CONTROL AND CONDITIONAL STATEMENT

Ex no : 1a	PERFECT NUMBERS
Date:	

AIM:

To check whether a number is prime or not.

ALGORITHM:

STEP 1 : start.

STEP 2 : Declare all required variables and Get all the required inputs from the users.

Step 3 : Using while loop , add all the divisors of the respected number.

Step 4 : Check whether the sum is equal to the respected number.

Step 5 : If sum = respected number. Print that it is a perfect number ,Else print It is not a perfect number.

Step 6 : STOP.

Source code:

```
#include<stdio.h>

int main(){

  int n,i=1,sum=0;

  printf("Enter a number: ");

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

```
 while(i<n){
      if(n%i==0)
           sum=sum+i;
          i++;
  }
  if(sum==n)
      printf("%d is a perfect number",i);
  else
      printf("%d is not a perfect number",i);
  return 0;
}
```

Sample output:

```
Enter a number: 6
```

Result:

The Program has been successfully executed and the result is found to be right.

Ex no. 1b	AREA of GEOMENTRIC SHAPES
Date:	

AIM:

To find area of the 4 Geometrical shapes.

ALGORITHM:

Step 1 : START

Step 2 : Declare all required variable and get input for required variables from the user.

Step 3 : Find area of circle using formula pi*r*r ,Rectangle using formula l*b, Triangle using formula 1/2*b*h and Square using formula a*a.

Step 4 : Print The area of the respected shapes.

Step 5 : STOP.

SOUCRCE CODE:

```
   #include <math.h>
#include <stdio.h>
#define PI 3.141

int a_circle()
{
 float r, a;
 printf("Radius: ");
 scanf("%f", &r);
 a = PI * r * r;
 printf("The area of the circle is:%f\n", a);
```

```
  printf("The area of the circle is:%f\n", a);
  return 0;
}

int a_rectangle()
{
   float l,w;
   float area;

   printf("Enter size of each sides of the rectangle : ");
   scanf("%f%f",&l,&w);

   area = l * w;
   printf("Area of rectangle is: %.3f\n",area);

   return 0;

}

int a_square()
{
float number1,number2;
   printf("The length is : ");
   if(scanf("%f",&number1)==1)
   {
      number2=number1*number1;
      printf("The area of square is : %f\n",number2);
   }
```

```
    else
    {
        printf("error,enter correct value");
    }
}

int a_triangle()
{
  float h,w;
    float area;

    printf("Enter height and width of the right angled triangle : ");
    scanf("%f%f",&h,&w);

    area = 0.5 * h * w;

    printf("Area of right angled triangle is: %.3f\n",area);

    return 0;


}

int main(void)
{
 a_circle();
 a_rectangle();
 a_triangle();
 a_square();
```

Sample output:

Radius: 30

The area of the circle is:2826.899902

Enter size of each sides of the rectangle : 10 5

Area of rectangle is: 50.000

Enter height and width of the right angled triangle : 20 10

Area of right angled triangle is: 100.000

Enter The side length of the the square : 20

The area of square is : 400.000000

Result:

The Program has been successfully executed and the result is found to be right.

Ex no. 1c	CALCULATOR
Date:	

AIM:

TO design a calculator using switch case.

Algorithm:

Step 1 : START.

Step 2 : Get two operands and the operator character from the user.

Step 3 : If the operator is ‗+' then add the operands.

Step 4 : If the operator is ‗-' then subtract one operand from the other operand.

Step 5 : If the operator is ‗*' then multiply the operands

Step 6 : If the operator is ‗/' then divide one operand by the other.

Step 7 : If the entered character is not matching any of these above four operators then print the default statement ―Enter the correct operator‖

Step 8 : STOP.

Source code:

```
#include <stdio.h>
int main()
{
   char o;
   float num1,num2;
   printf("Enter operator either + or - or * or / : ");
   scanf("%c",&o);
   printf("Enter two operands: ");
   scanf("%f%f",&num1,&num2);
   switch(o) {
      case '+':
         printf("%.1f + %.1f = %.1f\n",num1, num2, num1+num2);
         break;
      case '-':
         printf("%.1f - %.1f = %.1f\n",num1, num2, num1-num2);
         break;
      case '*':
         printf("%.1f * %.1f = %.1f\n",num1, num2, num1*num2);
         break;
```

```
        case '/':
            printf("%.1f / %.1f = %.1f\n",num1, num2, num1/num2);
            break;
        default:
            /* If operator is other than +, -, * or /, error message is shown */
            printf("Error! operator is not correct \n");
            break;
    }
    return 0;
}
```

Sample output:

Enter operator either + or - or * or / : *

Enter two operands: 3.14

99.2

3.1 * 99.2 = 311.5

Result:

The Program has been successfully executed and the result is found to be right..

Ex no. 1 d	NUMBERS DIVISIBLE BY 5
Date:	

AIM:

To find 'n' numbers divisible by 5.

Algorithm:

Step 1 : Start.

Step 2. Read the value of n.

Step 3 : Make a loop Declare variable i=4 Increment i by 1 upto n

Step 4 : if i%5=0 print i , else print " ".

Step 6 : Stop

Source code:

```
#include<stdio.h>
Void main()
    {
 Int  n,I;
 Puts("Enter the limit-");
 Scanf("%d",&n);
 For(i=4;i<=n;i++)
 If(n%5==0)
 Printf("%d ",i);
 }
```

Sample output:

```
Enter the limit-27
5     10  15  20  25
```

Result:

The Program has been successfully executed and the result is found to be right..

Ex no. 1 e	SUM,REVERSE and LARGEST of digits
Date:	

AIM:

To find the sum of digits ,reversal of digits and largest of digits.

Algorithm:

Step 1 : Start.

Step 2 :Accept the number, N

Step 3 : Assign max to 0.

Step 4 : Extract digit by digit and sum the digits in a variable sum.

Step 5 : Find the reverse of the number N by extracting the last digit, multiplying by 10 and reducing the number and store the reverse number in a variable rev

Step 6 : Extract digit by digit and compare it with max and store the largest digit in max.

Step 7 : print sum , rev , max.

Step 8 : Stop

Source code :

```
#include<stdio.h>
int main(void)
{
   int r=0,n,sum=0,d,lar=0;
   printf("Enter the number:");
   scanf("%d",&n);
   while(n>0)
   {
     d=n%10;
     sum=sum+d;
     r=(r*10)+d;
     n=n/10;
     if(lar<d)
       lar=d;
   }
   printf("\n sum of digits= %d",sum);
   printf("\n Reverse=%d",r);
   printf("\n largest of digits= %d",lar);}
```

Sample output:

Enter the number:123456

sum of digits= 21

Reverse=654321

largest of digits= 6

Result:

The Program has been successfully executed and the result is found to be right.

Ex no. 1 f	Armstrong number
Date:	

Aim :

To generate armstrong numbers from 1 to 1000..

Algorithm :

Step 1 : Start.

Step 2. Make a loop Declare variable i=1 Increment i by 1 upto 1000

Step 3 : Find the sum of the cube of the individual digits of the number i and store in SUM1.

Step 4 : If SUM1 equals to i then print I else print " ".

Step 5 : Stop

Source code :

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

int main(void)
{
   int number, temp, digit1, digit2, digit3;
   number = 001;
    printf("\n Armstrong number from 1 to 1000 as follows \n");
   while (number <= 900)
   {
      digit1 = number - ((number / 10) * 10);
      digit2 = (number / 10) - ((number / 100) * 10);
      digit3 = (number / 100) - ((number / 1000) * 10);
      temp = pow(digit1,3) + pow(digit2,3) + (digit3,3);

      if (temp == number)
      {
         printf("%d\t", temp);
      }
      number++;
   }
puts("\n");
}
```

Sample output:

Armstrong number from 1 to 1000 as follows

1 153 370 371 407

Result:

The Program has been successfully executed and the result is found to be right.

Ex no. 1 g	Prime numbers.
Date:	

AIM :

To generate first 'n' prime numbers..

Algorithm:

Step 1 : Start

Step 2 : Read the value of n.

Step 3 : To print the value 2 on screen.

Step 4 : Make a loop Declare variable i=3 Increment i by 1 upto n

Step 5 : Using another loop Declare variable j=2 From j=2 to j< i; divide i by j If i % j = = 0 then print the number.

Step 6 : Stop

Source code :

```
#include<stdio.h>
 int main(void)
{
  int n, i = 3, count, c;

  printf("Enter the number of prime numbers required\n");
  scanf("%d",&n);

  if ( n >= 1 )
  {
    printf("First %d prime numbers are :\n",n);
    printf("2\n");
  }

  for ( count = 2 ; count <= n ;  )
  {
    for ( c = 2 ; c <= i - 1 ; c++ )
    {
      if ( i%c == 0 )
        break;
    }
    if ( c == i )
    {
      printf("%d\t",i);
      count++;
```

```
    }
      i++;
   }


   puts("\n");
}
```

Sample output :

```
Enter the number of prime numbers required
5
First 5 prime numbers are :
2  3  5  7  11
```

Result:

The Program has been successfully executed and the result is found to be right.

Ex no.	
Date:	