Write the following expressions in C/C++ / or FORTRAN-95:

(i) x is not equal to y (ii) assign x equal to y (iii) verify whether u is equal to v

 (iv) p is greater than equal to q (v) r is less than equal to s. (vi) absolute value of x (vii) exponential x square. (viii) x to the power 24. (ix) tan inverse x (x) sin inverse y (xi) cos inverse z (xii) x to the power 11 plus x square and whole is divided y square plus 4 times x square.

Solution :-

1. x!=y /\*x then exclamatory sign means not and the equality sign indicate x is not equal to y \*/
2. x=y /\* assigning x equal to y \*/
3. u==v /\* double equality sign means C –compiler will verify whether u is equal to v or not.
4. p>=q /\* p with greater than sign and equality sign means p is greater than equal to q \*/
5. r<=s /\* r with less than sign and equality sign means r is less than equal to s \*/
6. abs(x) /\* abs means absolute value of x \*/
7. exp(x\*x) /\* exp stands for exponential quantity and x square can be written as product of two \*/
8. pow(x,24) /\* pow stands for power , the first number x will be raised to second number 24 \*/
9. atan(x) /\* atan means tan inverse trigonometric function\*/
10. asin(y) /\* asin means sine inverse trigonometric function \*/
11. acos(z) /\* acos means cosine inverse trigonometric function \*/
12. (pow(x,11) + x\*x)/(y\*y +4\*x\*x)

Similarly trigonometric function sinx, cosx and tanx can be written as sin(x), cos(x) and tan(x) .

Question 1. Write the difference between keywords and identifiers.

Keywords:- They are reserved words , that have standard predefined meaning in C. These Keywords can be used only for their intended purpose; they cannot be used as programmers defined identifiers. Keywords are always in small letters it is possible to use upper case keyword as identifiers. Normally it is not done as it is considered a poor programming practice.

Example of keywords :- 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.

Identifiers:- they are names that are given to various program elements, such as variables, functions and arrays.

Identifiers consist of letters and digits, in any order, except that the first character must be a letter. Both upper- and lowercase letters are permitted, though common usage favours the use of lowercase letters for most types of identifiers. Upper- and lowercase letters are not interchangeable ( an uppercase letter is not equivalent to the corresponding lowercase letter). The underscore ( \_ ) can also be included and is considered to be a letter. An underscore is used in the middle of an identifier. An identifier may also begin with an underscore, though this rarely done in practice.

Example of identifiers :- area, sum, total\_number, Time\_table, time123, x1, y1, X1,Y1 perimeter, volume, surface\_area, pi, circle, circumference, PI, value, Number\_of\_student.

Question 2:- Write the following algebraic expression in C/C++ / or FORTRAN-95:-

1. y=sin-1 x + cos-1 x + tan-1 x, (ii) y=$\frac{x^{99}tan^{-1}x +z^{8}}{u^{8}+v^{77}}$ (iii) $e^{x^{2}}+\cos(x)$ +$\left|x\right|$

Solution:- In C expressions can be written as-

1. y=asin(x)+acos(x)+atan(x) .
2. y=(pow(x,99)\*atan(x)+pow(z,8))/(pow(u,8)+pow(v,77).
3. exp(x\*x)+cos(x)+abs(x)

Question 3 :- Write the difference between Compiler and Interpreter.

Solution:- The Compiler is a kind of system software that translates the programs written in high level language to machine language. Example- C-complier, Java-compiler, Pascal-compiler. The compiler converts the whole program to machine language. This converted program may be stored somewhere in memory and then may be executed without requiring the presence of the compiler.

Interpreter is a software similar to the compiler. It is a system software which is used to convert high level language programs to machine language but unlike compiler it converts one line and executed at a time and then converts next line into machine language and executed and so on. Interpreter takes more time in execution than compiler. Example:- Python interpreter.

Question 4:- What are Machine language(low level) and High level language.

Question 5:- write a program to find (i) area of a circle (ii) area and perimeter of a square (iii) volume of a sphere and surface area.

Solution :- (i) /\* program to calculate area of a circle \*/

#include <stdio.h>

int main()

{ float radius, area, pi=3.1416;

printf(“ Enter value of radius = ? “);

scanf(“%f”, &radius);

area=pi\*radius\*radius;

printf(“Area of circle = %f”, area);

return 0

}

1. first 2 lines of the above program same

third and other line as follows

{ int length, area, perimeter;

/\* you may also write float length, area, perimeter\*/

printf(“ Enter length = ? “);

scanf(“%d”, &length); /\* incase if you write float then it should be scanf(“%f”, &length); \*/

area=length\*length;

 perimeter=4\*length;

printf(“Area of square = %d and perimeter= %d”, area, perimeter);

/\* in case if you define variable as float length, area, perimeter; then following change has to be made printf(“Area of square = %f and perimeter= %f”, area, perimeter); \*/

return 0

}

1. first 2 lines of the above program same

third and other line as follows

{ float radius, volume,surface\_area, pi=3.1416;

printf(“ Enter value of radius = ? “);

scanf(“%f”, &radius);

surface\_area=4\*radius\*radius;

volume=4\*pi\*radius\*radius\*radius/3;

printf(“Surface Area= %f and Volume of sphere = %f”, surface\_area, volume);

return 0

}

Question 6

Draw the block diagram of a digital computer and explain any two of them briefly.

Functional units of a digital computer:-

A computer has four main components, they are (i) Input device, (ii) Memory (iii) Central Processing Unit(CPU) and (iv) Output device.

BLOCK DIAGRAM OF DIGITAL COMPUTER

CPU

InpuIt Device

ALU

CU

Output Device

 Memory

Input Device

Input Device

CPU

**CPU**

## C Program to find the roots of Quadratic Equations

**/\*The following program finds out the roots of a quadriatic equation by taking the coefficents as the input from the user.\*/**

**#include<stdio.h>**

**#include<math.h>**

**int main()**

**{**

**double a,b,c;**

**printf("Enter the coefficents of the quadriatic equation of the form ax^2+bx+c\n");**

**scanf("%lf%lf%lf",&a,&b,&c);//We ask for the coefficents from the user**

**double D=pow(b,2)-4\*a\*c;**

**if(D<0)//We check the determinant whether it is greater or lesser than zero and accordingly find real or imaginary roots**

**{**

 **printf("The roots are imaginary\n");**

 **double real,img;**

 **real=(-1\*b)/(2\*a);//The real part of the root**

 **img=sqrt(-1\*D)/(2\*a);//The imaginary part of the root**

 **printf("The first root is %lf+i%lf\n",real,img);**

 **printf("The first root is %lf-i%lf\n",real,img);**

**}**

**if(D>=0)**

**{**

 **printf("The roots are real\n");**

 **if(D!=0)**

**{ //We find both roots if D>0**

 **double root1=(-1\*b+sqrt(D))/(2\*a);**

 **double root2=(-1\*b-sqrt(D))/(2\*a);**

 **printf("The first root is %lf\n",root1);**

 **printf("The second root is %lf\n",root2);**

**}**

**else**

**{ //We find the root if D==0 and display that we have equal roots**

 **double root=(-1\*b)/(2\*a);**

 **printf("The roots are equal\n");**

 **printf("The root is %lf\n",root);**

**}**

**}**

**getch();**

**}**

**Write a C program to generate AP series and find its sum.**
AP series is a series of numbers in which the difference between the consecutive elements is same.

**Example:** 1, 3, 5, 7, 9, 11, 13, 15, 17, 19

Here, the common difference is 2.

And the sum of the AP series can be found by using the below formula.

Sum = n \* (a1 + an)/2  => (10 \* (1 + 19)) / 10 = 100

n is the number of elements in the AP series.
a1 is the first element in the AP series.
an is the last element in the AP series.\*/

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

void main()
/\* If You write  int main() then use return0\*/{
        int i, n, \*data, diff, sum, value = 1;

        /\* get the number of elements in AP series \*/
        printf("Enter the value for number of elements:");
        scanf("%d", &n);

        /\* get the common difference from the user \*/
        printf("Common difference for AP series:");
        scanf("%d", &diff);

        /\* allocate memory to store the elements in AP series \*/
        data = (int \*)malloc(sizeof(int) \* n);

        /\* print the series and store the AP series in data array \*/
        printf("AP series: ");
        for (i = 0; i < n; i++) {
                printf("%d ", value);
                data[i] = value;
                value = value + diff;
        }

        /\* find the sum of the given AP series \*/
        sum = (n \* (data[0] + data[n - 1]))/2;

        /\* print the result \*/
        printf("\nSum of the AP series is %d\n", sum);

**getch (); /\* you can also write return 0;\*/**   }

Enter the value for number of elements:10
  Common difference for AP series:2
  AP series: 1 3 5 7 9 11 13 15 17 19
  Sum of the AP series is 100

/