ASSIGNMENT -05

Q1.Find the sum of first 10 natural numbers.

```
#include <stdio.h>
int main() {
int sum=0,i,n;
printf("Enter the 10 natural number :");
scanf("%d",&n);
for(i=1;i<=10;i++)
 {
sum=sum + i;
printf("%d", i);
  }
printf("\n The sum is :%d\n", sum);
return 0;
}
```

OUTPUT :

```
Enter the 10 natural number :12345678910
The sum is :55
```

Q2. Display the multiplication table of a given integer .

```
#include <stdio.h>
int main(){
 int i,n;
 printf("enter the given integer :");
scanf("%d", &n);
 i=1;
 while(i<=10){
 printf("%d * %d = %d \n",n,i,n*i);
   ++i;     }}
```

OUTPUT :

```
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

Q3. Display the n terms of odd natural number and their sum.

```
#include <stdio.h>
int main(){
int i,n,sum=0;
printf("enter the given :");
scanf("%d", &n);
i=1;  do  {
printf("%d ", 2*i-1);
sum=sum+i-1;
i++;  }
while(i<=10);  {
printf("\nthe sum of odd number :",sum);
 return 0; }}
```

OUTPUT :

```
enter the given :4
1 3 5 7 9 11 13 15 17 19
the sum of odd number :
```

Q4. Display the pattern like right angle triangles.

```
#include <stdio.h>
int main(){
int i,j;
for (i=1;i<=4;i++)
   {
for(j=1;j<=i;j++)     {
printf("*");
 }
printf("\n");}
return 0;}
```

OUTPUT :

```
*
**
***
****
```

Q5. Display the pattern like right angle triangles.

```
#include <stdio.h>
int main() {
int n,i=1,j,k=1;
printf("Please enter the number of rows :");
scanf("%d", &n);
while(i<=n)    {
 j=1;
while(j<=i)        {
printf("%d", k++);
 j++;        }
i++;
printf("\n");    }
return 0;}
```

OUTPUT :

```
Please enter the number of rows :4
1
23
456
78910
```

Q6. Make such a pattern like a pyramid with numbers

```
#include <stdio.h>
int main() {
 int i=1,j,k,n,t=1,g;
printf("Enter the value for n :");
scanf("%d",&n);
g=n+4-1;    do    {
for (k=g;k>=1;k++)
printf("%d", t++);
printf("\n\n\n");
g--;
 i++;    }
while(i<=n);
return 0;
}
```

OUTPUT :

```
Enter the value for n :4
       1
      23
     456
    78910
```

Q7. Display Pascal's triangle.

```
#include <stdio.h>
int main() {
int rows, coef = 1, space, i, j;
printf("Enter the number of rows: ");
scanf("%d", &rows);
for (i = 0; i < rows; i++) {
for (space = 1; space <= rows - i; space++)
printf("  ");
for (j = 0; j <= i; j++) {
if (j == 0 || i == 0)
coef = 1;
else
coef = coef * (i - j + 1) / j;
printf("%4d", coef);      }
printf("\n");    }
 return 0;
}
```

OUTPUT :

```
Enter the number of rows: 5
           1
         1   1
       1   2   1
     1   3   3   1
   1   4   6   4   1
```

Q8. Display the first n term of Fibonacci series.

```
#include <stdio.h>
int main() {
int prv=0,pre=1,trm,i,n;
printf("Input number of terms to display : ");
 scanf("%d",&n);
printf("Here is the Fibonacci series upto to %d terms : \n",n);
printf("% 5d % 5d", prv,pre);
for(i=3;i<=n;i++)   {
 trm=prv+pre;
printf("% 5d",trm);
 prv=pre;
pre=trm;   }
printf("\n");}
```

OUTPUT :

```
Input number of terms to display : 10
Here is the Fibonacci series upto to 10 terms :
    0     1    1    2    3    5    8   13   21   34
```

Q9. Check whether a given number is a perfect number or not.

```
#include <stdio.h>
 int main()  {
int num, count = 1, sum = 0;
printf("Enter a number\n");
 scanf("%d", &num);
while(count < num)      {
if(num%count == 0)          {
sum = sum + count;          }
count++;       }
if(sum == num)      {
printf("\n%d is a perfect number\n", num);      }
 else      {
printf("\n%d is not a perfect number\n", num);      }
 return 0;  }
```

OUTPUT :

```
Enter a number
3
3 is not a perfect number
```

Q10. Find the Armstrong number for a given range of number.

```
#include <stdio.h>
int main() {
int num, originalNum, remainder, result = 0;
printf("Enter a three-digit integer: ");
scanf("%d", &num);
originalNum = num;
while (originalNum != 0) {
remainder = originalNum % 10;
result += remainder * remainder * remainder;
originalNum /= 10;    }
if (result == num)
printf("%d is an Armstrong number.", num);    else
printf("%d is not an Armstrong number.", num);
return 0;
}
```

OUTPUT :

```
Enter a three-digit integer: 153
is an Armstrong number.
```

Q11. Determine whether a given number is prime or not.

```
#include <stdio.h>
int main() {
int n,i=2,flag=0;
printf("Enter the value :");
scanf("%d", &n);
while(i<=2) {
if(n%i==0){
flag=1;
break;        }
 ++i;    }
if(n==1)     {
 printf("1 is neither prime nor composite");    }
 else    {
if(flag==0)
printf("%d is a prime number", n);
else
printf("%d is not prime number", n);    }
return 0;
}
```

OUTPUT :

```
Enter the value :34
is not prime number
```

Q12. Display the number in reverse order.

```
#include <stdio.h>
int main() {
 int n, rev = 0, remainder;
printf("Enter an integer: ");
scanf("%d", &n);
do{
remainder = n % 10;
 rev = rev * 10 + remainder;
 n /= 10;    }
 while (n != 0);
printf("Reversed number = %d", rev);
return 0;
}
```

OUTPUT :

```
Enter an integer: 12345
Reversed number = 54321
```

Q13. Display the sum of the series [9+99+999+9999…..].

```
#include <stdio.h>
int main()  {
long int n,i,term=9;
int sum =0;
printf("the value :");
scanf("%ld", &n);
for(i=1;i<=n;i++)    {
sum=sum+term;
 printf("%ld    ", term);
 term=term*10+9;    }
 printf("\n the series %d\n", sum);
return 0;}
```

OUTPUT :

```
the value :5
9    99    999    9999    99999
 the series 111105
```

Q14. Find the sum of the series[1-x^2/2!+x^4/4-…..].

```
#include <stdio.h>
int main() {
float x,sum,t,d;
int n,i=1;
printf("Enter the value x:");
scanf("%f", &x);
printf("Enter the value n:");
scanf("%d", &n);
 sum=1;
t=1;
while(i<=n) {
d=(2*i)*(2*i-i);
t=-t*x*x/d;
sum=sum+t;
 i++;
}
printf("the sum =%f\n value of n=%d\n value of x=%.2f\n", sum,n,x);
  }
```

OUTPUT :

```
Enter the value x:2
Enter the value n:5
the sum =-0.196667
 value of n=5
 value of x=2.00
```

Q15. Find the sum of the series[x-x^3+x^5+…..]

```
#include <stdio.h>
int main() {
int x,sum,ctr,i=1,n,m,mm,nn;
printf("Enter the value x:");
scanf("%d", &x);
printf("Enter the value n:");
scanf("%d", &n);
sum=x;
m=-1;
printf("The value of the series :\n");
printf("%d\n", x);
do    {
ctr=(2*i+1);
mm=pow(x,ctr);
nn=mm*m;
printf("%d\n", nn);
sum=sum+nn;
m=m*(-1);
i++;        }
while(i<n);
printf("\n The sum=%d\n", sum);
 }
```

OUTPUT :

```
Enter the value x:2
Enter the value n:4
The value of the series :
2
-8
32
-128

 The sum=-102
```