b> <b>Output: area and perimeter</b> <b>Step1: Start</b> <b>Step2: Declare variables l, b, area, peri</b> <b>Step3: Read values for l and b</b> <b>Step4: Calculate area and perimeter by: area= length*breadth</b> <b>perimeter =(2*length)+(2*breadth)</b> <b>Step 5: Display value of area and perimeter</b> <b>Step6: Stop</b>	 <pre> graph TD     START([START]) --&gt; Declare[Declare variables length, breadth, area and perimeter]     Declare --&gt; Accept[Accept length and breadth]     Accept --&gt; Calculate[Calculate area=length*breadth Calculate perimeter =(2*length)+(2*breadth)]     Calculate --&gt; Display[Display value of area and perimeter]     Display --&gt; STOP([STOP]) </pre>

**14. Write an algorithm to swap the values of two numbers**

Algorithm	Flowchart
<p><b>Name Algorithm:</b> swap the values of two numbers</p> <p><b>Input:</b> 2 numbers</p> <p><b>Output:</b> exchanged values</p> <p><b>Step1:</b> Start</p> <p><b>Step2:</b> Declare variables a, b, temp</p> <p><b>Step3:</b> Read values for a, b</p> <p><b>Step4:</b> Swap the values of a and b by:</p> <pre>temp = a; a = b; b = temp;</pre> <p><b>Step 5:</b> Display value of swapped numbers</p> <p><b>Step6:</b> Stop</p>	<pre> graph TD     START([START]) --&gt; Declare[Declare variables a, b, temp]     Declare --&gt; Accept[/Accept value of a and b/]     Accept --&gt; Swap[temp=a a=b b= temp]     Swap --&gt; Display[/Display changed value of a and b/]     Display --&gt; STOP([STOP])   </pre>

**15. Write an algorithm to find greater of two numbers**

Algorithm	Flowchart
<p><b>Name of Algorithm :</b> Greater number</p> <p><b>Input:</b> Two numbers</p> <p><b>Output:</b> The greater number among two numbers</p> <p><b>Step1:</b> Start</p> <p><b>Step2:</b> Declare variables no1,no2</p> <p><b>Step3:</b> Read values for no1,no2</p> <p><b>Step4:</b> if no1&gt;no2</p> <p style="padding-left: 20px;">Display no1 is greater number</p> <p style="padding-left: 20px;">else</p> <p style="padding-left: 20px;">Display no2 is greater number</p> <p><b>Step5:</b> Stop</p>	<pre> graph TD     Start([Start]) --&gt; Declare[Declare variables no1 and no2]     Declare --&gt; Accept[/Accept values for no1 and no2/]     Accept --&gt; Decision{If(no1&gt;no2)}     Decision -- False --&gt; No2[Display message no2 is greater]     Decision -- True --&gt; No1[Display message no1 is greater]     No1 --&gt; Stop([Stop])     No2 --&gt; Stop   </pre>

**16. Write an algorithm to accept two numbers and find its product.**

Algorithm	Flowchart
<p><b>Name Algorithm:</b> Product of 2 numbers</p> <p><b>Input:</b> 2 numbers</p> <p><b>Output:</b> Product</p> <p><b>Step1:</b>Start</p> <p><b>Step2:</b>Declare variables no1, no2, product</p> <p><b>Step3:</b> Read values for no1 and no2</p> <p><b>Step4:</b> Calculate product of two numbers and assign the result to product.</p> <p>Or product= no1*no2</p> <p><b>Step 5:</b>Display value of product</p> <p><b>Step6:</b>Stop</p>	<pre> graph TD     start([start]) --&gt; Declare[Declare variables no1,no2,product]     Declare --&gt; Accept[/Accept the value of no1,no2/]     Accept --&gt; Product[Product=no1*no2]     Product --&gt; Display[/Display the value of product/]     Display --&gt; STOP([Stop])   </pre>

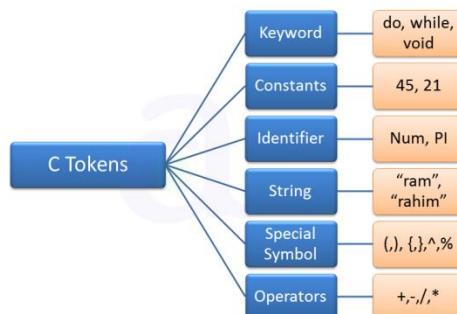
## CHAPTER: 2 – BASIC CONCEPTS OF C

### 1. What is a character Set? ®

Any alphabet(A-Z,a-z),digit(0-9) or special symbol used to represent the data.

### 2. What is a token?

The smallest individual unit in C programming is called a token.



### 3. What are Identifiers? ®

Identifiers are the names given to programming elements like variables, functions, arrays, structures etc.

Identifiers are collection of alphanumeric characters.

It can be written both in upper and lower case.

Identifiers are both user defined and system defined.

### 4. What are keywords? ®

Keywords are the reserved words in the language, whose meaning is already known to the compiler. Keywords are system defined identifiers.

Keywords cannot be used as a variable and are always written in lower case.

There are total 32 keywords in C programming language.

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

### 5. What is a variable? ®

A variable is any entity which can change.

Variable is a user defined element.

It is used to store the data. / Variable represents the memory location that can store a value.

Syntax for declaring a variable:

datatype variable name;

eg: int no;

## 6. What are constants? ®

A constant is an entity which remains same throughout the program.

eg: const float pi = 3.14;

Types:

- i. Primary Constant: Integer constant, Float constant, Character constant
- ii. Secondary Constant: Array, Pointer, Structure, Union, Enum.

**Rules for creating a Integer constant:**

- a) An integer constant should be at least one digit and can not have decimal point.
- b) It can be positive or negative if the sign is not given then the integer constant is considered positive.
- c) No commas or blank spaces are allowed within the integer constant.
- d) The range of integer constant is from -32768 to +32767
- e) Eg:- +426,9000,-945,89,-67

**Rules for creating Float constant:**

- a) A float constant should be at least one digit and must have decimal point.
- b) It can be positive or negative if the sign is not given then the float constant is considered positive.
- c) No commas or blank spaces are allowed within the float constant.
- d) The range of float constant is from 3.2E48 to +3.2E48
- e) Eg:- +426.23,9000.45,-945.89 float a = 425.45

**Rules for creating Character constant:**

- a) A single character enclosed in apostrophes is known as a character constant.
- b) A character constant can be a single alphabet, digit, or special symbol.
- c) A character constant is specified using "".
- d) e.g:- 'G','8','+', '@,'

## 7. What is a String?

A string can be defined as a collection of characters.

eg: "chirag", "errorbynigh" etc.

## 8. What are Operators?

The symbols which are used to indicate or create operations on operands are known as operators.

eg: 10+20=30 (10,20,30-operands, =,+ - operators)

Types:

- |                           |                                    |
|---------------------------|------------------------------------|
| i. Arithmetic operator    | v. Logical operator                |
| ii. Relational operator   | vi. Assignment operator            |
| iii. Conditional operator | vii. Special operator              |
| iv. Bitwise operator      | viii. Increment/Decrement operator |

## 9. Arithmetic Operators

These are operators used to do basic operations like addition, subtraction, multiplication, division etc.

Operators	Meaning	Example	Result
+	Addition	4+2	6
-	Subtraction	4-2	2
*	Multiplication	4*2	8
/	Division	4/2	2
%	Modulus operator to get remainder in integer division	5%2	1

## 10. Relational Operators®

Relational operators are also known as comparison operators.

These operators check the relation between two values or variables.

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
!=	Not equal to	5!=2	True

## 11. Logical Operators

The logical operators are used to evaluate the conditions or expressions.

Operator	Meaning	Example	Result
&&	Logical and	(5<2)&&(5>3)	False
	Logical or	(5<2)  ((5>3))	True
!	Logical not	!(5<2)	True

## 12. Assignment Operator

The assignment operator is used to assign some value to a variable.

eg: a=10;

## 13. Conditional Operator® ®

The conditional operator is also called as ternary operator. It is a combination of ? and : the syntax of conditional operator is:

exp1? exp2 : exp3

eg: max = a>b? a:b

## 14. Special Operator

These are the operators which are used to perform special operations.

eg: sizeof, pointer operators (\*,&)

## 15. Bitwise Operator

These are used to manipulate data at bit level.

0 and 1 are called bits, so to do multiple operations on these bits, we make use of bitwise operators.

eg: (& = Bitwise And), (| = Bitwise OR), (^ = Bitwise exclusive or), (>> = Bitwise Shift right), (<< = Bitwise Shift Left)

## **16. Increment/Decrement Operator®**

**Increment operator – it is used to increment the value of a variable by 1**

**types: pre-increment (`++i`) & post-increment (`i++`)**

**Decrement operator – it is used to decrement the value of the variable by 1**

**types: pre-decrement (`--i`) & post-decrement (`i--`)**

## **17. What are Data types?**

**A data type is a classification of data which tells the compiler or interpreter how the programmer intends to use the data.**

**Types:**

**i. Primary data type: `char`, `int`, `float`, `double`, `long`**

**ii. Secondary data type: `array`, `structure`, `enum`, `union`**

## **18. Pre-processor directive**

**pre-processor directive is `#define`**

**eg: `#define MAX 40`**

**`#define` is generally used to define constant values**

**It is generally written in uppercase (`MAX`)**

**It never terminates with a semicolon(`;`)**

## **19. Main function [`void main()`]**

**The main functions is used to start the actual C program**

**It contains two parts: Declarative statements & Executable statements**

**eg:**

**`void main()`**

**{**

```
    int a = 10;  
    clrscr();  
    printf("\nValue of a is %d",a);  
    getch();
```

**}**

## **20. What is the use of `printf()` statement? ®**

**The `printf()` statement is used to display the value of variable or data on the screen.**

**syntax: `printf("format specifier",variable name);`**

**eg: `printf("%d",a);`**

## **21. What is the use of `scanf()` statement? ®**

**The `scanf()` statement is used to accept the data from the user through keyboard.**

**syntax: `scanf("format specifier",&variable name);`**

**eg: `scanf("%d",&a);`**

## **22. Common format specifier's? ®**

**`%d` – integer**

**`%s` – string**

**`%u` – pointer**

**`%o` - octal**

**`%f` – float**

**`%c` – character**

**`%x` – hexadecimal**

**`%ld` – long int**

### **23. What is operator precedence?**

**If multiple operators are used in a single expression then operator precedence helps to solve the expression.**

**eg:  $7+3*2 = 13$**

### **24. What is operator associativity?**

**When multiple operators in an expression have the same priority, then the sequence of operations performed is decided by their associativity.**

**eg:  $7+3-2 = 8$**

### **25. Header files®**

**<stdio.h>:- It is a standard Input-Output header file.**

**This file is used to perform standard input-output operations like printf() and scanf() functions.**

**<conio.h>:- It is a Console Input Output header file. It is used to perform console (monitor) related operations like getch() function.**

**<math.h>:- This file is used to perform or supports all the mathematical related functions in C language.**

### **26. Input functions®**

**Input functions are used to accept the input by the user for a program.**

**Standard input functions:-**

**1. scanf() :- used for accepting a value or a data from keyboard**

**2. getchar() :- is used to get/read a one character from keyboard**

**3. getch() :- Accepts a character**

**4. gets() :- It accepts and stores the sequence of characters**

### **27. Output functions®**

**Output functions are used to print the values of variables on the monitor.**

**Standard Output functions:-**

**1. printf() :- It is a function used for displaying a value or a data on the screen**

**2. putchar() :- used to write a character on standard output/screen.**

**3. putch() :- it takes an ASCII int value as argument and then prints corresponding character**

**4. puts() :- takes an character array as argument and print the value stored in the character on the screen**

### **28. Data type conversion (Type casting) ®**

**The conversion of one data type to another is known as type casting.**

**Data type conversion can be done in two ways**

**i. Implicit data type conversion**

**int x = 123;**

**double y = x;**

**ii. Explicit data type conversion**

**double y = 123;**

**int x = (int) y;**

## CHAPTER:3 – CONTROL STRUCTURE

### **1. What are decision making statements?**

The statements which help check the condition and execute statements according to the condition criteria are called decision making statements or conditional statements.

### **2. List types of decision making statements?**

- i. If statement
- ii. If-else statement
- iii. If-elseif ladder / ladder if else
- iv. nested if-else
- v. switch case

### **3. List types of repetition?**

- i. for loop
- ii. while loop
- iii. do-while loop

### **4. State the use of if-statement?**

The if-statement is used to decide whether a statement or a block of statements will be executed or not.

If the condition is true then the statement will be executed otherwise not.

**syntax:**

```
if(condition)
{
    statement;
}
```

### **5. State the use of if-else statement?**

The if-else statement is used to perform operations based on specific conditions, if the condition is true then the statements in the if-block will be executed otherwise the statements in the else block will be executed.

**syntax:**

```
if(condition)
{
    statement;
}
else
{
    statement;
}
```

**6. State the use of ladder if-else statement? ®**

The ladder if-else statement is used to check a set of conditions in a sequence, the condition in ladder if else is only checked if the previous condition is false.

**syntax:**

```
if(condition)
{
    statement;
}
elseif(condition)
{
    statement;
}
.
.
.
else
{
    statement;
}
```

**7. State the use of nested if statement? ®**

The nested if statement is an if statement inside an if statement, it is used to check the condition inside a condition. The condition 2 will only be checked if the condition 1 is true.

**syntax:**

```
if(condition)
{
    if(condition)
    {
        statement;
    }
}
```

**8. State the use of switch statement?**

The switch statement is an alternative to the ladder if else statement, it is used to execute multiple statements based on different possible values of a variable known as switch variable.

**syntax:**

```
switch(expression)
{
    case value 1:
        statement;
        break;
```

```

case value 2:
statement;
break;
.
.
.
default:
statement;
break;
}

```

#### **9. Use of break statement? ®**

**It is a control statement which is used to terminate the loop or used to stop loop iterations.**

#### **10. Use of continue statement? ®**

**It is a control statement which is used to transfer the control to the beginning of the loop. It is generally used with if statement.**

**Eg:**

```

main()
{
    int i, j ;
    for ( i = 1 ; i <= 2 ; i++ ) //inner
    {
        for ( j = 1 ; j <= 2 ; j++ )
        {
            if ( i == j )
                continue ;
            printf ( "\n%d %d\n", i, j ) ;
        }
    }
}

```

#### **11. Use of goto statement?**

**It is a control statement which is used to transfer the control of the program from one statement to another statement.**

**The statement where controlled is transferred is labeled by the label i.e name is given to that statement.**

**Eg:**

```

main( )
{
    for ( int i = 1 ; i <= 100 ; i++ ) {
}

```

```

if ( i == 5 )
    goto out ;
else
    printf ( "\n %d ", i );
}
out :
printf ( "Out of the loop at last!" );
}

```

**12. State the use of for loop?**

**For loop is used to iterate the statements of the program several times.**

**Syntax:**

```

for(initialization; condition; increment/decrement)
{
    statements;
}

```

**13. State the use of while loop? ®**

**While loop is an entry controlled loop, it allows to repeatedly run the same block of code until the condition is met**

**Syntax:**

```

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

```

**14. State the use of do-while loop? ®**

**Do-while is an exit-controlled loop, it suggests to do the action or allows executing the statement in the loop at least once.**

**Syntax:**

```

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

```

**15. Difference between while and do-while loop**

While loop	Do-While loop
i. It is an entry-controlled loop	1. It is an exit-controlled loop
ii. It checks the condition at the beginning of the loop	2. It checks the condition at the end of the loop
iii. more accurate than do-while	3. not as accurate as while loop
iv. executes condition only if condition is true	4. executes the body statements at least once even if condition is false.

## CHAPTER:4 – ARRAY & STRUCTURE

### 1. Define Array and state its declaration? ® ®

**Array is a collection of similar data type.**

**Array can be created from any of C data types: int, float, char etc.**

**Declaration:**

**datatype arrayname[size];**

**eg: int no[5];**

### 2. List types of Array? ®

**i. Single dimensional array**

**ii. Two dimensional array**

**iii. Multi-dimensional array**

### 3. State syntax for declaring One dimensional array with example

**Syntax: datatype arrayname[size]={list of values};**

**Example:- int no[5]={ 10,20,30,40,50};**

### 4. Define Two-dimensional array and state its syntax and initialization ® ®

**Two-dimensional array is used to store tabular data i.e when the data is in the form of rows and columns.**

**Syntax:- datatype arrayname[size][size];**

**Example:- int arr[2][3]; //declaration**

**int arr[2][3]={ {1,2,3}, {4,5,6} }; //initialization**

### 5. Define String and state its declaration

**String can be defined as a collection of characters.**

**eg: "errorbynigh"**

**Syntax : datatype arrayname[size];**

**eg : char name[20];**

**char name[20] = "chirag";**

### 6. Define table of string

**Table of string is a collection of strings.**

**In one table we can accept multiple strings.**

**Syntax: datatype arrayname[size][size] //first [size] = length of string**

**Eg: char name[5][10] //10 strings of length not more than 5**

### 7. Explain string functions® ® ®

**i. strlen()®:- It is string length.**

**It calculates the length of the string.**

**Syntax:-**

**int a = 0;**

**a=strlen("welcome"); OR**

**char name[10] = "prathmesh";**

**int len=0;**

**len=strlen(name);**

ii. **strcpy()®:- String copy**

**This function is used to copy one string to another string.**

**It takes 2 arguments.**

**Syntax:- strcpy(str1,str2);**

**Here str2 will get copy on str1**

iii. **strcmp():- It is string compare function.**

**String compare function compares two strings with each other.**

**If two strings are equal then function returns 0 else the difference between two strings.**

iv. **strcat()®:- String concatenation.**

**String concatenation function is used to join the two strings.**

**It takes 3 arguments as follows:**

**Syntax: strcat(str1,str2,n);**

**This will add contents of n characters of string2 to string1.**

**8. State the advantages of Array**

- i. **An array can store number of values under single name.**
- ii. **In memory the array elements are stored contiguously.**
- iii. **It is easy to manipulate the values stored in an array**
- iv. **Array is a flexible data structure. Means we can perform adding or removing of element at any position.**

**9. State disadvantages of Array**

- i. **Array stores only a group of similar data type and it cannot hold set of different data types.**
- ii. **An array is static.**
- iii. **Allocating more memory than needed leads to wastage of memory.**

**10. What is Multi-dimensional array**

**An array which can store more than two subscripts is known as multi-dimensional array.**

**Syntax : datatype arrayname[size][size][size];**

**eg: int no[2][2][2];**

**11. Define Structure and state its declaration**

**Structure can be defined as a collection of dissimilar data types.**

**All the elements in the structure are known as structure members.**

**Syntax:**

**struct structure name**

**{**

**datatype var1;**

**datatype var2;**

**.....**

**};**

**12. State the two main parts of a structure program**

- i. Declaration of structure
- ii. Accessing structure members

**13. How to access structure members?**

To access structure members we need to declare a variable of structure type, and need to use the dot operator.

Syntax:

```
struct student
```

```
{
```

```
    int rollno;
```

```
    char name[20];
```

```
}
```

OR

```
struct student s;
```

```
//accessing structure members
```

Syntax:- structvariable.membername

i.e. s.name, s.rollno, s.percentage

**14. Define Array of structure**

Array is collection of same data type.

Structure is collection of dissimilar data type.

Array of structure is required whenever we need to store more records.

Syntax:

```
struct structurename
```

```
{
```

```
    datatype variablename1;
```

```
    datatype variablename2;
```

```
-----
```

```
}variablename[size];
```

Example:

```
struct student
```

```
{
```

```
    int rollno;
```

```
    char name[10];
```

```
    float per;
```

```
}s[3];
```

**15. Difference between Array and Structure**

Array	Structure
i. Array is a collection of similar data type	i. Structure is a collection of dissimilar data type
ii. No need of any keyword except data type.	ii. "struct" keyword is used for structure declaration.

## **16. Define nested structure**

**Nested structure can be defined as a structure within a structure.**

**Syntax:**

**struct date**

**{**

**char day[10];**

**int month;**

**int year;**

**}**

**struct account**

**{**

**char name[10];**

**int accno;**

**float balance;**

**}**

**struct date d = { "Monday",1,2018};**

**struct account a = {"abhi",1055,15000.00};**

## **17. Define typedef data type?**

**Typedef is a keyword which is used to give a new symbolic name for the existing name in C program**

## **18. Define Enum?**

**Enum is user defines**

**It is used when we already knew finite list of values that a data type can take.**

**eg: enum months{jan, feb, march, apr, may};**

## CHAPTER: 5 – FUNCTIONS

### 1. What is function?

The function is sub program or set of instructions written to do a particular task.

### 2. What is the need of function?

- i. It avoids rewriting of the same code again and again.
- ii. Using functions make it easier to write the program

### 3. State the types of functions?

- i. In build /Readymade functions (printf(), scanf())
- ii. User defined functions

### 4. State the advantages of functions?

- 1) Due to function ,program approach becomes top to down modular. It helps solving complex logical problems in the program.
- 2) The length of the program gets reduce because functions are called whenever they are required only.
- 3) While debugging errors, it becomes easy to find out any type of errors.
- 4) The function written once can be used many times.
- 5) Functional approach saves time and memory space.

### 5. State the elements of user-defined functions?

- 1) Function declaration
- 2) Function definition
- 3) Function call

### 6. What is function declaration?

- Function declaration is similar to the variable declaration.
- Function can be declared above main( ) or after main( ).
- If function is declared above main ( ) function then it is called global function declaration .
- If function is declared after main ( ) function then it is called local function declaration .

### 7. State the syntax for function declaration?

**datatype functionname (datatypes of arguments);**  
eg : void add(void);

### 8. What is function definition?

The actual statements which are to be executed are included in the function definition.

**Syntax: datatype functionname(formal arguments);**  
eg: void add() {  
}

**9. What is function call?**

**Function call is calling a function in the program by writing the function name and by passing the arguments if necessary.**

**Syntax: functionname(list of actual arguments);**

**eg: add(5,12);**

**10. What are the two types of Function call?**

- i. **Call by value**
- ii. **Call by reference**

**11. What is Call by value?**

**When we call the function by passing normal values or variables then it is called call by value.**

**eg: add(a,b); or add(5,12);**

**12. What is Call by reference?**

**When we call the function by passing the reference of variables or values then it is called call by reference.**

**eg: add(&a,&b);**

**13. What are formal parameters?**

**When we receive these values at the time of function definition then these values or variables are called as formal parameters.**

**14. What are actual parameters?**

**When we pass the values to the function at the time of function call then these values or variables are called as actual parameters.**

**15. State the function categories®**

1. **Function with no argument and no return values**
2. **Function with argument and no return values.**
3. **Function with no argument but return a value**
4. **Function with argument and return a value**

**16. Function with no argument & no return values**

**The function which does not return any value and to which we do not pass any arguments is called as function with no argument and no return values.**

**Syntax:- void functionname(void); Eg: void add(void);**

**17. Function with argument & no return value**

**The function which does not return any value and to which we pass any arguments is called as function with argument and no return values.**

**Syntax: void functionname(argument list); void add(int);**

**18. Function with no argument but return a value**

**The function which does not accept arguments but returns the value is called as Function with no argument but return a value.**

**Syntax:- returntype functionname(void); int add(void);**

**Eg: int add(void);**

## **19. Function with argument and return a value**

**Functions may have one or more than one arguments with it which can be of different data types.**

**Also this function returns a value of required data type.**

**Syntax: datatype functionname(datatype variable name); eg: int add(int);**

## **20. What is recursion?**

**When function calls itself in its own body then it is called as Recursive Function.**

**eg:**

```
void main()
{
    printf("\nHello");
    main();
    getch();
}
```

## **21. Difference between getch() and getche()**

<b>getch()</b>	<b>getche()</b>
i. getch() accepts the entered character but does not echo it on the screen	i. getche() accepts the entered character and also echos it on the screen
ii. int main() {     char ch;     clrscr();     printf("\n Enter character:");     ch=getch();     printf("\n entered character is %c ",ch);     return 0; }	ii. int main() {     char ch;     clrscr();     printf("\n Enter character:");     ch=getche();     printf("\n entered character is %c ",ch);     return 0 ; }
iii. o/p: Enter character: Entered character is C	iii. o/p: Enter character: C Entered character is C

## CHAPTER:6 - POINTERS

### 1. What is a pointer? ®

A pointer is a variable who stores the memory address of another variable.

### 2. Pointer declaration®

Syntax: datatype \* pointername;

eg: int \*ptr;

### 3. How to access pointer?

Syntax: pointer name = &variable name

eg: ptr = &no;

### 4. What is the format specifier used for printing pointer?

The format specifier is %u.

### 5. Explain pointer arithmetic? ®

Pointer arithmetic is related with the arithmetic operations with the pointers.

Basic operations + , - , \*, / can be done using pointer notation.

As \* pointer indicates the value whose address is stored in pointer Eg:

result = \*p1 + \*p2;

sum sum + \*p1;

x= 5 + \*p1 / \*p2;

Y= \*p1 - \*p2;

### 6. What is the use of (\*)

The \* operator is a unary operator and it is used to access the value of variable located at specified address given in operand.

Example: ptr=&no;

\*ptr:- it points to the value stored at location specified in ptr.

### 7. What is the use of &

The & is a unary operator it means it has only one operand.

It is used to store the memory address of operand.

Eg:

int no;

printf("\n enter the number:");

scanf("%d",&no);

### 8. Explain the array with a pointer

Array is a collection of the same data type. A pointer can be assigned to the array's zeroth element and the rest of the elements can be easily accessed by the pointer.

Example:

int no= [5,7,18,19,21];

int \*ptr;

ptr = &no[0]; //assigning base address to pointer

**9. Explain array of pointers**

**Array of pointers is formed by declaring its capacity with it.**

**We can declare array of pointers as follows:**

**syntax: datatype \*arrayname[size];**

**Example: char \*str[5];**

**10. Explain structure with pointer**

**Pointer can be declared of structure type. We can declare pointer to whole structure. The '>' operator is used to point to the members of the structure.**

**Syntax:**

```
struct structurename  
{  
    variables;  
};struct *ptr;
```

# **PROGRAMMING**

1. Q. Write a program to add two numbers® (and multiply)

```
//program to add two numbers
#include<stdio.h>
#include<conio.h>
void main()
{
    int a, b, add;
    clrscr();
    printf("\nEnter two numbers:");
    scanf("%d%d",&a,&b);
    add = a+b;
    printf("\nAddition of two numbers is %d",add);
    getch();
}
```

2. Q. Write a program to display the area and perimeter of rectangle

```
//program to display the area and perimeter of rectangle
#include<stdio.h>
#include<conio.h>
void main()
{
    int l,b,a,p;
    clrscr();
    printf("\nEnter the values of length and breadth:");
    scanf("%d%d",&l,&b);
    a = l*b;
    p = 2*l + 2*b;
    printf("\nArea of rectangle is %d",a);
    printf("\nPerimeter of rectangle is %d",p);
    getch();
}
```

3. Q. Write a program to calculate the area and circumference of circle

```
//program to calculate the area and circumference of circle
#include<stdio.h>
#include<conio.h>
const float pi = 3.14; //global variable declaration
void main()
{
    float r, a, cir;
    printf("\nEnter the value of radius:");


```

```

scanf("%f",&r);
area = pi*r*r;
cir = 2*pi*r;
printf("\nArea of circle is %f",a);
printf("\nCircumference of Circle is %f",cir);
getch();
}

}

```

4. Q. Write a program to swap the values of two numbers

```

//program to swap the values of two numbers
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,temp;
    clrscr();
    printf("\nEnter the values of a & b:");
    scanf("%d%d",&a,&b);
    printf("Values of a & b before swapping are %d and %d",a,b);
    temp = a;
    a = b;
    b = temp;
    printf("Values of a & b after swapping are %d and %d",a,b);
    getch();
}

```

5. Q. Write a program to find greater of two numbers

```

//program to find greater of two numbers
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b;
    clrscr();
    printf("\nEnter the values of a & b:");
    scanf("%d%d",&a,&b);
    if(a>b)
        printf("%d is greater",a);
    else
        printf("%d is greater",b);
    getch();      }

```

6. Q. Write a program to accept a number and find whether it is even or odd  
//program to find whether the entered number is even or odd

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int no;
    clrscr();
    printf("\nEnter a number");
    scanf("%d",&no);
    if(no%2==0)
        printf("%d is even",no);
    else
        printf("%d is odd",no);
    getch();
}
```

7. Q. Write a program to convert temperature from fahrenheit to celsius

//program to convert temperature from fahrenheit to celsius

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float f,c;
    clrscr();
    printf("\nEnter temperature in fahrenheit:");
    scanf("%f",&f);
    c = (f-32)*1.8;
    printf("\nTemperature in celsius is %f",c);
    getch();
}
```

8. Q. Write a program to wish hello to user

//program to wish hello to user

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char name[10];
    clrscr();
    printf("\nEnter name:");


```

```
    scanf("%s",name);
    printf("\nHello %s",name);
    getch();
}
```

9. Q. Write a program to convert temperature from celsius to kelvin

```
//program to convert temperature from celsius to kelvin
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float k,c;
    clrscr();
    printf("\nEnter temperature in celsius:");
    scanf("%f",&c);
    k = c+273;
    printf("\nTemperature in kelvin is %f",k);
    getch();
}
```

10. Q. Write a program to convert kilometer into miles and meters

```
//program to convert kilometer into miles and meters
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float km,m,mi;
    clrscr();
    printf("\nEnter distance in kilometer:");
    scanf("%d",&km);
    m = km*1000;
    mi = km/1.609;
    printf("\nDistance in meters is %f",m);
    printf("\nDistance in miles is %f",mi);
    getch();
}
```

11. Q. Write a program to reverse a given number

```
//program to reverse a given number
```

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

```

void main()
{
    int no, rev, rem=0;
    clrscr();
    printf("\nEnter a number:");
    scanf("%d",&no);
    while(no!=0)
    {
        rem=no%10;
        rev = rev*10 + rem;
        no=no/10;
    }
    printf("\nReversed number is %d",rev);
    getch();
}

```

12. Q. WAP to check entered number is positive or negative

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no;
    clrscr();
    printf("\nEnter a number");
    scanf("%d",&no);
    if(no>0)
        printf("%d is positive",no);
    else
        printf("%d is negative",no);
    getch();
}

```

13. Q. Write a program to check whether the entered year is leap or not®

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int year;
    clrscr();
    printf("\nEnter a year");
    scanf("%d",&year);
    if(year%4==0)
        printf("%d is a leap year",year);
}

```

```

        else
            printf("%d is not a leap year",year);
        getch();
    }
}

```

14. Q. Write a program to find the greatest of 3 entered numbers

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c;
    clrscr();
    printf("Enter three numbers:");
    scanf("%d%d%d",&a,&b,&c);

    if(a>b && a>c)
    {
        printf("\n%d is greatest of 3 numbers entered", a);
    }
    else if(b>a && b>c)
    {
        printf("\n %d is the greatest of 3 numbers entered", b);
    }
    else if(c>a && c>b)
    {
        printf("\n %d is the greatest of three numbers entered", c);
    }
    else
    {
        printf("\n All numbers are equal");
    }
    getch();
}

```

15. Q. Write a program to find the smallest of 3 entered numbers

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c;
    clrscr();
    printf("Enter three numbers:");

```

```

scanf("%d%d%d",&a,&b,&c);

if(a<b && a<c)
{
    printf("\n%d is smallest of 3 numbers entered", a);
}
else if(b<a && b<c)
{
    printf("\n %d is the smallest of 3 numbers entered", b);
}
else if(c<a && c<b)
{
    printf("\n %d is the smallest of three numbers entered", c);
}
else
    printf("\n All numbers are equal");
getch();
}

```

16. Q. Write a menu driven program

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no,choice;
    clrscr();
    printf("\n 1.Square");
    printf("\n 2.Cube");
    printf("\n 3.Octal");
    printf("\n 4.Hexadecimal");
    printf("\n enter the choice:");
    scanf("%d",&choice);
    printf("\n enter the number:");
    scanf("%d",&no);
    if(choice==1)
    {
        printf("\n the square of no is %d ",no*no);
    }
    else if(choice==2)
    {
        printf("\n the cube of no is %d ",no*no*no);
    }
}

```

```

    }
    else if(choice==3)
    {
        printf("\n the octal no is %o ",no);
    }
    else if(choice==4)
    {
        printf("\n the hexadecimal no is %x ",no);
    }
    else
    {
        printf("\n the choice is invalid");
    }
    getch();
}

```

17. Q. Write a program to check whether the entered number is prime or not

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no;
    clrscr();
    printf("\nEnter the number:");
    scanf("%d",&no);
    if(no==1 || no==2 || no==3 || no==5 || no==7)
    {
        printf("%d is a prime number",no);
    }
    else if(no%2>0 && no%3>0 && no%5>0 && no%7>0)
    {
        printf("%d is a prime number",no);
    }
    else
    {
        printf("%d is not a prime number",no);
    }
    getch();
}

```

18. Q .Write a program to print result of the student according to the marks entered

```

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

```

```

void main()
{
    int mark;
    clrscr();
    printf("\n enter the marks :");
    scanf("%d",&mark);
    if(mark > 0 && mark < 50)
    {
        printf("\n the student is fail");
    }
    else if(mark >= 50 && mark <= 54)
    {
        printf("\n the student is passed in pass class");
    }

    else if(mark >= 55 && mark <= 59)
    {
        printf("\n the student is passed in second class");
    }
    else if(mark >= 60 && mark <= 69)
    {
        printf("\n the student is passed in first class");
    }
    else if(mark >= 70 && mark <= 79)
    {
        printf("\n the student is passed in Distinction");
    }
    else if(mark >= 80 && mark <= 100)
    {
        printf("\n the student is passed in Merit");
    }
    else
    {
        printf("\n entered marks are invalid");
    }
    getch();
}

```

19. Q. Write a program to find greatest of 3 numbers using nested if

```

#include<stdio.h>
#include<conio.h>
void main()

```

```

{
    int a,b,c;
    clrscr();
    printf("\nEnter 3 numbers:");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)
    {
        if(a>c)
        {
            printf("%d is greatest",a);
        }
        else
        {
            printf("%d is greatest",c);
        }
    }
    else
    {
        if(b>c)
        {
            printf("%d is greatest",b);
        }
        else if(a==b && b==c)
        {
            printf("\nAll numbers are equal");
        }
        else
        {
            printf("%d is greatest",c);
        }
    }
    getch();
}

```

20. Q. Write a program to check whether the entered character is a vowel or not

```

#include<stdio.h>
#include<conio.h>
void main()
{
    char c;
    clrscr();
    printf("\nEnter a character:");

```

```

scanf("%c",&c);
if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u')
{
    printf("%c is a vowel",c);
}
else
{
    printf("%c is a consonant",c);
}
getch();
}

```

21. Q. Write a program to print factorial of a number

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n,f=1;
    clrscr();
    printf("\nEnter a number");
    scanf("%d",&n);
    while(n>=1)
    {
        f=f*n;
        n--;
    }
    printf("\nThe factorial is %d",f);
    getch();
}

```

22. Q. Write a program to accept position from the user and print Fibonacci series. ®

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int fno=0,sno=1,tno, pos, i=0;
    clrscr();
    printf("\nEnter the position:");
    scanf("%d",&pos);
    printf("%d\t%d",fno,sno);
    while(pos>=i)
    {
        tno=fno+sno;

```

```

        printf("\t%d",tno);
        fno=sno;
        sno=tno;
        i++;
    }
    getch();
}

```

23. Q. Write a program to accept a name and print it 5 times

```

#include<stdio.h>
#include<conio.h>
void main()
{
    char name[20];
    int i;
    clrscr();
    printf("\nEnter name:");
    scanf("%s",name);
    for(i=0;i<5;i++)
    {
        printf("\nHello %s",name);
    }
    getch();
}

```

24. Q. Write a program to accept 5 numbers, if their sum is 100 then terminate the loop.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,sum=0;
    clrscr();
    for(i=0;i<5;i++)
    {
        enter:
        printf("\nEnter number:");
        scanf("%d",&n);
        sum=sum+n;

        if(sum>=100)

```

```

    {
        printf("\nThe sum is %d",sum);
        break;
    }
    else
    {
        goto enter;
    }
}
getch();
}

```

25. Q. Write a program to find the sum of digits of number entered

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no, sum=0, rem=0;
    clrscr();
    printf("\nEnter a number:");
    scanf("%d",&no);
    while(no!=0)
    {
        rem=no%10;
        sum=sum+rem;
        no=no/10;
    }
    printf("\nSum of digits is %d",sum);
    getch();
}

```

26. Q. Write a program to print table of a number entered by user

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i=1;
    clrscr();
    printf("\nEnter a number:");
    scanf("%d",&n);
    do
    {
        printf("\n%d x\t%d = %d",n,i,n*i);
    }
}

```

```

    i++;
}while(i<=10);
getch();
}

```

27. Q. Write a program to display series of 11-30 in reverse order

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int sum=0,i;
    clrscr();
    for(i=30; i>=11; i--)
    {
        printf("\n%d",i);
        sum=sum+i;
    }
    printf("\nSum is %d", sum);
    getch();
}

```

28. Q. Write a menu driven program to show arithmetic operators

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no1,no2,choice;
    clrscr();
    printf("\n 1.Addition:");
    printf("\n 2.Subtraction:");
    printf("\n 3.Multiplication:");
    printf("\n 4.Division:");
    printf("\n Enter the choice:");
    scanf("%d",&choice);
    printf("\n enter the 2 numbers:");
    scanf("%d %d",&no1,&no2);
    switch(choice)
    {
        case 1:
            printf("\n the addition is %d",no1+no2);
            break;
        case 2:
            printf("\n the subtraction is %d",no1-no2);
    }
}

```

```

        break;
    case 3:
        printf("\n the multiplication is %d",no1*no2);
        break;
    case 4:
        printf("\n the division is %d",no1/no2);
        break;
    default:
        printf("\n invalid choice:");
        break;
    }
    getch();
}

```

29. Q. Write a program to display tables from 2-12

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n=1,i;
    clrscr()
    do
    {
        i=1;
        n++;
        do
        {
            printf("\n%d x\t%d = %d",n,i,n*i);
            i++;
        }while(i<=10);
        printf("\n");
    }while(n<=12);
    getch();
}

```

30. Q. Write a program to display Armstrong number

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int no, original, rem, res=0;
    clrscr();
    printf("\nEnter a three digit number:");

```

```

scanf("\n%d",&no);
original = no;
while(original!=0)
{
    rem = original%10;
    res = res+rem*rem*rem;
    original = original/10;
}
if(res==no)
{
    printf("\n%d is an armstrong number",no);
}
else
{
    printf("\n%d is not an armstrong number",no);
}
getch();
}

```

31. Q. Write a program to display factors of a number

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i=1;
    clrscr();
    printf("\nEnter a number:");
    scanf("\n%d",&n);
    printf("\nFactors of the number entered are:");
    for(i=1; i<=n; i++)
    {
        if(n%i==0)
        {
            printf("\n%d",i);
        }
    }
    getch();
}

```

32. Q. Write a program to produce pattern

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

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

```
void main()
```

```
{
```

```
    int i, j, pos, cnt=1;
```

```
    clrscr();
```

```
    printf("\nEnter position");
```

```
    scanf("%d",&pos);
```

```
    for(i=1; i<=pos; i++)
```

```
{
```

```
    for(j=1; j<=i; j++)
```

```
{
```

```
        printf("\t%d",cnt++);
```

```
}
```

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

```
}
```

```
    getch();
```

```
}
```

```
//program to count number of digits of the number entered by user
```

```
void main()
```

```
{
```

```
    int n,c=0;
```

```
    printf("\nEnter a number");
```

```
    scanf("\n%d",&n);
```

```
    while(n!=0)
```

```
{
```

```
    n=n/10
```

```
    c++;
```

```
}
```

```
    printf("\nNo of digits is %d",c);
```

```
}
```

```
//program to check greater of 3 no's using conditional operator
```

```
grt = a>b? (a>c? a:c) : (b>c? b:c);
```

1

2 3

4 5 6

7 8 9 10

33. Q. Write a program to initialize and print the array elements

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int number[5]={10,15,78,90,100};
    int i=0;
    clrscr();
    for (i=0; i<5;i++)
    {
        printf("\t %d ",number[i]);
    }
    getch();
}
```

34. Q. Write a program to accept and display 5 elements using integer array

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int no[5], i=0;
    clrscr();
    for (i=0;i<=4;i++)
    {
        printf("\n Enter the number:");
        scanf("%d",&no[i]);
    }
    for (i=0;i<=4;i++)
    {
        printf("\n %d is stored at %d location of array",no[i], i);
    }
    getch();
}
```

35. Q. Write a program to accept 5 numbers in an integer array and find it's sum and average.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int no[5], i=0, sum=0, avg=0;
    clrscr();
    for (i=0;i<=4;i++)
```

```

    {
        printf("\n Enter the number:");
        scanf("%d",&no[i]);
        sum = sum + no[i];
    }
    avg = sum/5;
    printf("\n Sum of array numbers=%d",sum);
    printf("\n Average of array numbers= %f",avg);
    getch();
}

```

36. Q. Write a program to accept 5 numbers in an integer array and find the smallest number

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no[5], i=0,small;
    clrscr();
    for (i=0;i<=4;i++)
    {
        printf("\n Enter the number:");
        scanf("%d",&no[i]);
    }
    small = no[0];
    for (i=0;i<=4;i++)
    {
        if (no[i] <small)
        {
            small= no[i];
        }
    }
    printf("\n %d is the smallest number", small);
    getch();
}

```

37. Q. Write a program to accept 5 numbers in an integer array and find the greatest number

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no[5], i=0,great;

```

```

clrscr();
for (i=0;i<=4;i++)
{
    printf("\n Enter the number:");
    scanf("%d",&no[i]);
}
great = no[0];
for (i=0;i<=4;i++)
{
    if (no[i] >great)
    {
        great= no[i];
    }
}
printf("\n %d is the greatest number", great);
getch();      }

```

38. Q. Write a program to accept numbers in two arrays and add them in third array

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a[3], b[3], c[3], i=0;
    clrscr();
    printf("\n Enter elements for first array ");
    for (i=0;i<=2;i++)
    {
        printf("\n Enter the number:");
        scanf("%d",&a[i]);
    }
    printf("\n Enter elements for second array ");
    for (i=0;i<=2;i++) //used to accept the numbers for second array
    {
        printf("\n Enter the number:");
        scanf("%d",&b[i]);
    }
    for (i=0;i<=2;i++) // used to add elements of array
    {
        c[i]= a[i] + b[i] ;
        printf("\n c[%d]=%d", i ,c[i] );
    }
    getch();      }

```

39. Q. Write a program to accept 10 elements in an array and find the number of positive, negative and zeros

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int no[10], i=0,cnt1=0, cnt2=0,cnt3=0;
    clrscr();
    for (i=0;i<=9;i++) // used to accept the numbers for first array
    {
        printf("\n Enter the number:");
        scanf("%d",&no[i]);
        if (no[i] > 0)
        {
            cnt1++;
        }
        if (no[i] < 0)
        {
            cnt2++;
        }
        if (no[i] == 0)
        {
            cnt3++;
        }
    }
    printf("\n Number of positive numbers are: %d",cnt1);
    printf("\n Number of negative numbers are: %d",cnt2);
    printf("\n Number of zeros are: %d",cnt3);
    getch();
}
```

40. Q. Write a program to accept 5 elements and copy them into another array

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[5],b[5],i=0;
    clrscr();
    printf("\nEnter the elements of first array:");
    for(i=0;i<5;i++)
    {
        printf("\nEnter number:");
    }
```

```

        scanf("%d",&a[i]);
    }
    printf("Elements of second array are:");
    for(i=0; i<5;i++)
    {
        b[i]=a[i];
        printf("\nb[%d]=%d",i,b[i]);
    }
    getch();
}

```

41. Q. Write a program to accept 5 characters and convert them from uppercase to lowercase

```

#include<stdio.h>
#include<conio.h>
void main()
{
    char alphabet[5];
    int i=0;
    clrscr();
    for(i=0; i <=4; i++)
    {
        printf("\n Enter the alphabet:");
        scanf("\n%c",&alphabet[i]); OR alphabet[i]=getche();
        alphabet[i]= alphabet[i] - 32;
    }
    for(i=0; i <=4; i++)
    {
        printf("\n The converted character is %c",alphabet[i]);
    }
    getch();
}

```

42. Q. Write a program to sort the elements of array in ascending order®

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int arr[10], i=0, j=0, temp=0;
    clrscr();
    printf("\nEnter the array elements:");
    for(i=0; i<10; i++)
    {

```

```

printf("\nEnter number:");
scanf("%d",&arr[i]);
}
temp=arr[0];
for(j=0; j<10;j++)
{
    for(i=0; i<10; i++)
    {
        if(arr[i+1]<arr[i])
        {
            temp=arr[i];
            arr[i]=arr[i+1];
            arr[i+1]=temp;
        }
    }
}
printf("\nThe sorted array is:");
for(i=0; i<10; i++)
{
    printf("%d\t",arr[i]);
}
getch();
}

```

43. Q. Write a program to sort the elements of array in descending order

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int arr[10], i=0, j=0, temp=0;
    clrscr();
    printf("\nEnter the array elements:");
    for(i=0; i<10; i++)
    {
        printf("\nEnter number:");
        scanf("%d",&arr[i]);
    }
    temp=arr[0];
    for(j=0; j<10;j++)
    {
        for(i=0; i<10; i++)
        {

```

```

        if(arr[i+1]>arr[i])
        {
            temp=arr[i];
            arr[i]=arr[i+1];
            arr[i+1]=temp;
        }
    }
}

printf("\nThe sorted array is:");
for(i=0; i<10; i++)
{
    printf("%d\t",arr[i]);
}
getch();
}

```

44. Q. Write a program to initialize 2 dimensional array and display its contents

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no[2][2]={ {1,2}, {3,4}  };
    int i=0 , j = 0;
    clrscr();
    printf("\n Array contents are:\n");
    for( i = 0 ; i<=1; i++)
    {
        for(j=0 ; j<=1; j++)
        {
            printf("no[%d][%d]=%d\t", i , j , no[i][j] );
        }
        printf("\n");
    }
    getch();
}

```

45. Q. Write a program to accept the elements for 3 by 3 matrix and display the contents

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no[3][3];

```

```

int i=0, j=0;
clrscr();
for(i=0;i<=2;i++)
    //accept the array elements
{
    printf("\n ");
    for(j=0;j<=2; j++)
    {
        printf("Enter no[%d][%d]:\t", i , j );
        scanf("%d",&no[i][j]);
    }
}
for( i=0;i<=2;i++) //display the array element
{
    printf("\n ");
    for(j=0; j<=2; j++)
    {
        printf(" no[%d][%d]:%d \t", i , j , no[i][j]);
    }
}
getch();
}

```

46. Q. Write a program to add two dimensional 3x3 matrix® ®

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a[3][3] , b[3][3] , c[3][3];
    int i=0, j=0;
    clrscr();
    printf("\n Enter elements for first array:");
    for( i = 0 ; i <= 2 ; i++)
    {
        printf("\n ");
        for( j = 0 ; j <= 2 ; j++)
        {
            printf("Enter a[%d][%d] :\t ", i , j );
            scanf("%d",&a[i][j] );
        }
    }
    printf("\n Enter elements for second array:");
    for( i = 0 ; i <= 2 ; i++)

```

```

{
    printf("\n ");
    for(j = 0 ; j <= 2 ;j++)
    {
        printf("Enter b[%d][%d] :\t ", i,j );
        scanf("%d",&a[i][j] );
    }
}
for(i = 0 ; i <= 2; i++)
{
    for (j = 0 ; j<= 2; j++)
    {
        c[i][j] = a[i][j] + b[i][j];
    }
}
printf("\n Addition of two matrices\n");
for( i = 0 ; i <= 2; i++)
{
    for (j = 0 ; j<= 2; j++)
    {
        printf(c[%d][%d] = %d \t", i ,j, c[%d][%d]);
    }
    printf("\n");
}
getch();
}

```

47. Q. Write a program to multiply two dimensional 3x3 matrix

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a[3][3] , b[3][3] , c[3][3];
    int i=0, j=0;
    clrscr();
    printf("\n Enter elements for first array:");
    for( i = 0 ; i <= 2 ; i++)
    {
        printf("\n ");
        for(j = 0 ; j <= 2 ;j++)
        {
            printf("Enter a[%d][%d] :\t ", i ,j );

```

```

        scanf("%d",&a[i][j] );
    }
}

printf("\n Enter elements for second array:");
for( i = 0 ; i <= 2 ; i++)
{
    printf("\n ");
    for(j = 0 ; j <= 2 ; j++)
    {
        printf("Enter b[%d][%d] :\t ", i ,j );
        scanf("%d",&a[i][j] );
    }
}
for( i = 0 ; i <= 2; i++)
{
    for ( j = 0 ; j<= 2; j++)
    {
        c[i][j] = a[i][j] x b[i][j];
    }
}
printf("\n Multiplication of two matrices\n");
for( i = 0 ; i <= 2; i++)
{
    for ( j = 0 ; j<= 2; j++)
    {
        printf(c[%d][%d] = %d \t", i ,j, c[%d][%d]);
    }
    printf("\n");
}
getch();
}

```

48. Q. Write a program to display string

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char str[20] = "Hello Students";
    int i=0;
    printf("\n Entered string is %s",str);
    printf("\n Values stored at each location are:\n");
    for(i=0 ; i<=19; i++)
    {
        printf("str[%d]=%c \n",i,str[i]);
    }
    getch();
}
```

49. Q. Write a program to calculate the length of string

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char str[10];
    int i=0;
    clrscr();
    printf("\n Enter the string:");
    scanf("%s",str);
    while(str[i] != NULL)
    {
        i++;
    }
    printf("\n String length of %s =%d", str, i );
    getch();
}
```

50. Q. Write a program to reverse a string

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char str[10];
    int i=0, j=1;
    clrscr();
    printf("\n Enter the string:");
```

```

scanf("%s",str);
while(str[i] != NULL)
{
    i++;
}
printf("\n Reverse of the string is: " );
for(j = i ; j >=0 ;j -- )
{
    printf("%c", str[ j ] );
}
getch();
}

```

51. Q. Write a program to calculate number of vowels and consonants in the string ®

```

#include<stdio.h>
#include<conio.h>
void main()
{
    char str[10];
    int i= 0,cnt1 =0 ,cnt2=0;
    clrscr();
    printf("\n Enter the string:");
    scanf("%s",str);
    while(str[i] != NULL)
    {
        if ( str[i]==‘a’ || str[i]==‘e’ || str[i]== ‘i’ || str[i]==‘o’ ||str[i]==‘u’)
        {
            cnt1++;
        }
        else
        {
            cnt2++;
        }
        i++;
    }
    printf("\n The string entered is %s ",str);
    printf("\n Vowels=%d",cnt1);
    printf("\n Consonants =%d",cnt2);
    getch();      }

```

52. Q. Write a program to copy one string to another string

```

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

```

```

void main()
{
    char str1[10],str2[10];
    int i= 0;
    clrscr();
    printf("\n Enter the first string:");
    scanf("%s",str1);
    printf("\n Enter the second string:");
    scanf("%s",str2);
    printf("\n Before copying");
    printf("\n first string =%s second string=%s ",str1,str2);
    for(i=0;i<=9;i++)
    {
        str2[i]=str1[i];
    }
    printf("\n after copying");
    printf("\n first string =%s second string=%s ",str1,str2);
    getch();
}

```

53. Q. Write a program to check whether two strings are equal or not®

```

#include<stdio.h>
#include<conio.h>
void main()
{
    char str1[10],str2[10];
    int i= 0,flag=0;
    clrscr();
    printf("\n Enter the first string:");
    scanf("%s",str1);
    printf("\n Enter the second string:");
    scanf("%s",str2);
    for(i=0;i<=9;i++)
    {
        if(str[ i ] == str2[ i ])
        {
            flag=1;
        }
        else
        {
            flag=2;
            break
        }
    }
    if(flag==1)
        printf("The strings are equal");
    else
        printf("The strings are not equal");
    getch();
}

```

```

        }
    }
    if(flag==1)
    {
        printf("\n %s and %s are equal",str1,str2);
    }
    else
    {
        printf("\n %s and %s are not equal",str1,str2);
    }
    getch();
}

```

54. Q. Write a program to check whether entered string is palindrome or not

```

#include<stdio.h>
#include<conio.h>
void main()
{
    char str[10];
    int i=0, j=0 , flag =0;
    clrscr();
    printf("\n Enter the string:");
    scanf("%s",str);
    while(str[i] != NULL)
    {
        i++;      //calculate string length
    }
    i--;
    while( j != i )    //till value of j doesnot become i
    {
        if(str[ i ] == str[ j ])
        {
            flag=1;    // if characters are equal
        }
        else
        {
            flag=2; //if characters are not equal
            break;
        }
        j++ ;
        i--;
    }
}

```

```

if( flag == 1)
{
    printf("%s is palindrome" ,str);
}
else
{
    printf("%s is not palindrome" ,str);
}
getch();
}

```

OR

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char str[10];
    int i=0, len=0 , flag =0;
    clrscr();
    printf("\n Enter the string:");
    scanf("%s",str);
    len = strlen(str);
    for(i=0; i<len; i++)
    {
        if(str[i]==str[len-i-1]
        {
            flag=1;
            break;
        }
    }
    if(flag)
    {
        printf("%s is palindrome" ,str);
    }
    else
    {
        printf("%s is not palindrome" ,str);
    }
    getch();
}

```

55. Q. Write a program to check whether the entered integer is palindrome or not®

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int no, org, rev, rem=0;
    clrscr();
    printf("\nEnter the number");
    scanf("%d",&no);
    org=no;
    while(org!=0)
    {
        rem=org%10;
        rev=rev*10+rem;
        org=org/10;
    }
    if(no==rev)
    {
        printf("\n%d is palindrome number",no);
    }
    else
    {
        printf("\n%d is not a palindrome number",no);
    }
    getch();
}
```

56. Q. Write a program to accept names of 5 students and display them

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char name[5][10];
    int i= 0;
    clrscr();
    for( i=0 ; i<=4; i++)
    {
        printf("\nEnter the string:");
        scanf("%s",name[i]);
    }
    printf("\n List of students");
    for( i=0 ; i<=4; i++)
```

```
{  
    printf("\n %s ",name[i]);  
}  
getch();  
}
```

57. Q. Write a program to demonstrate standard library string functions

```
#include<stdio.h>  
#include<conio.h>  
#include<string.h>  
void main( )  
{  
    char str1[20] = "computer";  
    char str2[10] = "world";  
    clrscr();  
    printf("\n String length of %s is %d ",str1,strlen(str1));  
    printf("\n String length of %s is %d ",str2,strlen(str2));  
    if(strcmp(str1,str2)!=0)  
    {  
        printf("\n %s and %s are not equal ",str1,str2);  
    }  
    strcat(str1,str2,5);  
    printf("\n After concatenation str1 is %s",str1);  
    strcpy(str1,str2);  
    printf("\n After copying str1=%s str2=%s",str1,str2);  
    getch();  
}
```

58. Q.WAP to demonstrate the initialization of structures®

```
#include<stdio.h>
#include<conio.h>
void main()
{
    struct student
    {
        int rollno;
        char name[10];
        float per;
    }s;
    struct student s = { 10,"ashish",89.90 };
    printf("\n student roll no is: %d", s.rollno);
    printf("\n student name is: %s", s.name);
    printf("\n student percentage are: %f ", s.per);
    getch();
}
```

59. Q. WAP to accept structure members from the user and display them. Use book details for the structure®

```
#include<stdio.h>
#include<conio.h>
void main()
{
    struct book
    {
        int bookid;
        char bookname[10];
        float price;
    }b;
    clrscr( );
    // accept details from the user
    printf("\n Enter bookid : ");
    scanf("%d", &b.bookid);
    printf("\n Enter bookname : ");
    scanf("%s", b.bookname);
    printf("\n Enter book price : ");
    scanf("%f", &b.price);
    //display the book details
    printf("\n Book id is : %d", b.bookid);
    printf("\n Book name is : %s", b.bookname);
    printf("\n Book price is : %f", b.price);
```

```
getch();
```

```
}
```

60. Q. WAP to define a structure employee with members empname, empid and salary. Accept data for one employee and display it

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

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

```
void main( )
```

```
{
```

```
    struct employee
```

```
{
```

```
        int empid;
```

```
        char empname[10];
```

```
        float salary;
```

```
    }e;
```

```
    clrscr( );
```

```
// accept details from the user
```

```
printf("\n Enter Employee id : ");
```

```
scanf("%d", &e.empid);
```

```
printf("\n Enter Employee name : ");
```

```
scanf("%s", e.empname);
```

```
printf("\n Enter Employee salary : ");
```

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

```
//display the employee details
```

```
printf("\n Employee id is : %d", e.empid);
```

```
printf("\n Employee name is : %s", e.empname);
```

```
printf("\n Employee salary is : %f", e.salary);
```

```
getch();
```

```
}
```

61. Q. WAP to define a structure employee with members empname, empid and salary. Accept data for two employee and display it®

```
#include<stdio.h>
#include<conio.h>
void main()
{
    struct employee
    {
        int empid;
        char empname[10];
        float salary;
    }e1,e2;
    clrscr( );
    // accept details for one employee from the user
    printf("\n Enter Employee id : ");
    scanf("%d", &e1.empid);
    printf("\n Enter Employee name : ");
    scanf("%s", e1.empname);
    printf("\n Enter Employee salary : ");
    scanf("%f", &e1.salary);
    // accept details for second employee from the user
    printf("\n Enter Employee id : ");
    scanf("%d", &e2.empid);
    printf("\n Enter Employee name : ");
    scanf("%s", e2.empname);
    printf("\n Enter Employee salary : ");
    scanf("%f", &e2.salary);
    //display the employee details
    printf("\n Employee id is : %d", e1.empid);
    printf("\n Employee name is : %s", e1.empname);
    printf("\n Employee salary is : %f", e1.salary);
    printf("\n Employee id is : %d", e2.empid);
    printf("\n Employee name is : %s", e2.empname);
    printf("\n Employee salary is : %f", e2.salary);
    getch();
}
```

62. Q.WAP to declare structure employee having data member name, street, city.

Accept data for three employees and display it®

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    struct employee
    {
        char empname[10];
        char street[10];
        char city[10];
    }e[3];
    int i=0;
    clrscr();
    printf("\n Enter the details of employees");
    for(i=0;i<=2;i++)
    {
        printf("\n Enter the employee name:");
        scanf("%s",e[i].empname);
        printf("\n Enter the employee street:");
        scanf("%s",e[i].street);
        printf("\n Enter the employee city:");
        scanf("%s",e[i].city);
    }
    printf("\n Details of employees are");
    for(i=0;i<=2;i++)
    {
        printf("\n The employee name: %s",e[i].empname);
        printf("\n The employee street:%s",e[i].street);
        printf("\n The employee city:%s",e[i].city);
    }
    getch();
}
```

63. Q. WAP to accept structure members from the user and display the details of 5 books. Use book details for the structure®

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    struct book
    {
        int bookid;
        char bookname[10];
        float price;
    }b[5];
    int i=0;
    clrscr( );
    for(i=0;i<=4;i++)
    {
        printf("\n Enter bookid : ");
        scanf("%d", &b[i].bookid);
        printf("\n Enter bookname : ");
        scanf("%s", b[i].bookname);
        printf("\n Enter book price : ");
        scanf("%f", &b[i].price);
    }
    for(i=0;i<=4;i++)
    {
        printf("\n Book id is : %d", b.bookid);
        printf("\n Book name is : %s", b.bookname);
        printf("\n Book price is : %f", b.price);
    }
    getch();
}
```

64. Q. Write a program to demonstrate nested structure®

```
#include<conio.h>
#include<stdio.h>
void main()
{
    struct date
    {
        char day[10];
        int month;
        int year;
    };
    struct account
    {
        char name[10];
        int accno;
        float balance;
    };
    struct date d = {"Monday",1,2018};
    struct account a = {"abhi",1055,15000.00};
    printf("\n Day is: %s",d.day);
    printf("\n Month is %d",d.month);
    printf("\n Year is %d",d.year);
    printf("\n Name is %s",a.name);
    printf("\n Accno is %d",a.accno);
    printf("\n Balance is %f",a.balance);
    getch();
}
```

65. Q. Write a program to demonstrate typdef

```
#include<conio.h>
#include<stdio.h>
typedef struct employee
{
    char name[50];
    int salary;
}emp;
void main( )
{
    emp e1;
    printf("\nEnter Employee name:\t");
    scanf("%s", e1.name);
    printf("\nEnter Employee salary: \t");
    scanf("%d", &e1.salary);
    printf("\n Employee name is %s", e1.name); printf("\nSalary is %d", e1.salary);
    getch();
}
```

66. Q. Write a program to demonstrate enum

```
#include<conio.h>
#include<stdio.h>
enum months { jan , feb, mar, apr, may};
int main()
{
    months m1 ,m2;
    m1=jan;
    m2= apr;
    int diff = m2- m1;
    printf("months between %d",diff);
    if(m1 > m2)
        printf("\n m2 comes before m1");
    getch();
}
```

67. Q. Write a program to add 2 numbers using function

```
//function with no argument and no return value
#include<conio.h>
#include<stdio.h>
void main()
{
    void add(void);
    clrscr();
    add();
    getch();
}
void add()
{
    int a , b ;
    printf("\n Enter value of a and b:");
    scanf("%d %d",&a,&b);
    printf("\n The addition of %d + %d =%d",a,b,a+b);
}
```

68. Q. Write a program to calculate area of circle using function®

```
//function with no argument and no return value
#include<conio.h>
#include<stdio.h>
const float pi=3.14; //global variable declaration
void area(void); //global function declaration
void main()
{
    clrscr();
    area(); //function call
    getch();
}
void area() //function definition
{
    float radius;
    printf("\n Enter radius of circle:");
    scanf("%f",&radius);
    printf("\n Area of circle is %f",pi*radius*radius);
}
```

69. Q. Write a program to check whether the number is even or odd

```
//function with no argument and no return value
#include<conio.h>
#include<stdio.h>
```

```

void main()
{
    void evenodd(void); //function declaration
    clrscr();
    evenodd(); //function call
    getch();
}
void evenodd() //function definition
{
    int no;
    printf("\n enter the number:");
    scanf("%d",&no);
    if(no%2==0)
    {
        printf("%d is even",no);
    }
    else
    {
        printf("%d is odd",no);
    }
}

```

70. Q. Write a program to print odd numbers from 11-50

```

//function with no argument and no return value
#include<conio.h>
#include<stdio.h>
void oddprint(void); //function declaration
void main()
{
    clrscr();
    oddprint(); //function call
    getch();
}
void oddprint()
{
    int i=1;
    printf("\n Odd numbers from 1 to 50:");
    for( i=1; i<=50; i=i+2)
    {
        printf("%d\t",i);
    }
}

```

71. Q. Write a program to print the square of the number till the position entered by the user

```
//function with argument and no return value
#include<conio.h>
#include<stdio.h>
void square (int); //function declaration with one integer argument
void main()
{
    int position=0;
    printf("\n Enter the last position:");
    scanf("%d",&position);
    square(position); //function call with actual argument position
    getch();
}
void square( int x) //receiving position as x in function definition
{
    int i=1;
    for(i=1;i<=x; i++)
    {
        printf("%d\t ", i * i);
    }
}
```

72. Q. Write a program to accept a number and print whether it is prime or not

```
//function with argument and no return value
#include<conio.h>
#include<stdio.h>
void prime(int); //function declaration with argument
void main()
{
    int no;
    clrscr();
    printf("\n Enter the number:");
    scanf("%d",&no);
    prime(no); //function call with actual argument no
    getch();
}
void prime(int n) //receiving number as n in function definition
{
    if(n==3 || n==5 || n==7)
    {
        printf("\n %d is prime",n);
    }
}
```

```

        }
    else
    {
        if(n % 2 > 0 && n % 3 >0 && n % 5 >0 && n%7 > 0)
        {
            printf("\n %d is prime",n);
        }
        else
        {
            printf("\n %d is not prime",n);
        }
    }
}

```

73. Q. Write a program to accept a character from the user and display its case

```

//function with argument and no return value
#include<conio.h>
#include<stdio.h>
void charcase(char); //function declaration with argument
{
    void main()
    {
        char code;
        clrscr();
        printf("\n Enter the character:");
        scanf("%c",&code);
        charcase(code);      //function call with actual argument no
        getch();
    }
    void charcase(char x) //receiving number as n in function definition
    {
        if ( x >=65 && x<=91)
        {
            printf("\n %c is in upper case",x);
        }
        else if(x> =97 && x<=122)
        {
            printf("\n %c is in lower case",x);
        }
        else
        {
            printf("\n entered value is not character");
        }
    }
}

```

74. Q. Write a program to display sum of only prime numbers from 11-50

```
//function with no argument and return a value
#include<conio.h>
#include<stdio.h>
int primeadd(void); //function declaration
void main()
{
    int ans;
    clrscr();
    ans=primeadd(); //function call
    printf("\n Addition of prime numbers from 1 to 50 = %d",ans);
    getch();
}
int primeadd()          //function definition
{
    int i =1,sum=0;
    printf("\n Prime numbers are:\n");
    for( i= 1 ;i<= 50; i++ )
    {
        if ( i == 3 || i == 5 || i == 7) //checking numbers in between 3 , 5 ,7
        {
            printf("%d\t ", i );
            sum = sum + i ;
        }
        if(i % 2 > 0 && i % 3 >0 && i % 5 >0 && i%7 > 0)
        {
            sum = sum + i ;
            printf("%d\t ", i );
        }
    }
    return sum;
}
```

75. Q. Write a program to calculate string length using function

```
//function with no argument and return a value
#include<conio.h>
#include<stdio.h>
int strlen(); //function declaration
void main()
{
    int len = 0;
    clrscr();
```

```

len= strlength(); // function call
printf("\n String length = %d ",len );
getch();
}
int strlength() // function definition
{
    int i =0;
    char str[10];
    printf("\n Enter the string:");
    scanf("%s",str);
    while( str[i] !=NULL)
    {
        i++ ;
    }
    return i;
}

```

76. Q. Write a program to calculate factorial of a number using function

```

//function with no argument and return a value
#include<conio.h>
#include<stdio.h>
void main()
{
    long int fact(void); //function declaration
    clrscr();
    printf("\n The factorial of number %ld ",fact( ) ); //function call
    getch();
}
long int fact(void) //function definition
{
    int no;
    long int fact=1,i=1;
    printf("\n Enter the number:");
    scanf("%ld",&no);
    for( i =1 ; i<=no; i++ )
    {
        fact = fact * i ;
    }
    return fact;
}

```

77. Q. Write a program to find sum of digits using function

```
//function with arguments and return a value
#include<conio.h>
#include<stdio.h>
int digitsum(int); //declaration of function with return value and with argument
void main()
{
    int no;
    clrscr();
    printf("\n Enter the number:");
    scanf("%d",&no);
    printf("\n Sum of digits of %d",digitsum(no)); //function call
    getch();
}
int digitsum (int num) //function defination
{
    int rem=0,sum =0;
    while (num !=0)
    {
        rem=num % 10;
        sum = sum + rem;
        no=  no/10;
    }
    return sum;
}
```

78. Q. Write a program to accept values of 3 angles and check whether it is a triangle or not

```
//function with arguments and return a value
#include<conio.h>
#include<stdio.h>
int triangle(int,int,int); //declaration of function
void main()
{
    int a1,a2,a3,total;
    clrscr();
    printf("\n Enter values of 3 triangles:");
    pcanf("%d %d %d",&a1,&a2,&a3);
    total = triangle(a1,a2,a3); // function call
    if( total == 180)
    {
        printf("\n this is a triangle");
    }
    else
    {
        printf("\n this is not a triangle");
    }
    getch();
}
int triangle(int x,int y,int z)
{
    return (x+y+z);
}
```

79. Q. Write a program to find factorial of a number using recursion® ® ®

```
//function with arguments and return a value
#include<conio.h>
#include<stdio.h>
long int fact(int); //function declaration
void main()
{
    int no;
    clrscr();
    printf("\n Enter the number:");
    scanf("%d",&no);
    printf("\n The factorial of number %ld ",fact(no)); //function call
    getch();
}
long int fact(int n)
{
    long int f;
    if( n==1)
    {
        return 1;
    }
    else
    {
        f=n*fact(n-1);
        return (f);
    }
}
```

80. Q. Write a program to swap the numbers using call by value® ®

```
//function with arguments and no return value
#include<conio.h>
#include<stdio.h>
void swap(int, int);
void main()
{
    int x, y;
    printf("Enter the value of x and y\n");
    scanf("%d%d", &x, &y);
    printf("Before Swapping\nx = %d\ny = %d\n", x, y);
    swap(x, y);
    printf("After Swapping\nx = %d\ny = %d\n", x, y);
    return 0;
}
void swap(int a, int b)
{
    int temp;
    temp = b;
    b = a;
    a = temp;
}
```

81. Q. Write a program to add two numbers using call by reference

```
#include<stdio.h>
#include<conio.h>
int add(int *, int *); //declaring function
void main()
{
    int a,b;
    clrscr();
    printf("\n enter value for a and b:");
    scanf("%d %d ", &a, &b);
    printf("\n Addition = %d",add(&a,&b)) //call by reference
    getch();
}
int add(int *x, int *y) //function defination
{
    return *x + *y;
}
```

82. Q. Write a program to swap two numbers using call by reference

```
#include<stdio.h>
#include<conio.h>
void swap(int *, int *); //declaring function
void main()
{
    int a,b;
    clrscr();
    printf("\n enter value for a and b:");
    scanf("%d %d ",&a,&b);
    printf("\nBefore swapping a: %d \t b: %d",a,b);
    swap(&a,&b); //function call
    printf("\n After swapping a: %d \t b:%d",a,b);
    getch();
}
void swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
```

83. Q. Write a program to find product of two numbers using pointers

```
#include<stdio.h>
#include<conio.h>
void main()
{
int no1,no2;
int *p1,*p2 ; //pointer declaration
clrscr();
printf("\n Enter the value for no1 and no2:");
scanf("%d %d ",&no1,&no2);
p1= &no1; //assigning address of no1 to p1
p2=&no2; // assigning address of no2 to p2
printf("\n %d * %d =%d" ,*p1,*p2,(*p1)*(*p2));
getch();
}
```

84. Q. Write a program to print values of variables and their address®

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=5,b=10;
clrscr();
int *p1,*p2;
p1=&a;
p2=&b;
printf("\n Value of a=%d",a);
printf("\n Address of a=%u",p1);
printf("\n Value of b=%d",b);
printf("\n Address of b=%u",p2);
getch();
}
```

85. Q. Write a program to swap two values using pointer

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
int *p1,*p2,temp=0;
clrscr();
```

```

printf("\n Enter value for a and b:");
scanf("%d %d",&a, &b);
printf("\n Before swapping a=%d b=%d", a,b);
p1=&a; //assigning address to p1
p2=&b; //assigning address to p2
temp = *p1;
*p1=*p2;
*p2=temp;
printf("\n After swapping: a=%d and b=%d", a, b);
getch();
}

```

86. Q. Develop a program to swap two numbers using pointer and add swapped numbers also print their addition.

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int a,b,*p1,*p2,temp=0,add=0;
    clrscr();
    printf("\nEnter the value of a:");
    scanf("%d",&a);
    printf("\nEnter the value of b:");
    scanf("%d",&b);
    p1=&a;
    p2=&b;
    printf("\nValues of a and b before swapping are %d and %d",a,b);
    temp = *p1;
    *p1=*p2;
    *p2=temp;
    printf("\nValues of a and b after swapping are %d and %d",a,b);
    add = *p1+*p2;
    printf("\nAddition of a and b is %d",add);
    getch();
}

```

87. Q. Write a program to accept two numbers from user and perform addition, subtraction, multiplication and division operations using pointer. ®

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int no1,no2,*ptr1,*ptr2,result;
    clrscr();
    printf("Enter no1:");
    scanf("%d",&no1);
    printf("\nEnter no2:");
    scanf("%d",&no2);
    ptr1=&no1; //assigning address of no1
    ptr2=&no2; //assigning address of no2
    result=*ptr1+*ptr2;
    printf("\n Addition=%d",result);
    result=*ptr1-*ptr2;
    printf("\n Subtraction=%d",result);
    result=*ptr1**ptr2;
    printf("\n Multiplication=%d",result);
    result=*ptr1/(*ptr2);
    printf("\n Division=%d",result);
    getch();
}
```

88. Q. Write a program to swap two numbers using call by reference and pointer with function®

```
#include<stdio.h>
#include<conio.h>
void swap(int *, int *); //declaring function
void main()
{
    int a,b;
    clrscr();
    printf("\n enter value for a and b:");
    scanf("%d %d ",&a,&b);
    printf("\nBefore swapping a: %d \t b: %d",a,b);
    swap(&a,&b); //function call
    printf("\n After swapping a: %d \t b:%d",a,b);
    getch();
}
```

```

void swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

```

89. WAP to add numbers till the position entered and display result by call by reference function®

```

int add( int *); //function declaration with pointer as an argument
void main()
{
    int position;
    int *ptr;      //pointer declaration
    clrscr();
    printf("\n Enter the position:");
    scanf("%d",&position);
    ptr=&position;
    printf("Sum = %d", add(ptr)); //function call
    getch();
}

int add( int *p)
{
    int i=0;
    int sum=0;
    for(i=1; i<=(*p); i++)
    {
        sum =sum + i ;
    }
    return sum;
}

```

90. Q. WAP to accept array elements and print them using pointer

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int no[5],i=0;
    int *ptr; // pointer declaration
    clrscr();
    for(i=0;i<=4;i++)
    {

```

```

printf("\n enter the number:");
scanf("%d",&no[i]);
}
for(i=0;i<=4;i++) // print array elements
{
ptr = &no[i];
printf("\n %d is stored at %u",*ptr, ptr);
}
getch();
}

```

91. Q. Write a program to find length of string using pointer and functions

```

#include<stdio.h>
#include<conio.h>
int length(char *); //declaration of function with char pointer
void main()
{
    char str[10];
    char *ptr;
    clrscr();
    printf("\n Enter the string:");
    scanf("%s",str);
    ptr=&str[0]; //assigning base address to the pointer
    printf("\n length of string = %d",length(ptr));
    getch();
}
int length(char *p)
{
    int cnt=0;
    while(*p!=NULL)
    {
        cnt++;
        p++;
    }
    return cnt;
}

```

92. Q. Write a program to compute sum of all elements stored in an array using pointer®

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int no[5],i=0,sum=0;
    int *ptr;
    clrscr();
    for(i=0;i<=4;i++)
    {
        printf("\n enter the number:");
        scanf("%d",&no[i]);
        ptr=&no[i];
        sum= sum + (*ptr);
    }
    printf("\n Sum of array numbers =%d",sum);
    getch();
}
```

93. Q. Write a program to demonstrate Array of pointer

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char *str[5]={"abc", "pqr","xyz","rgb","nmk"};
    int i=0;
    clrscr();
    for(i=0;i<=4;i++)
    {
        printf("\n %d element=%s",i,str[i]);
    }
    getch();
}
```

94. Q. Write a program to demonstrate structure with pointer

```
#include<stdio.h>
#include<conio.h>
void main()
{
```

```

struct movie
{
    char moviename[20];
    char actorname[10];
    int year;
};

struct movie *ptr;
clrscr();
printf("\n Enter movie name:");
scanf("%s",ptr->moviename);
printf("\n Enter movie actor:");
scanf("%s",ptr->actorname);
printf("\n Enter release year:");
scanf("%d",&ptr->year);
printf("\n Movie name is %s",ptr->moviename);
printf("\n Actorname is %s",ptr->actorname);
printf("\n Release year is %d",ptr->year);
getch();
}

```

95. WAP to read the n numbers of values in an array and display it in reverse order

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10],i,n;
    clrscr();
    printf("\n Enter number of elements:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("\n Enter array element:");
        scanf("%d",&a[i]);
    }
    printf("\n Array elements in reverse order:");
    for(i=n-1;i>=0;i--)
    {
        printf("\t%d",a[i]);
    }
    getch();
}

```

## EXTRA QUESTIONS

### 1. What is scope and lifetime of variables?

Scope of Variable defines in which part of the program the variable is actually available for use.

The lifetime of variables is a period in which the variable holds the given value at the time of program execution.

### 2. State the functions under math.h header file®

- **sqrt( )** :- It is used to find out square root of an integer.
- **abs( )** :- It is used to find out absolute value of an integer
- **sin ( )**:- It is used to compute the sine value of an input value.
- **cos ( )**:- It is used to compute the cosine value of an input value
- **pow ( )**:- it is used to compute the power of a input value.
- **floor ( )** :- it is used to round down the input value.
- **ceil ( )** :- it is used to round up the input value.

### 3. Define Multi-dimensional Array®

An array with more than one dimension is known as multidimensional array.

eg : float x[3][4];

We can also declare 3 dimensional array

eg: int a[3][3][3];

### 4. State the advantages of pointer?®

- it reduces the length and complexity of the program
- it increases the execution speed
- it enables us to access the variable that is defined outside the function
- they are more efficient in handling data tables
- it supports dynamic memory management

### 5. How to pass pointer to a function? ®

When pointer (address) is passed to the function as an argument instead of value then function is called as call by reference.

### 6. Write any 4 library functions under conio.h header file? ®

**clrscr()** – it is used to clear the output screen

**getch()** – it reads the character from the keyboard

**getche()** – it reads the character from the keyboard and echoes it on the o/p screen

**putch()** – writes the character directly to the console

**textcolor()** – it is used to change the text color

**textbackground()** – this function is used to change the text background

### 7. What are the library functions under stdio.h header file?

**getchar()**

**putchar()**

**printf()**

**scanf()**

8. Explain how formatted input can be obtain? ®  
when the input data is arranged in a specific format, it is called formatted input. scanf() is used for this purpose in C

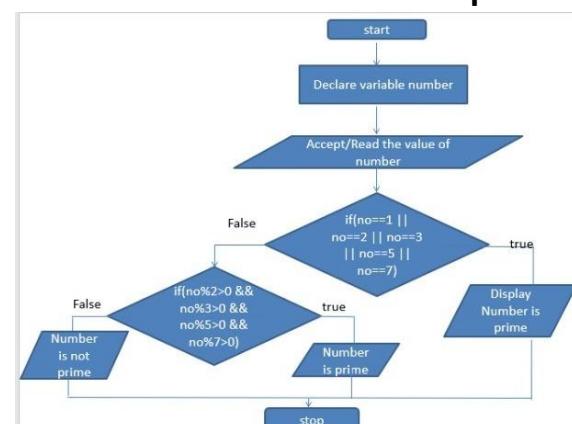
**General syntax:**

```
scanf("format specifier",&variable name);
```

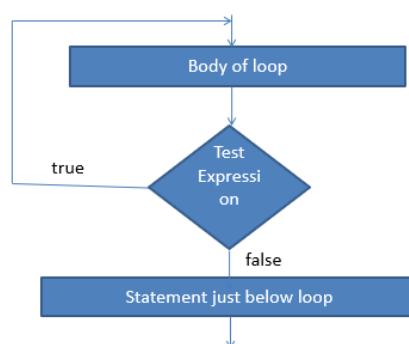
Eg:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a;
    clrscr();
    printf("\nEnter a number:");
    scanf("%d",&a);
    printf("\nEnterd number is %d",a);
    getch(); }
```

9. Flowchart to check whether the entered number is prime or not® ®



10. Draw the flowchart for do-while loop and write a program to add numbers until user enters zero®



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

```

void main()
{
    int n, sum=0;
    clrscr();
    do
    {
        printf("\nEnter number:");
        scanf("%d",&n);
        sum = sum+n;
    }while(n!=0);
    printf("\nSum of entered numbers is %d",sum);
    getch();
}

```

**11. List any 4 keywords with their use in C®**

**int** – used for declaration of integer data type  
**float** – used for declaration of float data type  
**char** – used for declaration of character data type  
**double** – used for declaration of double data type

**12. State any two difference between Array and String®**

<b>Array</b>	<b>String</b>
<b>It is collection of similar data type</b>	<b>It is a collection of characters</b>
<b>Array can be of any data type like char, int, float etc</b>	<b>String can contain only characters.</b>

**13. Difference between Call by Value and Call by reference® ®**

<b>Call by Value</b>	<b>Call by reference</b>
<b>A copy of actual arguments is passed to the formal arguments</b>	<b>The address of the actual arguments is passed to the formal arguments</b>
<b>The address of actual and formal arguments is different</b>	<b>The address of actual and formal arguments is same.</b>

**14. Difference between character array and integer array with respect to size and initialization®**

<b>Parameter</b>	<b>Character array</b>	<b>Integer array</b>
<b>Size</b>	<b>character array is filled with '\0' so the array size should be so declared that it should have one last location for '\0' character.</b>	<b>No extra location than the number of elements is required.</b>
<b>Initialization</b>	<b>Initialization can be done like : char str[4]={'a','b','c','\0'}; char str[4]="abc";</b>	<b>Initialization can be done like : int arr[4]={1,2,3,4};</b>

**15. Write a program to print reverse of an entered string using pointer®**

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

```

#include<conio.h>
void main()
{
    char str[10],*ptr;
    int l=0;
    clrscr();
    printf("Enter string:");
    scanf("%s",str);
    ptr=str;
    while(*ptr!='\0')
    {
        l=l+1;
        ptr=ptr+1;
    }
    while(l>0)
    {
        ptr=ptr-1;
        printf("%c",*ptr);
        l=l-1;
    }
    getch();      }

```

**16. How to use pow() function of <math.h> header file®**

**pow()** is used to compute the power of an input value

**syntax:** double pow(double x, double y);

**17. Explain user defined functions with example ®**

**Functions are the basic building blocks of the C program, they can be either predefined or user defined.**

**Predefined functions are those which are already available in C library.**

**User defined functions are those which are written by users to complete a specific task.**

Eg: #include<stdio.h>  
# include<conio.h>  
void myFunc(int a)  
{  
 printf("The value is %d",a);  
}  
void main()  
{  
 myFunc(10);  
 getch();  
}



"Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the universe trying to produce bigger and better idiots. So far, the universe is winning."



**C h i r a g   F e r w a n i**