PCI

STUDY NOTES Q&A [FOR FINAL MSBTE]

Q.1 Define array. List its type.

- Array is defined as collection of element which is ordered, Homogeneous (Same Type of Data) and fixed size
- types of array are
  - One dimensional array
  - Multi-dimensional array

Q.2 State the syntax to declare a pointer variable with an example

- It is variable which stores the address of another variable
- DECLARATION SYNTAX:- data_type*pointer_name;
- Pointer is declared using special character '*'
- Example:- int *p;

Q.3 Write a program to declare structure students having roll no, name & marks Accept & display data for 3 students.

```
#include <stdio.h>
struct student
{
char name[50];
int roll;
float marks;
}
s[100];
int main()
{
int i,n;
struct student s[100];
printf("Enter total of students:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
```

```
printf("\n Enter information of student %d:\n",i+1);
printf("Enter name: ");
scanf("%s", s[i].name);
printf("Enter roll number: ");
scanf("%d", &s[i].roll);
printf("Enter marks: ");
scanf("%f", &s[i].marks);
}
printf("Displaying Information:\n");
for(i=0;i<n;i++)
{
printf("\n %d no. student info\n",i+1);
printf("\t Name:%s\n ",s[i].name);
printf("\t Roll number: %d\n",s[i].roll);
printf("\t Marks: %.1f\n\n",s[i].marks);
}
return 0;
}
```

Q.4 Explain pointer arithmetic with examples.

- Increment/Decrement of a Pointer
  - If an integer pointer that stores address 1000 is incremented, then it will increment by 2(size of an int) and the new address it points to 1002. While if a float type pointer is incremented then it will increment by 4(size of a float) and the new address will be 1004.
  - If an integer pointer that stores address 1000 is decremented, then it will decrement by 2(size of an int) and the new address it points to 998. While if a float type pointer is decremented then it will decrement by 4(size of a float) and the new address will be 996.
- Addition of integer to a pointer
  - For 32-bit int variable, it will add 2 * number.
  - For 64-bit int variable, it will add 4 * number.
- Subtraction of integer to a pointer
  - For 32-bit int variable, it will subtract 2 * number.

- For 64-bit int variable, it will subtract 4 * number.

- Subtracting two pointers of the same type
  - Two integer pointers say ptr1(address:1000) and ptr2(address:1016) are subtracted. The difference between addresses is 16 bytes. Since the size of int is 2 bytes, therefore the increment between ptr1 and ptr2 is given by (16/2) = 8.

Q.5 Explain nested if-else with example

- When an if else statement is present inside the body of another "if" or "else" then this is called nested if else.
- SYNTAX

```
if(condition)
{
//Nested if else inside the body of "if"
if(condition2)
{
//Statements inside the body of nested "if"
}
else
{
//Statements inside the body of nested "else"
}
}
else
{
//Statements inside the body of "else"
}
```

- EXAMPLE

```
#include <stdio.h>
int main()
{
int var1, var2;
printf("Input the value of var1:");
scanf("%d", &var1);
```

```
printf("Input the value of var2:");
scanf("%d",&var2);
if (var1 != var2)
{
printf("var1 is not equal to var2\n");
if (var1 > var2)
{
printf("var1 is greater than var2\n");
}
else
{
printf("var2 is greater than var1\n");
}
}
else
{
printf("var1 is equal to var2\n");
}
return 0;
}
```

Q.6 Describe the following terms:

(i) Keyword :- are predefined, reserved words used in programming that have special meanings to the compiler.

- Example :- int age; (here int is a keyword)

(ii) Identifier :- Identifier refers to name given to entities such as variables, functions, structures etc.Identifiers must be unique.

- Example :- int money; (here money is identifier)

(iii) Variable :- A variable is a name of the memory location. It is used to store data. Its value can be changed, and it can be reused many times.

- Example :- int a=10;

(iv) Constant :- values which cannot be modified once they are defined They are fixed values in a program.

- Example :- const int pi = 3.14;

Q.7 Differentiate between call by value and call by reference.

CALL BY VALUE	CALL BY REFERNCE
➢ A copy of the value is passed into the function	➢ An address of value is passed into the function.
➢ Changes made in this function is limited to the function only	➢ Change made inside the function validate outside of the function also.
➢ The values of the actual parameters do not change by changing the formal parameters.	➢ the values of the actual parameters do change by changing the formal parameters
➢ Actual and formal arguments are created at the different memory location	➢ Actual and formal arguments are created at the same memory location

Q.8 Explain conditional operator with example.

- The conditional statements are the decision-making statements which depends upon the output of the expression.
- As conditional operator works on three operands, so it is also known as the ternary operator.

- Example

```
#include <stdio.h>
int main()
{
int age;
printf("Enter your age: ");
scanf("%d", &age);
(age >= 18) ? printf("You can vote") : printf("You cannot vote");
```

```
return 0;
}
```

Q.9 List the categories of functions and explain any one with example.

- Argument & return value
- Argument &no return value
- No argument & return value
- No argument & no return value

- Program: -

```
#include <stdio.h>
void my_function()
{
printf("This is a function that takes no argument, and returns nothing.");
}
main()
{
my_function();
}
```

- Explanation:- This is a function that takes no argument, and returns nothing. Here this function is not taking any input argument, and also the return type is void. So this returns nothing.

Q.10 Write an algorithm to determine the given number is odd or even

- Step 1-Start
- Step 2- Read / input the number.
- Step 3- if n%2==0 then number is even.
- Step 4- else number is odd.
- Step 5- display the output.

- Step 6- Stop

Q.11 Illustrate the use of break and continue statement with example.

1. Break :-

- It helps make an early exit from block of a code
- It is used in program by a keyword break;
- It can be used in all control statements

- Example: -

```
#include <stdio.h>
int main()
{
int i;
for (i = 0; i < 10; i++)
{
if (i == 4)
{
break;
}
printf("%d\n", i);
}
return 0;
}
```

- Output: -

```
0
1
2
3
```

2. Continue:-

- It helps in avoiding remaining statement in the current iteration and continues with the next iteration.

- It can be used in loop only

- Example:-

```
#include <stdio.h>
int main()
{
int i;
for (i = 0; i < 10; i++)
{
if (i == 4)
{
continue;
}
printf("%d\n", i);
}
return 0;
}
```

- Output: -

```
0
1
2
3
4
5
6
7
8
9
```

Q.12 Write a program to add, subtract, multiply and divide two numbers, accepted from user using switch case.

```
#include<stdio.h>
int main()
{
int n,m,t;
```

```
char c;
printf("Enter two numbers and operator :\n");
scanf("%d %d %c", &n, &m, &c);
switch(c)
{
case '+' : printf("Addition is : %d", n+m);
break;
case '-' : printf("Substraction is %d", n-m);
break;
case '*' : printf("Multiplication is %d", n*m);
break;
case '/' : printf("Division is %f", (float)n/m);
break;
default  : printf("Not valid");
}
return 0;
}
```

Q.13 Illustrate initialization of two dimensional array with example.

- Two-dimensional arrays may be initialized by specifying bracketed values for each row.
- To declare a two-dimensional integer array of size [x][y], you would write something as follows −
- SYNTAX: - data_type array_name [x][y];
- Where type can be any valid C data type and arrayname will be a valid C identifier.
- A two-dimensional array can be considered as a table which will have x number of rows and y number of columns.

- Example: -

```
int a[3][4] = {0,1,2,3,4,5,6,7,8,9,10,11};
```

Q.14 Write a program to read two strings and find whether they are equal or not.

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

```
#include <string.h>
int main()
{
char str1[20],str1[20];
printf (“ENTER FIRST STRING”);
gets(str1);
printf(“ENTER SECOND STRING”);
gets(str2);
if(strcmp(str1,str2)==0)
printf(“BOTH ARE EQUAL”);
else
printf (“NOT EQUAL”);
return 0;
}
```

Q.15 Write a program for addition of two 3× 3 matrices.

```
#include <stdio.h>
int main()
{
int mat1[3][3];
int mat2[3][3];
printf("Enter numbers of 1st matrix\n");
for (int i = 0; i < 3; i++)
{
for (int j = 0; j < 3; j++)
{
printf("Enter number at row %d, column %d: ", i+1, j+1);
scanf("%d", &mat1[i][j]);
}
}
printf("Enter numbers of 2nd matrix\n");
for (int i = 0; i < 3; i++)
{
for (int j = 0; j < 3; j++)
{
printf("Enter number at row %d, column %d: ", i+1, j+1);
```

```
scanf("%d", &mat2[i][j]);
}
}
int sum_mat[3][3];
for (int i = 0; i < 3; i++)
{
for (int j = 0; j < 3; j++)
{
sum_mat[i][j] = mat1[i][j] + mat2[i][j];
}
}
printf("Sum of the matrices is: \n");
for (int i = 0; i < 3; i++)
{
for (int j = 0; j < 3; j++)
{
printf("%d ", sum_mat[i][j]);
}
printf("\n");
}
}
```

Q.16 Write a program to compute the sum of all elements stored in an array using pointers.

```
#include <stdio.h>
void main()
{
int arr1[10];
int i,n, sum = 0;
int *pt;
printf("\n\n Pointer : Sum of all elements in an array :\n");
printf("-------------------------------------------\n");
printf(" Input the number of elements to store in the array: ");
scanf("%d",&n);
printf(" Input %d number of elements in the array : \n",n);
for(i=0;i<n;i++)
```

```
{
printf(" element - %d : ",i+1);
scanf("%d",&arr1[i]);
}
pt = arr1;
for (i = 0; i < n; i++)
{
sum = sum + *pt;
pt++;
}
printf(" The sum of array is : %d\n\n", sum);
}
```

Q.17 Write a program to sort elements of an array in ascending order.

```
#include <stdio.h>
void main ()
{
int num[20];
int i, j, a, n;
printf("enter number of elements in an array\n");
scanf("%d", &n);
printf("Enter the elements\n");
for (i = 0; i < n; ++i)
scanf("%d", &num[i]);
for (i = 0; i < n; ++i)
{
for (j = i + 1; j < n; ++j)
{
if (num[i] > num[j])
{
a = num[i];
num[i] = num[j];
num[j] = a;
}
}
```

```
}
printf("The numbers in ascending order is:\n");
for (i = 0; i < n; ++i)
{
printf("%d\n", num[i]);
}
}
```

Q.18 Write a function to print Fibonacci series starting from 0, 1.

```
#include <stdio.h>
int main()
{
int i, n;
int t1 = 0, t2 = 1;
int nextTerm = t1 + t2;
printf("Enter the number of terms: ");
scanf("%d", &n);
printf("Fibonacci Series: %d, %d, ", t1, t2);
for (i = 3; i <= n; ++i)
{
printf("%d, ", nextTerm);
t1 = t2;
t2 = nextTerm;
nextTerm = t1 + t2;
}
return 0;
}
```

Q.19 Calculate factorial of a number using recursion.

```
#include <stdio.h>
int fact (int);
int main()
{
int n,f;
printf("Enter the number whose factorial you want to calculate?");
```

```
scanf("%d",&n);
f = fact(n);
printf("factorial = %d",f);
}
int fact(int n)
{
if (n==0)
{
return 0;
}
else if (n == 1)
{
return 1;
}
else
{
return n*fact(n-1);
}
}
```

Q.20 Give any four advantages of pointer.

- Pointers are helpful in allocation and de-allocation of memory during the execution of the program. Thus, pointers are the instruments of dynamic memory management.
- Pointers enhance the execution speed of a program.
- Pointers make the programs simple and reduce their length.
- Passing on arrays by pointers saves lot of memory because we are passing on only the address of array instead of all the elements of an array

Q.21 Define type casting. Give any one example.

- Converting one datatype into another is known as type casting or, type-conversion.
- If you want to store a 'long' value into a simple integer, then you can type cast 'long' to 'int'.

- Example

A. Without Type Casting:

```
int f= 9/4;
printf("f : %d\n", f );
//Output: 2
```

B.With Type Casting:

```
float f=(float) 9/4;
printf("f : %f\n", f );
//Output: 2.250000
```

Q.22 State any four decision making statements.

- if statement
- if else statement
- switch statement
- Jump statement

a) break
b) continue
c) go to
d) return

Q.23 State any four math functions with its use.

- To find the square root of a number, use the sqrt() function:
- Ex:- printf("%f", sqrt(16)); // output 4.0000

- The ceil() function rounds a number upwards to its nearest integer
- Ex:- printf("%f", ceil(1.4)); // output 2.0000

- floor() method rounds a number downwards to its nearest integer, and returns the result
- Ex:- printf("%f", floor(1.4)); // output 1.0000

- The pow() function returns the value of x to the power of y (xy):
- Ex:- printf("%f", pow(4, 3)); // output 64.0000

Q.24 State use of while loop with syntax.

- While loop in C programming repeatedly executes a target statement as long as a given condition is true.
- Syntax: -

```
while(condition)
{
statement(s);
}
```

- Here, statement(s) may be a single statement or a block of statements.
- The condition may be any expression, and true is any non-zero value.
- The loop iterates while the condition is true.
- When the condition becomes false, the program control passes to the line immediately following the loop.

Q.25 Develop a simple 'C' program for addition and multiplication of two integer numbers.

```
#include <stdio.h>
int main()
{
int num1, num2, sum,m;
printf("Enter two integers: ");
scanf("%d %d", &num1, &num2);
sum = num1 + num2;
printf("Addition is : ", sum);
m= num1*num2;
printf("Multiplication is:"m);
return 0;
}
```

Q.26 Explain how to pass pointer to function with example.

```
#include <stdio.h>
int main()
{
int* p, i = 10;
p = &i;
addOne(p);
printf("%d", *p); // 11
return 0;
}
void addOne(int* ptr)
{
(*ptr)++; // adding 1 to *ptr
}
```

- Here, the value stored at p, *p, is 10 initially.
- We then passed the pointer p to the addOne() function.
- The ptr pointer gets this address in the addOne() function.
- Inside the function, we increased the value stored at ptr by 1 using (*ptr)++;. Since ptr and p pointers both have the same address, *p inside main() is 11

Q.27 Explain following functions:

- getchar( ) The getchar function is part of the <stdio. h> header file in C.
- It is used when single character input is required from the user.
- The function reads the input as an unsigned char

```
#include<stdio.h>
int main ()
{
char c;
printf("Enter character: ");
c = getchar();
printf("Character entered: ");
putchar(c);
```

```
return(0);
}
```

2. putchar( ) method in C is used to write a character, of unsigned char type, to stdout.

- This character is passed as the parameter to this method.
- This method accepts a mandatory parameter char which is the character to be written to stdout.

- Example:-

```
#include<stdio.h>
int main()
{
char ch='g';
putchar(ch);
return (0);
}
```

3. getch( ) method pauses the Output Console until a key is pressed.

- It does not use any buffer to store the input character.
- The entered character is immediately returned without waiting for the enter key
- Example

```
#include <stdio.h>
#include <conio.h>
int main()
{
printf(" Passed value by the User is: %c", getch());
return 0;
}
```

4. putch( )

- The putch() function is used for printing character to a screen at current cursor location.

- It is unformatted character output functions. It is defined in header file conio.
- Example

```
void main()
{
char ch='a';
putch(ch)
}
```

Q.28 Develop a program to accept an integer number and print whether it is palindrome or not

```
#include<stdio.h>
int main()
{
int n,r,sum=0,temp;
printf("enter the number=");
scanf("%d",&n);
temp=n;
while(n>0)
{
r=n%10;
sum=(sum*10)+r;
n=n/10;
}
if(temp==sum)
printf("palindrome number ");
else
printf("not palindrome");
return 0;
}
```

Q.29 State the use of printf( ) &scanf( ) with suitable example

1. The printf() function is used for output.
   - It prints the given statement to the console
   - SYNTAX :- printf("format string",argument_list);

- Example

```
#include<stdio.h>
int  main()
{
int num = 450;
printf("Number is %d \n", num);
return 0;
}
```

2. The scanf() function is used for input.
   - It reads the input data from the console.
   - SYNTAX:- scanf("format string",argument_list);
   - Example:-

```
#include<stdio.h>
int main()
{
int x;
printf("ENTER THE NUMBER");
scanf("%d",&x);
printf("The number is=%d",x);
return 0;
}
```

Q.30 Explain any four library functions under conio.h header file.

- clrscr() -This function is used to clear the output screen.
- getch() -It reads character from keyboard
- getche() -It reads character from keyboard and to o/p screen
- textcolor() -This function is used to change the text color

Q.31 Explain how formatted input can be obtained, give suitable example.

- Formatted I/O functions are used to take various inputs from the user and display multiple outputs to the user.
- These types of I/O functions can help to display the output to the user in different formats using the format specifiers.
- These I/O supports all data types like int, float, char, and many more.

- These functions are called formatted I/O functions because we can use format specifiers in these functions and hence, we can format these functions according to our needs.
- EXAMPLE:

FORMAT SPECIFIER	NAME	USE
%d	int	Used for I/O signed integer value
%c	char	Used for I/O character value
%f	float	Used for I/O decimal floating-point value
%s	string	Used for I/O string/group of characters
%ld	long int	Used for I/O long signed integer value

Q.32 Write a program to sweep the values of variables a = 10, b = 5 using function.

```
#include<stdio.h>
void swap(int *,int *);
int main()
{
int n1,n2;
printf("\n\n Function : swap two numbers using function :\n");
printf("-----------------------------------------------\n");
printf("Input 1st number : ");
scanf("%d",&n1);
printf("Input 2nd number : ");
scanf("%d",&n2);
printf("Before swapping: n1 = %d, n2 = %d ",n1,n2);
swap(&n1,&n2);
printf("\nAfter swapping: n1 = %d, n2 = %d \n\n",n1,n2);
return 0;
}
void swap(int *p,int *q)
```

```
{
int tmp;
tmp = *p;
*p=*q;
*q=tmp;
}
```

Q.33 Develop a program using structure to print data of three students having data members name, class, percentage.

```
#include <stdio.h>
struct student
{
char name[50];
int class;
float percentage;
}
s[100];
int main()
{
int i,n;
struct student s[100];
printf("Enter total of students:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("\n Enter information of student %d:\n",i+1);
printf("Enter name: ");
scanf("%s", s[i].name);
printf("Enter class: ");
scanf("%d", &s[i].class);
printf("Enter percentage: ");
scanf("%f", &s[i]. percentage);
}
printf("Displaying Information:\n");
for(i=0;i<n;i++)
```

```
{
printf("\n %d no. student info\n",i+1);
printf("Name:%s\n ",s[i].name);
printf("Class: %d\n",s[i].class);
printf("percentage: %.1f\n\n",s[i]. percentage);
}
return 0;
}
```

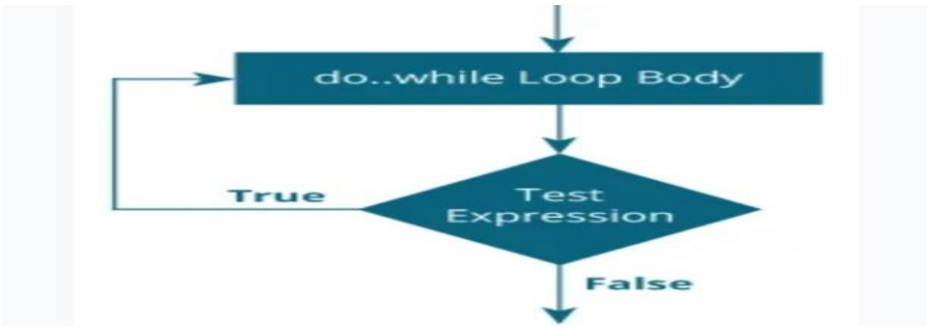
Q.34 Design a program to print a message 10 times.

```
#include<stdio.h>
int main()
{
int i;
for(i=1;i<=10;printf("  Hello World ! \n",i++));
return 0;
}
```

Q.34 Implement a program to demonstrate logical AND operator.

```
#include <stdio.h>
#include <conio.h>
int main ()
{
int n = 20;
printf (" %d \n", (n == 20 && n >= 8));
printf (" %d \n", (n >= 1 && n >= 20));
printf (" %d \n", (n == 10 && n >= 0));
printf (" %d \n", (n >= 20 && n <= 40));
return 0;
}
```

Q.35 Draw a flowchart of Do-while loop and write a program to add numbers until user enters zero.




```
#include <stdio.h>
int main()
{
double number, sum = 0;
do
{
printf("Enter a number: ");
scanf("%lf", &number);
sum += number;
}
while(number != 0.0);
printf("Sum = %.2lf",sum);
return 0;
}
```

Q.36 Give a method to create, declare and initialize structure also develop a program to demonstrate nested structure.

- ‘struct’ keyword is used to create a structure.
- Here's how we create structure variables:

```
struct Person
{
  // code
};
int main()
{
struct Person person1, person2, p[20];
```

```
return 0;
}
```

- To initialize a structure's data member, create a structure variable.
- This variable can access all the members of the structure and modify their values.
- Consider the following example which initializes a structure's members using the structure variables.

```
struct rectangle
{
// structure definition
int length;
int breadth;
};
int main()
{
struct rectangle my_rect; // structure variables;
my_rect.length = 10;
my_rect.breadth = 6;
}
```

Example of nested structure: -

```
#include<stdio.h>
struct address
{
char city[20];
int pin;
char phone[14];
};
struct employee
{
char name[20];
struct address add;
};
```

```
void main ()
{
struct employee emp;
printf("Enter employee information?\n");
scanf("%s %s %d %s",emp.name,emp.add.city, &emp.add.pin,
emp.add.phone);
printf("Printing the employee information....\n");
printf("name: %s\nCity: %s\nPincode: %d\nPhone:
%s",emp.name,emp.add.city,emp.add.pin,emp.add.phone);
}
```

Q.37 Implement a program to demonstrate concept of pointers to function

```
#include<stdio.h>
void swap(int *,int *);
int main()
{
int n1,n2;
printf("\n\n Function : swap two numbers using function :\n");
printf("-----------------------------------------------\n");
printf("Input 1st number : ");
scanf("%d",&n1);
printf("Input 2nd number : ");
scanf("%d",&n2);
printf("Before swapping: n1 = %d, n2 = %d ",n1,n2);
swap(&n1,&n2);
printf("\nAfter swapping: n1 = %d, n2 = %d \n\n",n1,n2);
return 0;
}
void swap(int *p,int *q)
{
int tmp;
tmp = *p;
*p=*q;
*q=tmp;
}
```

Q.38 Develop a program to swap two numbers using pointer and add swaped numbers also print their addition.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int x,y,*ptr_x,*ptr_y,temp;
clrscr();
printf("Enter the value of x and y\n");
scanf("%d%d", &x, &y);
printf("Before Swapping\nx = %d\ny = %d\n", x, y);
ptr_x = &x;
ptr_y = &y;
temp = *ptr_y;
*ptr_y = *ptr_x;
*ptr_x = temp;
printf("After Swapping\nx = %d\ny = %d\n", x, y);
sum=*ptr x+*ptr_y;
printf("\nSum = %d",sum);
getch();
}
```

Q.39 Design a programme in C to read the n numbers of values in an array and display it in reverse order.

```
#include <stdio.h>
void main()
{
int i,n,a[100];
printf("\n\nRead n number of values in an array and display it in
reverse order:\n");
printf("-----------------------------------------------------------------\n");
printf("Input the number of elements to store in the array :");
```

```
scanf("%d",&n);
printf("Input %d number of elements in the array :\n",n);
for(i=0;i<n;i++)
{
printf("element - %d : ",i);
scanf("%d",&a[i]);
}
printf("\nThe values store into the array are : \n");
for(i=0;i<n;i++)
{
printf("% 5d",a[i]);
}
printf("\n\nThe values store into the array in reverse are :\n");
for(i=n-1;i>=0;i--)
{
printf("% 5d",a[i]);
}
printf("\n\n");
}
```

Q.40 Develop a program to find diameter, circumference and area of circle using function.

```
#include<stdio.h>
int main()
{
float diameter(float);
float circumference(float);
float area(float);
float d, c, a, r;
printf("Enter radius\n");
scanf("%f", &r);
d=diameter(r);
c=circumference(r);
a=area(r);
printf("Diameter = %0.2f\n",d);
printf("Circumference = %0.2f\n",c);
```

```
printf("Area = %0.2f", a);
return 0;
}
float diameter(float r)
{
return (2* r);
}
float circumference(float r)
{
return (2* 3.14 *r);
}
float area(float r)
{
return (3.14*r*r);
}
```

Q.41 Define Algorithm.

- Algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output

Q.42 Give the significance of <math.h> and <stdio.h> header files.

- The C <math. h> header file declares a set of functions to perform mathematical operations such as: sqrt() to calculate the square root, log() to find natural logarithm of a number etc.

- The C <stdio. h> is a header file which has the necessary information to include the input/output related functions in our program. Example printf, scanf etc. If we want to use printf or scanf function in our program, we should include the stdio.

Q.43 Give syntax of if-else ladder.

```
if(expression)
{
```

```
//code to be executed if condition is true
}
Else
{
//code to be executed if condition is false
}
```

Q.44 Write syntax and use of pow( ) function of <math.h> header file.

- Syntax: - int pow (int x, int y);
- The pow() function has two integer arguments for raising the power of base value (x) by the exponent (y).

Q.45 Define pointer. Write syntax for pointer declaration.

- It is variable which stores the address of another variable
- DECLARATION SYNTAX:- data_type*pointer_name;
- Pointer is declared using special character '*'

Q.46 Write an algorithm to determine whether a given number is divisible by 5 or not.

Step 1- Start
Step 2- Read / input the number.
Step 3- if n%5==0 then goto step 5.
Step 4- else number is not divisible by 5 goto step 6.
Step 5- display the output number is divisible by 5.
Step 6- Stop

Q.47 Explain do – while loop with example.

- A do...while loop is similar to a while loop, except the fact that it is guaranteed to execute at least one time.
- SYNTAX: -

```
do
{
```

```
statement(s);
}
while(condition);
```

- PROGRAM:

```
#include <stdio.h>
int main ()
{
nt a = 10;
do
{
printf("value of a: %d\n", a);
a = a + 1;
}
while( a < 20 );
return 0;
}
```

Q.48 Explain one dimension and two dimension arrays.

Basis	**One Dimension Array**	**Two Dimension Array**
Definition	Store a single list of the element of a similar data type.	Store a 'list of lists' of the element of a similar data type.
Representation	Represent multiple data items as a list.	Represent multiple data items as a table consisting of rows and columns.
Dimension	One	Two
Size(bytes)	size of(datatype of the variable of the	size of(datatype of the variable of the array)* the

	array) * size of the array	number of rows* the number of columns.
Address calculation.	Address of a[index] is equal to (base Address+ Size of each element of array * index).	Address of a[i[[j] can be calculated in two ways row-major and column-major 1. **Column Major:** Base Address + Size of each element (number of rows(j-lower bound of the column)+(i-lower bound of the rows)) 2. **Row Major:** Base Address + Size of each element (number of columns(i-lower bound of the row)+(j-lower bound of the column))
Example	int arr[5];	int arr[2][5];

Q.49 Write the output of following c program

```
#include<stdio.h>
 int main ( )
 {
 char *ptr;
 charstr[]="MAHARASHTRA STATE BOARD OF
TECHNICAL EDUCATION";
 ptr=str;
 ptr=ptr+11;
 printf("%s",++ptr);
```

```
return 0;
}
```

ERROR

- 'str' undeclared
- 'charstr' undeclared
- expected expression before '%' token

Q.50 Explain increment and decrement operator.

1. INCREMENT

- Increment operators are used to increase the value of the variable by one
- SYNTAX: ++var_name; (or) var_name++;
- Example: Increment operator : ++ i ; i ++ ;

2. DECREMENT

- decrement operators are used to decrease the value of the variable by one
- Syntax: – -var_name; (or) var_name – -;
- Example: – – i ; i – – ;

Q.51 Explain User defined function with example.

- A function is a block of code that performs a specific task. C allows you to define functions according to your need.
- These functions are known as user-defined functions.

PROGRAM:

```
#include<stdio.h>
void printName();
void main ()
{
printf("Hello ");
printName();
}
void printName()
```

```
{
printf("Javatpoint");
}
```

Q.52 Write a program to accept marks of four subjects as input from user. Calculate and display total and percentage marks of student.

```
#include <stdio.h>
#include <string.h>
void main()
{
int rl,phy,che,ca,total;
float per;
char nm[20],div[10];
printf("Input the Roll Number of the student :");
scanf("%d",&rl);
printf("Input the Name of the Student :");
scanf("%s",nm);
printf("Input  the marks of Physics, Chemistry and Computer Application : ");
scanf("%d%d%d",&phy,&che,&ca);
total = phy+che+ca;
per = total/3.0;
if (per>=60)
strcpy(div,"First");
else
if (per<60&&per>=48)
strcpy(div,"Second");
else
if (per<48&&per>=36)
strcpy(div,"Pass");
else
strcpy(div,"Fail");
printf("\nRoll No : %d\nName of Student : %s\n",rl,nm);
printf("Marks in Physics : %d\nMarks in Chemistry : %d\nMarks in Computer Application : %d\n",phy,che,ca);
```

```
printf("Total Marks = %d\nPercentage = %5.2f\nDivision = %s\n",total,per,div);
}
```

Q.53 Write a program to accept the value of year as input from the keyboard & print whether it is a leap year or not.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int year;
printf("Enter a year: ");
scanf("%d", &year);
if(((year%4==0) && ((year%400==0) || (year%100!==0))
{
printf("%d is a leap year", &year);
} else {
printf("%d is not a leap year", &year);
}
getch();
}
```

Q.54 Write a program to accept a string as input from user and determine its length. [Don’t use built in library function strlen()]

```
#include <stdio.h>
int main() {
char s[] = "Programming is fun";
int i;
for (i = 0; s[i] != '\0'; ++i);
printf("Length of the string: %d", i);
return 0;
}
```

Q.55 Write a program to swap two numbers using call by value.

```
#include<stdio.h>
int main()
{
double first, second, temp;
printf("Enter first number: ");
scanf("%lf", &first);
printf("Enter second number: ");
scanf("%lf", &second);
temp = first;
first = second;
second = temp;
printf("\nAfter swapping, first number = %.2lf\n", first);
printf("After swapping, second number = %.2lf", second);
return 0;
}
```

Q.56 Write a program using switch statement to check whether entered character is VOWEL or CONSONANT.

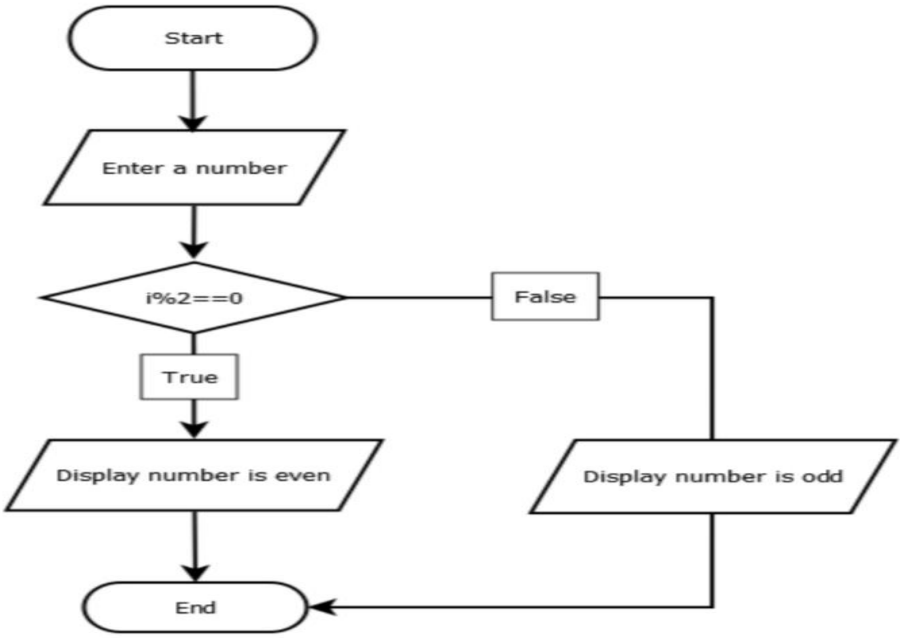
```
#include<stdio.h>
int main()
{
char ch;
printf("Input a Character :  ");
scanf("%c", &ch);
switch(ch)
{
case 'a':
case 'A':
case 'e':
case 'E':
case 'i':
case 'I':
case 'o':
case 'O':
case 'u':
case 'U':
```

```
printf("\n\n%c is a vowel.\n\n", ch);
break;
default:
printf("%c is not a vowel.\n\n", ch);
}
return 0;
}
```

Q.57 Write a program to Print values of variables and their addresses.

```
# include < stdio.h >
int  main( )
{
int  a ;
int  *p ;
printf(" Enter any integer: ") ;
scanf("%d ",& a) ;
p = &a ;
printf("\n Value of Integer : %d ",a) ;
printf("\n Value of Integer : %d ",*p) ;
printf("\n Value of Integer : %d ",*(&a)) ;
printf("\n Address of Integer : %u ",p) ;
printf("\n Address of Integer : %u ",&a) ;
return ( 0 );
}
```

Q.58 Draw flowchart for checking whether given number is even or odd.



Q.59 List any four keywords used in 'C' with their use.

- int:-The int keyword is used to declare integer type variables.
- return:-The return keyword terminates the function and returns the value.
- struct:- The struct keyword is used for declaring a structure.
- void:-The void keyword meaning nothing or no value.

Q.60 Write the syntax of switch case statement.

```
switch (expression)
{
case constant1:
// statements
break;
case constant2:
// statements
break;
.
.
.
```

```
default:
// default statements
}
```

Q.61 State any two differences between while and do-while statement.

WHILE LOOP	DO - WHILE LOOP
➢ Condition is checked first then statement(s) is executed.	➢ Statement(s) is executed atleast once, thereafter condition is checked.
➢ It might occur statement(s) is executed zero times, If condition is false.	➢ At least once the statement(s) is executed.
➢ while loop is entry controlled loop.	➢ do-while loop is exit controlled loop.

Q.62 State difference between array and string.

ARRAY	STRING
➢ Arrays can hold items of any data type	➢ Strings can hold items of only the char data type.
➢ The array is a data structure	➢ The string is an object.
➢ The length of an array is fixed, whether by the programmer or the user when performing the operation.	➢ The length of a string is not fixed.

Q.63 Declare a structure student with element roll-no and name.

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

Q.64 State four arithmetic operations perform on pointer with example

1. Increment of a Pointer

- When a pointer is incremented, it actually increments by the number equal to the size of the data type for which it is a pointer.

- Ex:- If an integer pointer that stores address 1000 is incremented, then it will increment by 2(size of an int) and the new address it will points to 1002. While if a float type pointer is incremented then it will increment by 4(size of a float) and the new address will be 1004.

2. Decrement of a pointer

- It is a condition that also comes under subtraction. When a pointer is decremented, it actually decrements by the number equal to the size of the data type for which it is a pointer.

- Ex:-If an integer pointer that stores address 1000 is decremented, then it will decrement by 2(size of an int) and the new address it will points to 998

3. Addition of integer to a pointer

- When a pointer is added with a value, the value is first multiplied by the size of data type and then added to the pointer
- Ex:-

```
#include<stdio.h>
int main(){
int number=50;
int *p;//pointer to int
p=&number;//stores the address of number variable
printf("Address of p variable is %u \n",p);
p=p+3;   //adding 3 to pointer variable
printf("After adding 3: Address of p variable is %u \n",p);
return 0;
}
```

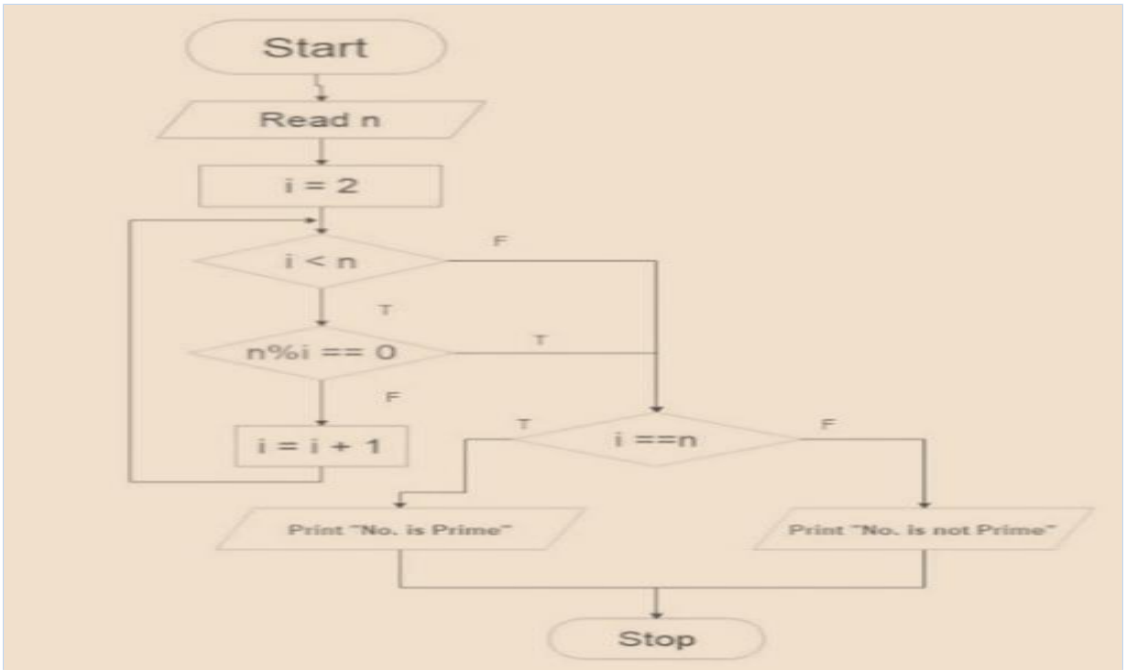
4. Subtraction of integer to a pointer

- We can subtract a value from the pointer variable. Subtracting any number from a pointer will give an address

- Ex:-

```
#include<stdio.h>
int main()
{
int number=50;
int *p;
p=&number
printf("Address of p variable is %u \n",p);
p=p-3;
printf("After subtracting 3: Address of p variable is %u \n",p);
return 0;
}
```

Q.65 Draw flowchart for checking weather given number is prime or not.



Q.66 Write a program to reverse the number 1234 (i.e. 4321) using function.

```
#include<stdio.h>
int main()
{
int number, reverse;
printf("Enter a positive interger: ");
scanf("%d", &number);
reverse = findReverse(number);
printf("The reverse of %d is: %d", number, reverse);
return 0;
}
int findReverse(int n)
{
int sum=0;
while (n!=0)
{
sum = sum*10 + n%10;
n /= 10;
}
return sum;
}
```

Q.67 Differentiate between character array and integer array with respect to size and initialization.

INTEGER ARRAY	CHAR ARRAY
➢ An integer array is an object capable of holding values of type int.	➢ A char array is an object capable of holding values of type char.
➢ Int array take more memory to allocate	➢ char take less memory allocate
➢ int array[10]; //should allocate consequtive 40 bytes	➢ char array[10]; //should allocate consequtive 10 bytes

Q.68 Write a program to sum all the odd numbers between 1 to 20.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int sum=0,i;
clrscr();
for(i=1;i<=20;i++)
{
if(i%2==1)
sum=sum+i;
}
printf("sum of odd no"s between 1 to 20 is %d",sum);
getch();
}
```

Q 69 Explain any four bit-wise operator used in 'C' with example.

1. Bitwise AND (&)

- The output of bitwise AND is 1 if the corresponding bits of two operands is 1. If either bit of an operand is 0, the result of corresponding bit is evaluated to 0.

- Ex:-

```
#include <stdio.h>
int main()
{
int a = 12, b = 25;
printf("Output = %d", a&b);
return 0;
} // Output : 8
```

2. Bitwise OR operator (|)

- The output of bitwise OR is 1 if at least one corresponding bit of two operands is 1

```
#include <stdio.h>
int main()
{
int a = 12, b = 25;
printf("Output = %d", a|b);
return 0;
} // Output: 29
```

3. Right Shift Operator

- Right shift operator shifts all bits towards right by certain number of specified bits. It is denoted by >>.

- Ex

212 = 11010100 (In binary)
212>>2 = 00110101 (In binary)

4. Left shift operator

- shifts all bits towards left by a certain number of specified bits. The bit positions that have been vacated by the left shift operator are filled with 0.

- Ex:

212 = 11010100 (In binary)
212<<1 = 110101000 (In binary)

Q.70 Explain all string functions with examples

1. strlen() - this function returns the length of given string

- Syntax :- strlen(string_name);
- EXAMPLE:

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

```
int main()
{
char a[100];
int length;
printf("Enter a string to calculate its length\n");
gets(a);
length = strlen(a);
printf("Length of the string = %d\n", length);
return 0;
}
```

2. strcpy()- It copies the source string in destination

   - Syntax:- strcpy (destination string_name, source string_name);
   - EXAMPLE:

```
#include<stdio.h>
#include<string.h>
void main()
{
char str1[100],str2[50];
printf("Enter string str1\n");
gets(str1);
strcpy(str2,str1);
printf("Copied String(str2) is %s",str2);
}
```

3. strcat()-This function joins the 2 string and result is stored at the first string

   - Syntax:- strcat(string_name1, string_name2);
   - EXAMPLE

```
#include <stdio.h>
#include<string.h>
int main()
{
char s1[20]; // declaration of char array
char s2[20]; // declaration of char array
printf("Enter the first string : ");
```

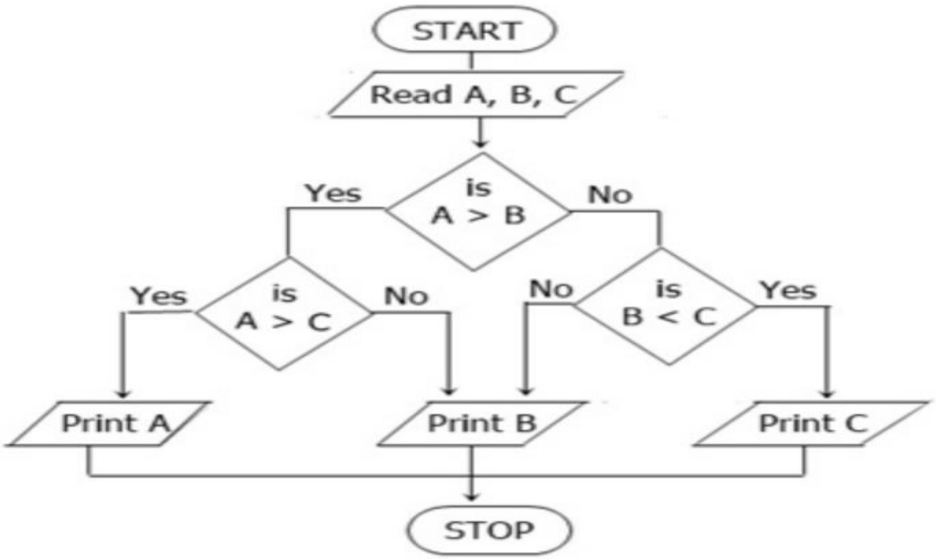
```
scanf("%s", s1);
printf("\nEnter the second string :");
scanf("%s",s2);
strcat(s1,s2);
printf("The concatenated string is : %s",s1);
return 0;
}
```

4. strrev()-This function returns the reverse of string

- Syntax:- strrev(string_name);
- EXAMPLE

```
#include <stdio.h>
#include <string.h>
int main()
{
char str[40];
printf (" \n Enter a string to be reversed: ");
scanf ("%s", str);
printf (" \n After the reverse of a string: %s ", strrev(str));
return 0;
}
```

Q.71 Draw flowchart for finding largest number among three numbers



Q.72 Describe generic structure of ‘C’ program.

- The preprocessor section contains all the header files used in a program.
- It informs the system to link the header files to the system libraries
- main() is the first function to be executed by the computer.
- It is necessary for a code to include the main().
- Main function is further categorized into local declarations, statements, and expressions
- The statements refers to if, else, while, do, for, etc. used in a program within the main function
- Return function is generally the last section of a code. But, it is not necessary to include. It is used when we want to return a value.

- Ex:

```
#include<stdio.h>
int main()
{
int a, b, sum;
printf("Enter two numbers to be added ");
scanf("%d %d", &a, &b);
// calculating sum
sum = a + b;
printf("%d + %d = %d", a, b, sum);
return 0;  // return the integer value in the sum
}
```

Q.73 Write a program to take input as a number and reverse it by while loop.

```
#include <stdio.h>
int main()
{
int num, rnum = 0, rem;
printf("Enter any number: ");
scanf("%d", &num);
```

```
while (num != 0) {
rem = num % 10;
rnum = rnum * 10 + rem;
num = num / 10;
}
printf("\nReverse of input number is: %d", rnum);
return 0;
}
```

Q.74 Write a program to accept 10 numbers in array and arrange them in ascending order.

```
#include <stdio.h>
void main()
{
int i, j, a, n, number[10];
printf("Enter the value of N \n");
scanf("%d", &n);
printf("Enter the numbers \n");
for (i = 0; i < n; ++i)
scanf("%d", &number[i]);
for (i = 0; i < n; ++i)
{
for (j = i + 1; j < n; ++j)
{
if (number[i] > number[j])
{
a = number[i];
number[i] = number[j];
number[j] = a;
}
}
}
printf("The numbers arranged in ascending order are \n");
for (i = 0; i < n; ++i)
printf("%d\n", number[i]);
}
```

Q.75 Explain meaning of following statement with reference to pointers:

- int *a, b; - a is a pointer of integer type and b is a variable of int
- b = 20; - b is a variable which holds value 20
- *a = b; - means value at a be b
- A = &b; - means let A be the address of b

Q.76 Write a program to perform addition, subtraction, multiplication and division of two integer number using function.

```
#include<stdio.h>
int add(int n1, int n2);
int subtract(int n1, int n2);
int multiply(int n1, int n2);
int divide(int n1, int n2);
int main()
{
int num1, num2;
printf("Enter two numbers: ");
scanf("%d %d", &num1, &num2);
printf("%d + %d = %d\n", num1, num2, add(num1, num2));
printf("%d - %d = %d\n", num1, num2, subtract(num1, num2));
printf("%d * %d = %d\n", num1, num2, multiply(num1, num2));
printf("%d / %d = %d\n", num1, num2, divide(num1, num2));
return 0;
}
int add(int n1, int n2)
{
int result;
result = n1 + n2;
return result;
}
int subtract(int n1, int n2)
{
int result;
```

```
result = n1 - n2;
return result;
}
int multiply(int n1, int n2)
{
int result;
result = n1 * n2;
return result;
}
int divide(int n1, int n2)
{
int result;
result = n1 / n2;
return result;
}
```

Q.77 Write a program to accept ten numbers in array. Sort array element and display.

```
#include<stdio.h>
int main()
{
int element[10],i,j,temp;
printf("enter 10 integer numbers:");
for(i=0;i<10;i++)
{
scanf("%d",&element[i]);
}
for(i=0;i<10-1;i++)
{
for(j=i+1;j<10;j++)
{
if(element[i]>element[j])
{
temp=element[i];
element[i]=element[j];
element[j]=temp;
```

```
}
}
}
printf("Elements are now in ascending order:");
for(i=0;i<10;i++)
printf("%d\n",element[i]);
return 0;
}
```

Q.78 Write a program to print reverse of an entered string using pointer.

```
#include <stdio.h>
#include <conio.h>
void main()
{
char *s;
int len,i;
clrscr();
printf("\nENTER A STRING: ");
gets(s);
len=strlen(s);
printf("\nTHE REVERSE OF THE STRING IS:");
for(i=len;i>=0;i--)
printf("%c",*(s+i));
getch();
}
```

Q.79 Explain recursion with suitable example. List any two advantages

- The process in which a function calls itself directly or indirectly is called recursion and the corresponding function until a particular condition is reached is called a recursive function.

- Advantages :
- Reduce unnecessary calling of functions

- It is very logical and extremely useful when applying same solution

```
#include <stdio.h>
int fact (int);
int main()
{
int n,f;
printf("Enter the number whose factorial you want to calculate?");
scanf("%d",&n);
f = fact(n);
printf("factorial = %d",f);
}
int fact(int n)
{
if (n==0)
{
return 0;
}
else if ( n == 1)
{
return 1;
}
else
{
return n*fact(n-1);
}
}
```

Q 80 Write a program to accept ten numbers and print average of it.

```
#include <stdio.h>
void main()
{
int i,n,sum=0;
float avg;
printf("Input the 10 numbers : \n");
```

```
for (i=1;i<=10;i++)
{
printf("Number-%d :",i);
scanf("%d",&n);
sum +=n;
}
avg=sum/10.0;
printf("The sum of 10 no is : %d\nThe Average is : %f\n",sum,avg);
}
```

Q.81 Enlist different format specifiers with its use.

- %c - to accept character value
- %f - to accept float value
- %d - to accept integer value
- %f - to accept double value
- %s - to accept string value
- %p - to accept pointer value
- %% - for printing

Q.82 Find the output of the following program:

```
# include <stdio.h>
void main( )
{
int x = 10, y = 10, v1, v2 ;
v1 = x++ ;
v2 = ++y ;
printf ("value of v1:%d", v1) ;
printf ("value of v2:%d", v2) ;
}
```

Output:-

value of v1 is 10

value of v2 is 11

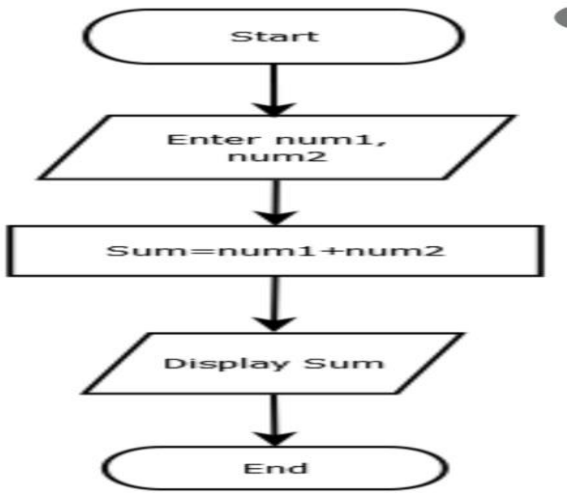
1.

2.
3.
4.
5.
6.
7.

Q.83 State the Relational operators with example

Operator	Meaning of Operator	Example
==	Equal to	`5 == 3` is evaluated to 0
>	Greater than	`5 > 3` is evaluated to 1
<	Less than	`5 < 3` is evaluated to 0
!=	Not equal to	`5 != 3` is evaluated to 1
>=	Greater than or equal to	`5 >= 3` is evaluated to 1
<=	Less than or equal to	`5 <= 3` is evaluated to 0

Q.84Draw flow chart for addition of two numbers.



Q.85 State the importance of flow chart.

- Flowchart are the better way of communicating the logic of the system
- With the help of flowchart problem can be analyzed effectively
- Flowchart act as a guide during the program implementation stage

Q.86 Draw & label different symbols used in flowcharts.

Terminal

Terminal symbol is used to indicate the start and end of the program. Here is the symbol of terminal:

Input/Output

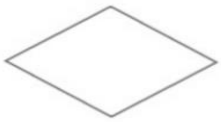
A parallelogram is used to represent an input and output representation or operation. Here is the demo symbol of a input/output used in flowchart:

Processing Symbol

To represent any processing operation in flowchart, we have to use the rectangle symbol. Here is the demo symbol of rectangle used in drawing flowchart to use storage or arithmetic operation:

Decision Symbol

To indicate the step of decision making statement, we have to use diamond shaped box. Here is the demo symbol of diamond shaped box:



Flow of Lines

Flow of lines indicates the path of logic flow in a program. Between each boxes used in flowchart, we have to use lines to indicate where the program flow goes. Here is the demo symbol of lines flow:

Q.87 Write algorithm and draw flow-chart to print even numbers from 1 to 100.

- Algorithm

1. Start
2. Initialize the variable i to 1.
3. while i<=100
4. if i%2==0
5. print the number
6. increment value of i
7. stop

- Flowchart