PAGE NO.
DATE :

(1) What is C language? Explain its feature and application.

→ C language is a Procedural Programing language. It is an example of 3rd generation High level Programing language. It is also Called Modular Programing language. C language was developed by Dennis Ritche in 1972 at Bell telephone laboratory in U.S.A. C language developed from BPCL (Basic combined Programing language), latter known as B language.

C language used to design any Computer application or System software and Other softwares.

C language become Popular due to its high level language Quality. that is

C language was approved by ANSI (American National Standard Institute) in December 1989 and renamed as ANSI-C. C language has Simplicity of High level language as well as Power of low level language.

C language provide the facilities to do Programing at Register level also.

features and applications of C language are as:-

a) C language has Only 32 Keywords. So it is easy to remember and to make Software or Program using it.

b) C language has large no. of built in functions and Operators that can be used to write any Complex problem.

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

built-in function help the programmer to write less no. of code.
~~i.e Save unnecessary memory space during programming.~~

c) C-language is heighly portable language so that program written in C language can run on another computer without modification or little modification.

d) C language used can be used for low level as well as high level language programming. Hence it bridges the gap of low level language and high level language.

e) C is a core language as many other programing language like C++, Java, pearl etc. are based on C.

f) C is an extensible language as it enables to user to add his own functions to the C-library so that the programmer make functions of frequently needed codes to save time and less code has to be written.

g) It support pointer to refer the computer memory which helps the in accessing array and structures so that the variable can be accessed in other functions using pointer. It is called call by reference in C programing.

Scanned with CamScanner

Scanned with CamScanner

Application of C language

1) C language is used for creating computer applications
2) It is used in writing embeded software.
3) It is also used in developing verification, software, test code etc.
4) To develop application software like database and spread sheets.
5) for develop graphical related application like computer and mobile games.
6) C Programming language can be used to design the compilers.
7) UNIX kernal is completely developed in C language.
8) C programing language can be used to design network devices.

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

(2) Describe the basic data type available in C.

→ Data types:- C language is rich in its data type. the variety of data type available in C allows the Programmer to Select the appropriate type according to the need of application. ANSI C Supports four Classes of datatypes.

Basic datatype :- The datatype which represents the fundamental data i.e characters and are already defined to the Compiler is known as Basic data type. Integers, Characters and float are Basic datatype in C language. Some of these datatypes are further divided into different Categories like Signed, unsigned Short, long etc.

The following chart represents the Classification of basic datatype and their Size and Symbol.

Type	Size (byte)	Symbol
char (Signed char)	1	%c
unsigned char	1	%c
Short int	1	%d
unsigned Short int	1	%u
Signed int	2	%d
unsigned int	2	%u
Signed long int	4	%ld
unsigned long int	4	%lu
float	4	%f
double	8	%lf
long double	10	%lf

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

(3) What are the characteristics of C language?
→ C is a Programing language that has been developed by dennis ritchie in 1970's.

1) Simple and Clear:- C has richest Collection of data types (int, float, char), Operators and inbuilt functions.

ii) efficiency:- A Program written in C language executes at high speed with efficient memory utilizations.

iii) Modularity:- A large Program in C language Can be devided into different identifiable Parts Called modules, Hence C language Supports modularity.

iv) Portable - C has the characteristics of Portability. Portable means Computer independent. The programs Made in C on One Computer Can be run on different Computers by not changing anything.

v) Easily available and Quick:-
The C Compiler are Very easily available and require Very less disk Space.

vi) Case Sensitive :- Case Sensitive means that almost everything in C program is written is Small case.

vii) large no. of data types:-
C Programing is rich in data types that Help the Programmer to Save Computer memory space using appropriate data type in Program.

Scanned with CamScanner
Scanned with CamScanner

PAGE NO.
DATE :

(4) Discuss the different type of decision making and Branching Statement in C with Syntax and example.

→ The decision making Statement is a Condition Control Statement. where we can Check - Condition, If condition is true then execute the next Statement otherwise terminate or leave the next Statement.

There are 6 types of decision making and branching Statement

i) If --
ii) If - else
iii) nested If else
iv) else-If ladder
v) Condition Statement
vi) Switch- Case

Syntax and example :-

* If ⇒ If (condition)
{
true Statement;
}

* Example:- If (x > y)
{
Printf (" %d is large ", x);
}

* If-- else ⇒ If (condition)
{
true statement;
}

Scanned with CamScanner

Scanned with CamScanner

```
else
{
    false statement;
}
```

* **example :-**

```
if (x>y)
printf("%d is large", x);
else
printf("%d is large", y);
```

Nested if else ⇒

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

* **example :-**

```
if (x>y) → if (x>z)
printf("%d is large", x);
else
printf("%d is large", z);
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

```
else
    if (y>z)
    printf("%d is large", y);
    else
        printf("%d is large", z);
```

* **else - if ladder** ⇒

```
if (condition -1)
{
    Statement;
}
else if (condition -2)
{
    Statement;
}
else if (condition -3)
{
    Statement;
}
else
{
    Statement;
}
```

* **Example** :-

```
if (n=1)
    printf("One");
else if (n=2)
    printf("two");
else if (n=3)
    printf("three");
else
    printf("wrong code");
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

- Condition Statement

(Condition ? true Statement : False statement)

* example :-

```
x>y ? Printf (":%d is large",x) :
      Printf (" %d is large", y);
```

* Switch case :-

```
Switch (expression)
{
    Case 1:
    {
        Statement;
    }
    Case 2:
    {
        Statement;
    }
    Case 3:
    {
        Statement;
    }
    Case 4:
    {
        Statement;
    }
    default
    {
        statement;
    }
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

* example

```
Switch (x)
{
    Case 1:
        printf ("green");
    Case 2:
        Printf ("red");
    Case 3:
        Printf ("Yellow");
    default
        Printf ("wrong code");
}
```

(5) What is loop ? Distinguish between for loop and do - while with example?

→ A loop tests certain condition and then directs the repeated execution of the statement contained in the body of loop.

It allows to repeate the given statements no. of times.

A loop is a fundamental programing ideas that is commonly used in writing programs.

Difference between for loop and do while :-

do while	for

1) Syntax :- do
{ statement;
}
while (condition);
	1) Syntax :-
	for (initilization; condition; incr / decr)
	{
	statement;
	}

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

(2) It is exit Controlled loop.	(2) It is entry Controlled loop.
(3) If 1st time Condition is not true do not enter in a loop.	(3) It enter in the loop even when the 1st time condition is not true.
(4) Initilization and Increment, decrement is not Part of Syntan.	(4) Initilization and Increment, decrement is part of Syntan.

Example of do while loop

```
Void main
{
int n, i=1;
Clrscr();
Printf ("enter no");
Scanf ("%d", &n);
do
{
Printf ("\n %d", i*n);
i++;
} while (i<=10);
getch();
}
```

Example of for loop:-

```
# Include <Stdio.h>
# Include <Conio.h>
Void main()
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

```
{
    int n, i;
    clrscr();
    Printf("enter no");
    scanf("%d", &n);
    for (i=1; i<=10; i++)
    {
        Printf("\n %d", i*n);
    }
    getch();
}
```

(6) What is a pointer? explain its use with an example.

→ Pointer is a variable declaration method use for store address of any same type variable.

Pointer use for referencing purpose. In C-language using pointer we can access any variable by using multiple methods.

The pointer variable is an unsigned integer value. It is never fractional value in any condition. Pointer represent a storage location in memory. A pointer is derived data type in C. It is also known as referencial variable. It reduce length and complexity of program. It increases the execution speed and thus reduce the program execution time. Generally we can denote

Scanned with CamScanner

PAGE NO.
DATE :

Pointer by * Symbol. Pointer variable consume 2-byte memory space for any data type.

uses of Pointer:-

1) Pointer are used to pass parameter by reference. so that we can access the variable in other function using pointer called call by reference.

2) This is useful for ~~[illegible]~~ returning multiple values from a function. C programming do not provide the multiple returning ~~valu~~ facility but by the use of pointer we can return multiple values from a function.

3) Pointer can also be used to allocate and delocate dynamic variables and arrays in memory.

4) using pointer we access array elements.

5) By the use of pointer we reduce the execution time of the program.

6) We can also reduce the storage space and complicity of program by using pointer.

7) pointer helps us to built complex data structures like linked lists, queues, stacks, trees, graphs etc.

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

Example of Pointer :- Swap two numbers using pointer

```
# Include <Stdio.h>
# Include <Conio.h>
Void Swap (int *, int *);
Void main ()
{
    int x, y;
    Clrscr ();
    Printf(" enter two value x and y");
    Scanf ("%d %d", &x, &y);
    Printf ("before Swaping x = %d and
            y = %d", x, y);
    Swap (&x, &y);
    Printf ("\n after Swaping x = %d
            and y = %d", x, y);
    getch ();
}
Void Swap (int *m, int *n)
{
    int t;
    t = *m;
    *m = *n;
    *n = t;
}
```

(7) What are C keyword? List any three keywords with their use.

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

Keywords are sequence of characters whose meaning has already been explained to the C compiler. Keywords can't be use as variable names. The keywords are also called reserved words. These keywords can be used only for their specified purpose. They can't be used as programmer defined identifiers. The keywords are written in lower case. There are 32 keywords which are follows :- auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

i) int :- It is a keyword in C which is a primary or basic data type.
It consume 2- byte memory space.
It is use to declare variable of integer type values.
ex:- int P = 10;

ii) if :- The if — statement is a powerful decision making statement and is used to control the flow of execution of statements. The general form of a simple if statement is as follows.

```
if ( test expression)
{
    Statement block;
}
Statement X
```

Scanned with CamScanner
Scanned with CamScanner

iii) * for :- The for loop is an entry controlled loop which has more concise loop control structure.
The general form of for loop is as follows:-

```
for ( initilization ; conditional check ; incr/ decr)
{
    body of loop;
}
Statement - x;
```

(8) What are formatted and unformatted functions? give examples.

Scanned with CamScanner

Formatted I/O

Functions used for input and output, which must also define the format of the input or output value, are called Formatted input output function.

```
scanf(), printf()
```

The printf () function is an output formatted function, and its scanf () is an input formatted function, in which scanf () is inputted into this type of data type, It is mandatory to create a formatted form specifier for each data type.

```
int a; scanf ("%d", &a);
```

In the above the value input to a is being done by scanf () a. An integer variabale so the function specifier% d of the integer data type is also implied in the function 'c' programming contains the following

Scanned with CamScanner

the function 'c' programming contains the following data type specifier. like :-

Specifier	Data type
%d	int
%c	character
%f	float
%s	string

The printf () function is a formatted output function whereby the output of the value of all types is printed. In this function also the format (type specifier) of the value to be output is defined.

```
int a = 50; printf ("% d", & a);
```

In the above example a is an integeral variable whose value is 20 The printf () function is used to get the output of a value of a and the integer type specifier% d is also used.

Scanned with CamScanner

6 Formatted I/O Examples:

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

void main()
{
 int a;
 clrscr();
 printf("Enter value of a
 scanf("%d", &a);
 printf(" a = %d", a);
 getch();
}
```

Output of the above program is:

```
Enter value of a:5↲
a = 5
```

Scanned with CamScanner

Unformatted I/O

Functions used for input and output, in which the format of the input or output value is not defined, are called Unformatted input output functions.

Syntax:-

```
getchar();
          getche();
          gets();
          putchar();
          puts();
```

getchar() This is an unformatted input function. This function is used to input a character value. This function can input a character value so there is no need to define the format of the value being input in this function.

Syntax:-

```
char a; a=getchar();
```

Scanned with CamScanner

6 Unformatted I/O Examples:

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

void main()
{
 char ch ;
 clrscr();
 printf("Press any charac
 ch = getche();
 printf("\nYou pressed :'
 putchar(ch);
getch();
}
```

Output of the above program is:

```
Press any character: L
You pressed: L
```

Scanned with CamScanner

PAGE NO.
DATE :

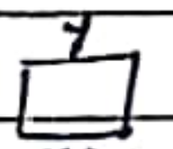
(9) Explain the terms Variables and Constants. How many types of Variables are Supported by C ?

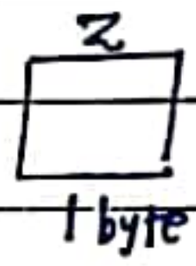
→ Variable :- Variable Create memory block on which we can store any value. Variable memory block Size depends on data type.

Syntax:

< data type > < Variable Name >

eg:- int x;

float y; 

char z; 

a) int :- It is a Keyword in C which is a Primary or basic data type. It Consume 2- byte memory Space. It is use to declare Variable of integer type values. ex:- int p = 10;

b) float :- It is a Keyword in C which is a Primary or basic data type. It Consume 4- byte memory Space. It is use to declare Variable of float Real type values. ex:- float y = 3.5;

c) char :- It is a Keyword in C which is a Primary or basic data type. It Consume 1 - byte memory Space. It is use to declare Variable of Character type.

ex:- char A;

PAGE NO.
DATE :

* Constants :- It is also called "literals". Constant represent fix values.
There are several types of constants, use with C language.

Types of constants :-

i) Integer constants → 10, 20, 50 etc.
ii) Real constants → 2.66, 3.44 etc.
iii) Character constants → 'A', 'x', '+' etc.
iv) String constants → "ABC", "XYZ" etc.
v) Boolean constants → 0, 1, true, false etc.

(10) What do you mean by decision making statements? explain if - else and switch case statements.

→ The decision making statement is a condition control statements. where we can check condition. If condition is true then execute the next statement otherwise terminate or leave the next statement.

* if - else :- The if --- else statement is two way decision making statement. The general form of a simple if --- else statement is as follows:-

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

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

Example:-

```
if (x>y)
    Printf ("%d is large", x);
else
    Printf ("%d is large", y);
```

In if --- else Statement, the test expression is evaluated first if it is found to be true, the true Statement block is executed and the false Statement block is skipped. Otherwise, the false Statement block is executed and true Statement block is skipped.
In no case both the blocks executed and in no case both the blocks are skipped.
The Statement block may have one or more than one executable Statements. The test expression may be relational or logical expression.

* Switch case Statement:-

The Switch -- case Statement tests the value of a given expression / variable against a list of case values and when a match is found, a block of Statements associated with that case is executed.

The general form of Switch -- case Statement is as follows:-

```
Switch ( expression)
{
    Case 1:
    {
        Statement;
    }
```

Scanned with CamScanner

Scanned with CamScanner

Case 2:
{
Statement;
}
Case 3:
{
Statement;
}
Case 4:
{
Statement;
}
default
{
Statement;
}

When the switch is executed then the value of the expression is successfully compared against the case value 1, case value 2 ---, case value -n.
The switch expression must be of integral type.
Case labels must be a constant or a constant expression.
Case label must be unique.
There can be at most one default label in a switch case statement.

(11) What is string? write down any 5 string function with syntax and example.

→ Strings are actually 1-D array of character, terminated by null character that is \0.

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

String use for store any word or sentence. where String Variable stored maximum (Size - 1) character. String variable aur automatically stored null character. where null represent end of String. String use double code symbol and store character in ASCI formate. We can use gets to input the String and puts to display the String

Syntax :-

Char <String Name> [Size];

5 String function with Syntax and example are as :-

1) Strlen (str) :- This function is used to return length of given String or count no. of character in the String. Strlen function use String.h headerfile

Syntax :-

variable = Strlen ("String");

Example:-

```
# Include <Stdio.h>
# Include <Conio.h>
# Include <String.h>
Void main()
{
    Char * str;
    int l;
```

Scanned with CamScanner

Scanned with CamScanner

```
clrscr();
printf("enter a string");
gets(str);
l = strlen(str);
printf("length of string = %d", l);
getch()
}
```

2) strlwr(str) :- This function converts all the uppercase character of a string to the lower case.

Syntax:-

```
strlwr("string");
```

Example:-

```
#include <string.h>
#include <stdio.h>
#include <conio.h>
void main()
{
    char *str;
    clrscr();
    printf("enter a string");
    gets(str);
    strlwr(str);
    printf("converted string=
            %s", str);
    getch();
}
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

3) Strupper (Str) :- This function converts all the lower case character of String "Str" into upper case.

Syntax :-

Strupper ("String");

Example :-

```
# Include <Stdio.h>
# Include <Conio.h>
# Include <String.h>
Void main ()
{
    char * Str;
    Clrscr ();
    Printf (" enter a String");
    gets (Str);
    Strupper (Str);
    Printf (" Converted String =
            %s", Str);
    getch ();
}
```

4) Strrev (Str) :- This function is used to return reverse of given String. Strrev function use String.h Headerfile.

Syntax :- Strrev ("String");

Examples :-

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char *str;
    printf("enter a string");
    gets(str);
    strrev(str);
    printf("New string = %s", str);
    getch();
}
```

5) Strcpy(str1, str2):- This function copies str2 to str1.
Strcpy function use string.h heyderfile.

Syntax:-

```
strcpy("string 2, string 1);
```

Example:-

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char *str1, *str2;
    clrscr();
    printf("enter a string");
    gets(str2);
    strcpy(str2, str1);
    printf("coppied string = %s", str2);
    getch();
}
```

Scanned with CamScanner

Scanned with CamScanner

(12) Distinguish between while loop and do-- while loop in c with example.

while	do while

1) Syntax :-
 Initilization
 while (Condition)
 {
 Statement;
 Statement;
 Increment/ Decrement;
 }
	1) Syntax :-
	 Initilization
	 do
	 {
	 Statement;
	 Statement;
	 Incr/ decr;
	 }
	 while (Condition);

2) while loop is most simplest loop structure in c - language.	2) The do -- while loop is an exit control loop structure in c language.
3) In while loop the controlling condition appears at the start of the loop.	3) In do while loop the controlling condition appears at the end of the loop.
4) In while loop semicolon is not used.	4) In do-while loop semicolon is used at the end of the loop.
5) In while loop if the condition is false the statement block will not be executed.	5) In do- while loop if the condition is ~~false~~ false the statement block will be executed at least once.

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

The example of while loop and do-while loop to print 1st 10 natural number

while :-

```
# include <Stdio.h>
# include <conio.h>
Void main()
{
    int n;
    n=1;
    while (n<=10)
    {
        Printf("%d", n);
        n++;
    }
    getch();
}
```

do while :-

```
# include <Stdio.h>
# include <conio.h>
Void main()
{
    int n;
    n=1;
    do
    {
        printf("%d", n);
        n++;
    } while (n<=10)
    getch();
}
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

(13) What is function ? Describe the normal and recursive function with example.

→ function is a block of statement allow to perform specific task using function. we can distribute large program into various small modules.

function also known as method, processure or formula or sub-routine.

function reduce line of code and implement recursivility feature. so using function, we can call any block by single calling statement. function devided into two categories.

i) call by value
ii) call by reference

where C language use system define function also allow to create user define function.

we cannot use the variable in the function that is declare in another function.

Normal function :-

Scanned with CamScanner

Scanned with CamScanner

Example of Normal function to find the sum of two numbers.

```
#include <stdio.h>
#include <conio.h>
int sum (int, int);
void main()
{
int x, y, z;
clrscr();
printf(" enter two number");
scanf ("%d %d", &x, &y);
z = sum (x, y);
printf (" sum = %d ", z);
getch();
}
int sum (int a, int b)
{
int i;
i = a+b;
return i;
}
```

* **Recursion :-** Recursion is a programming technique that allows the programmer to express operation in terms of themself.
any function which called itself called recursive function.
Recursion cannot be applied to all problems.

Example of recursive function to find the factorial of given no.

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

```
#include <stdio.h>
#include <conio.h>
int factorial(int);
void main()
{
int n, f;
clrscr();
printf("enter value of n");
scanf("%d", &n);
f = factorial(n);
printf("factorial = %d", f);
getch();
}
int factorial (int n)
{
if (n==1)
return 1;
else
return (n * factorial(n-1));
}
```

(14) What is actual and formal Parameter?

→ * Actual Parameter :- The Parameters that are passed in a function call are called actual arguments.
This arguments are defined in the calling function.

* Formal parameter :- The formal parameter are parameter in a function declaration.
The scope of formal arguments is local to the function defination in which they are used.

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

formal arguments are copy of actual arguments. Change in formal arguments would not be reflected in the actual arguments.

```
# Include < Stdio.h>
# Include < Conio.h>
int Sum ( int. int);
void main ( )
{
int x, y. z;
Clrscr();
Printf (" enter two numbers");
Scanf (" %d %d ", &x. &y);
z = Sum (x, y);
Printf (" Sum = %d ", z);
getch();
int Sum (int a. int b)
{
int i;
i = a+b;
return i;
}
```

actual parameters → z = Sum (x, y);

formal parameters → int Sum (int a. int b)

(15) What is Structure? Write a C program to show the concept of Structure.

→ A structure is a collection of dissimilar data type. Structure use struct keyword to define itself. Structure element can be access through a structure variable using a dot operator. Using structure we can create int, float, char, variable in a container. Structure must be terminated with semicolon (;)

Scanned with CamScanner

Scanned with CamScanner

DATE :

Syntax :-

```
Struct  Structure name
{
    data type 1
    data type 2
    data type 3
};
```

example of Structure to specify data in Student like as roll no, Name. department, Course or year of Joining and also display the given data.

```
# Include <Stdio.h>
# Include <Conio.h>

Struct std
{
    int roll, yoj;
    Char name[10], course [10],
                dept[10];
};
void main()
{
    Struct  Std, S;
    Printf ("enter roll");
    Scanf ("%d", & S.roll);
    Printf ("enter name");
    gets (S.name);
    Printf ("enter department");
    gets (S.dept);
```

Scanned with CamScanner

Scanned with CamScanner

```
Printf ("enter course");
    gets (s.course);
Printf ("enter year of joining");
    scanf ("%d", s.yoj);
    printf ("after input data are: \n");
        Printf ("Roll = %d", s.roll);
                Puts (s.name);
                puts (s.course);
                Puts (s.dept);
                Printf ("Yoj = %d", s.yoj);
                    getch();
                }
```

(16) What is recursion? explain with example.

→ Recursion is a Programing technique that allows the Programmer to express operations in terms of themself.

Any function which called itself called recursive function.

Recursion cannot be applied to all Problems.

In recursion the calling function and the called function are same.

Recursion is a powerful technique of writing a complicated Program in an easy way

using recursion we can avoid looping.

example of recursion function to find the factorial of give no.

```
# Include < Stdio.h>
# Include < Conio.h>
```

Scanned with CamScanner

Scanned with CamScanner

```
int factorial (int);
void main()
{
    int n, f;
    clrscr();
    Printf(" Enter value of n");
    Scanf(" %d", &n);
    f = factorial(n);
    Printf("factorial = %d", f);
    getch();
}
int factorial (int n);
{
    if (n==1)
    return 1;
    else
    return (n * factorial(n-1));
}
```

(17) Write short notes on any two of the following.

a) Control Structure in C

In Programming "Control Structures" are the Structures used to control the flow of a Program. Control Structures are used in C Program to check the given condition and to execute certain statement accordingly.

C has three types of control statements.

1) Decision making and branching statements

Scanned with CamScanner

Scanned with CamScanner

a) if
b) if -- else
c) nested if -- else
d) else --- if ladder
e) Conditional Statement
f) switch -- case

2) Decision making and looping Statements.
a) while
b) do - while
c) for

3) Control Statements without decision making or Jump Statements.
a) break;
b) continue;
c) goto;
d) return;

4) Array :- Array is a linear data structure use for store various pre define values sequently. using array data structure we can avoid multiply variable declaration for common types of element.

Array Create Multiply Memory block but represent by using single name.

Array divided into two Categories
1) 1-D array
2) 2-D array.

It is simply grouping of like data type data such as list of numbers, list of names. etc

Scanned with CamScanner

PAGE NO.
DATE :

It allocates sequential memory location. individual values are called as elements.

Some example where array can be used to store the values are -

1) List of temperatures recorded every hour in a day, or a month, or a year.
2) List of employes in an organization.
3) List of products and their cost sold by a store.
4) test scores of a class of students.
5) List of customers and their telephone numbers.

c) Storing :-

Strings are actually 1-D array of character terminated by null character that is /0. String use for store any word or sentence. where String variable stored maximum (size-1) character.

String variable automatically stored null character. where null represent end of string.

String use double code symbol and store character in ASCI formate.

We can use gets to input the string and puts to display the string.

Syntax:- Char *< String name > [size];

d) Jump Statements in C :-

Jump statement is used within to transfer control from one point to another point in the program due to some specified code while execution of program.

Scanned with CamScanner

Scanned with CamScanner

There are 4 types of jump statement are in c. These are as follows.

a) break — आगे निकलना है.

b) Continue:- It is used within loop control statement to skip over the execution part of the loop on certain condition. after that it transfer the control to the begning of the loop.

gen. form :- continue;

c) goto:- It is a type of jump statement which is used to branch the control from one point to another point without any condition or unconditionally. The label is placed immediatly before the goto statement where the control is to be transfered.

General form. goto LABEL;

eg:- goto x;

d) return:- It is used to transfer the control from the called function to the calling function. It can be return the value or may be used to just transfer the control. It is also used to return a value from called function to the calling function.

gen form:- return;

or → return value;

ex.- return p;

e) Parameter Passing technique :-

(A) The C Programming language, there are two method to pass parameter from calling function to called function. There are as:-

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

a) Call by Value :- It is a method of Copies the Value of an actual Parameter into formal Parameter of that function. If any change are made to the formal Parameter do not effect the actual Parameter. declare variable for both Parameter is to be Same or different. Both Parameter is Create in different memory location.

b) Call by reference :- It is a method of to Copies the address of an actual Parameter into the formal Parameter. The addresed is used to access the function Call. If any change made to the formal Parameter effect the actual Parameter.

Both actual and formal Parameter Created in the Same memory location.

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

g) break:- Break Statement is used within loop control structure to break the repetition of the loop. It is also used at the end of each case block of switch case statement to transfer the control outside the switch case statement.

The general format of break statement is as follows:-

```
break;
```

When loop control structures are nested the break statement stops the execution of nearest loop or the loop directly containing the break statement.

h) for loop :- The for loop is an entry controlled loop which has more concise loop control structure. The general form of for loop is as follows:-

Syntax:-

```
for ( initilization; conditional check; incr/decr)
{
    body of loop;
}
```

There are three parts in for statement seperated by two semi colons(;). First part is called initilization part where the loop control variables are initialized. The second part is test condition where the value of loop control variable is tested evaluating the test condition. The loop continues if the test condition is found to be true otherwise the loop is terminated and the control jumps out of the loop and

Scanned with CamScanner

Scanned with CamScanner

executes the statement the next statement following the loop.
When the body of loop is executed, the control is transferred back to the third part of for statement which is known as incr/decr part.

(18) Write a program in C language to calculate area of triangle using function with error handling

```
# Include < math.h>
# Include < Stdio.h>
# Include < Conio.h>
float input ();
float area ( float, float, float);
Void Show ( float);
Void main()
{
    int ar, a, b, c;
    Clrscr ();
    a= input ();
    b= input ();
    c= input ();
    ar = area ( a, b, c);
    Show (ar);
    getch();
}
float input ( )
{
    float x;
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

```
Printf ( "enter sides");
Scanf ( " %f", &x);
    return (x);
}
    float ar( float a, float b, float c)
{
    float s, ar;
    s = (a+b+c) / 2;
    ar = sqrt (s * (s-a) * (s-b) *(s-c) );
    return( ar);
}
    void show ( float ar)
{
    Printf ( "\n area = %f", ar);
}
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

(19) Write a Program in C language to Check a given year for leap year.

```
# Include < Stdio.h>
# Include < Conio.h>
Void main ()
{
    int y;
    Printf( " enter year:");
    Scanf ( " %d", &y);
    If ( y % 4 == 0 && y % 100 != 0 ||
                        y % 400 == 0)
    Printf ( " leap year");
    else
        Printf( " Not a leap year");
        getch();
}
```

Output:- enter year: 2016
leap year

(20) Write a Program in C language to reverse a given number using for loop?

```
# Include < Stdio.h>
# Include < Conio.h>
Void main ()
{
    long int rev = 0, n, r;
    clrscr ();
    Printf( " enter any number");
    Scanf ( " %ld ", &n);
```

Scanned with CamScanner

Scanned with CamScanner

```
for ( ; n>0 ; )
{
    q = n%10;
    n = n/10;
    rev = rev*10+ q;
}
    Printf(" reverse of number = %d",
                                rev);
    getch();
}
```

(21) Write a Program in C language to Interchange the value of two variables using pointer.

```
#Include <Stdio.h>
#Include <Conio.h>
Void Swap (int*, int*);
Void main()
{
    int x, y;
    Clrscr();
    Printf("enter the value of
            x and y");
    Scanf("%d %d", &x, &y);
    Printf(" before Swaping x = %d
            and y = %d", x, y);
    Swap(&x, &y);
    Printf("\n After Swaping x = %d
            and y = %d", x, y);
    getch();
}
```

Scanned with CamScanner

Scanned with CamScanner

```
Void Swap (int *m, int *n)
{
    int t;
    t = *m;
    *m = *n;
    *n = t;
}
```

(22) Write a C programe to enter three number and print largest of the given three number

```
# Include <Stdio.h>
# include <conio.h>
Void main()
{
    int x, y, z;
    Clrscr();
    Printf("enter three number:");
    Scanf("%d %d %d", &x, &y, &z);
    if (x>y)
        if (x>z)
        printf("%d is largest", x);
    else
        printf("%d is largest", z);
    else
        if (y>z)
        Printf("%d is largest", y);
    else
        Printf("%d is largest", z);
    getch();
}
```

Output :- enter three number:
2 . 4 . 3
4 is largest

Scanned with CamScanner

PAGE NO.
DATE :

(23) Write a Program in C to Print a table of given number

While loop.

```
# Include <Stdio.h>
# Include <Conio.h>
Void main()
{
int n, i;
Clrscr();
Printf ("enter number:");
Scanf ("%d",&n);
i=1;
while (i<=10)
{
Printf ("\n %d*%d= %d", n,
i, n*i);
i++;
getch();
}
```

for loop :-

```
# Include <Stdio.h>
# Include <Conio.h>
Void main()
{
int n, i;
Clrscr();
printf ("enter number:");
Scanf ("%d", &n);
for (i=1; i<=10; i++)
{
Printf ("\n %d* %d= %d", n, i, n*i)
}
getch();
}
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

(24) Write a Program to Print all Prime number from 2 to n given number

```
#include <stdio.h>
#include <conio.h>
void main()
{
int i, n, p, l;
clrscr();
Printf(" enter the last limit:");
scanf("%d", &l);
Printf(" Prime numbers between 2 to
%d:-(n", l);
for(i=2; i<=l ; n++);
{
p=0;
for(n=2; n*n<=i ; n++)
{
if(i%n==0)
{
p++;
break;
}
}
if(p==0)
Printf("%t%d", i);
}
getch();
}
```

Output:- enter the last limit: 7
Prime numbers between 2 to 7 :-
2, 3, 5

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

(25) Write a C program to print factorial values of given integer using function.

```
# Include < Stdio.h>
# Include < Conio.h>
int factorial( int);
Void main()
{
    int n, f;
    clrscr();
    Printf ("enter Value of n");
    Scanf ("%d" . &n)
      f = factorial (n);
    Printf ("factorial = %d", f);
    getch();
}
int factorial (int n);
{
    if (n == 1)
    return 1;
else
    return (n * factorial (n-1);
}
```

Scanned with CamScanner

Scanned with CamScanner

(26) write a program to multiply two mxn matrix

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[10][10],b[10][10],c[10][10],i,j,k,r1,r2,c1,c2;
        clrscr();
        printf("enter order of 1st matrix\n");
        scanf("%d%d",&r1,&c1);
        printf("enter order of 2nd matrix\n");
        scanf("%d%d",&r2,&c2);
        if(r1==c2)
         {
          printf("enter %dx%d element of 1st matrix\n",r1,c1);
          for(i=0;i<r1;i++)
            for(j=0;j<c1;j++)
             scanf("%d",&a[i][j]);
          printf("enter %dx%d element of 2nd matrix\n",r2,c2);
           for(i=0;i<r1;i++)
            for(j=0;j<c1;j++)
             scanf("%d",&b[i][j]);
          printf("\n element of 1st matrix\n");
           for(i=0;i<r1;i++)
           {
             for(j=0;j<c1;j++)
              printf("%d\t",a[i][j]);
            printf("\n");
           }
         printf(" \nelement of 2nd matrix\n");
          for(i=0;i<r2;i++)
          {
           for(j=0;j<c2;j++)
            printf("%d\t",b[i][j]);
           printf("\n");
          }
```

Scanned with CamScanner

```
for(i=0;i<r2;i++)
   for(j=0;j<c2;j++)
    c[i][j]=0;
for(i=0;i<r2;i++)
 for(j=0;j<c2;j++)
      for(k=0;k<c1;k++)
       c[i][j]=c[i][j]+a[i][k]*b[k][j];
 printf("\n after multiply the resultant matrix is\n");
for(i=0;i<r1;i++)
 {
  for(j=0;j<c1;j++)
    printf("%d\t",c[i][j]);
   printf("\n");
 }
}
 else
   printf("multiplication not possible");
getch();
}
```

2021/2/1 19:08

Scanned with CamScanner

PAGE NO.
DATE :

(27) Write a C Program to enter a String. Print How many vowels, Consonant and words are Present in the given Strings.

```
# Include <Stdio.h>
# Include <Conio.h>
# Include <String.h>
Void main()
{
    char *Str;
    int i, v=0, c=0, w=0;
    Clrscr();
    Printf("Enter String:");
    gets(Str);
    Strlwr(Str);
    For(i=0; Str[i]!='\0'; i++)
     if(Str[i]=='a' || Str[i]==
      'e' || Str[i]=='i' || Str[i]==
      'o' || Str[i]=='u')
      v++;
     else if (Str[i]==' '&&
                Str[i+1]!=' ')
      w++;
     else
      c++;
    Printf("given String:");
    Puts(Str);
    Printf("total Vowel = %d", v);
    Printf("\n total Consonant = %d", c);
    Printf("\n total word = %d", w+1);
    getch();
}
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

Output :- Enter String: Sundaram kumar
given String: Sundaram kumar
total Vowel = 4
total Consonant = 8
total word = 2

(28) Write a "C" Programe to check that given number is armstrong or not.

```
# Include < Stdio.h>
# include < Conio.h>
Void main()
{
    int n, sum=0, org;
    Clrscr();
    Printf (" enter 3 digit number:");
    Scanf ( " %d", &n);
    Org = =n;
    while (n>0)
    {
    a = n % 10;
    n = n/10;
    sum = sum + a*a*a;
    }
    if (org = sum).
    printf ("no is armstrong");
    else
    printf ("No is not armstrong");
    getch();
}
```

Scanned with CamScanner

Scanned with CamScanner

PAGE NO.
DATE :

(29) Check a given number is palindrome or not.

```
# Include <stdio.h>
# Include <conio.h>
Void main ()
{
    long int n, n1, rev=0, d;
    clrscr();
    Printf ("enter number");
    Scanf ("%d", &n);
    n1 = n ;
    while (n>0)
    {
      d = n % 10;
      rev = rev*10+d;
      n = n/10;
    }
    if (n1 == rev)
    Printf ("%ld is palindrome", n1);
    else
    Printf ("%ld is not Pallindrome",
                                n1);
    getch();
}
```

Output:- enter number 1579751
1579751 is Pallindrome.

Scanned with CamScanner
Scanned with CamScanner

PAGE NO.
DATE :

(30) Write a Program in C to check a given number is prime or not.

```
# Include < Stdio.h>
# Include < Conio.h>
Void main ()
{
    int n, i, f=0;
    Clrscr ();
    Printf( "enter number: ");
    Scanf ( " %d ",&n);
        i=1;
    while ( i<=n)
    {
    if (n%i ==0)
        f= f+1;
        i= i+1;
    }
    if (f==2)
    Printf ( "Prime number");
    else
    Printf ( " not Prime number");
        getch ();
}
```

Output : Enter number: 3
Prime number.

Scanned with CamScanner

Scanned with CamScanner