Assingment - 7
Iterative control Statement

(1) WAP to find the Nth term of the Fibonnasi Series.

Ans -

```
# include<stdio.h>
int main( )
{ int prev = 0, cur = 1, next = 0, n, i ;
  printf ("enter a number\n") ;
  Scanf ("%d", &n) ;

  printf ("1") ;
  for (i = 0 ; i<n-1 ; i++ )
  {
    next = prev + cur ;
    printf ("%d", next ) ;
    prev = cur ;
    Cur = next ;
  }
}
```

(2) WAP to print first N term of Fibonacci Series.

Ans -

```
# include < stdio.h>
int main( )
{ int prev = 0 , cur = 1, next = 0, n, i ;
  printf ("enter a number\n") ;
  Scanf ("%d", &n) ;

  printf ("1")
    for (i = 0; i<n-1 ; i++ )
```

```
{
    next = prev + cur;
    prev = cur;
    cur = next;
}
printf("%d", next);
}
```

(3) WAP to check whether a given number is there in the fibonaci Series or not.

Ans.

```
# include <stdio.h>
int main()
{ int a, b, i;
printf("Enter 2 numbers\n");
Scanf("%d %d", &a, &b);

for(i= 0; 1; i++)
{

  next = prev + cur;
  prev = cur;
  cur = next;
if (next == n)
 {
    printf("Number found");
     break;
}
if (next > n)
{
    printf("Not found");
    break;
}
```

(4) WAP to calculate HCF of two numbers.

Ans

```
#include <stdio.h>
int main()
{
  int a, b, i, hcf = 1;
printf("enter 2 number \n");
Scanf ("%d %d", &a, &b);

int min = a<b ? a : b;
for (i=1; i<= min ; i++)
{  if(a%i==0) && (b%i ==0)
      hcf = i;

}
printf("HCF is %d", hcf);
 return 0;
}
```

(6) WAP to print all prime numbers under 100

Ans.

```
  # include<stdio.h>
   int main()
{ int i, n, flag = 0;

for(n=1; n<=100 ; n++)
{
   flag = 0;
   for(i=2; i<= n/2 ; i++)
{
   if (n%i==0)
    flag = 1;
}
```

```
    if (flag == 0)
        printf ("%d", n);
    }
    return 0;
}
```

(7) WAP to print all prime numbers between two given numbers.

Ans

```
#include <stdio.h>
int main()
{
    int i, n, flag = 0;
    for (n = 10; n <= 50; n++)
    {
        flag = 0;
        for (i = 2; i <= n/2; i++)
        {
            if (n % i == 0)
            flag = 1;
        }
        if (flag == 0)
            printf ("%d", n);
    }
    return 0;
}
```

(8) WAP to find next prime number of a given number.

Ans

```
int i, n, a = 0;
for (n = 20; 1; n++)
{
    a = 0
```

```
for (i = 2 ; i <= n/2 ; i++ )
{
   if (n%i == 0)
      a = 1 ;
}
if (a == 0)
{
   printf ("%d", n);
      break ;
}
}
   return 0;
}
```

(11) WAP to find the position of first 1 in LSB.

Ans

```
int main( )
{ int x = ? , count = 0 ;
int   result = 0 ;

while (x != 0)
{
result = x & 1 ;
Count++ ;
      if(result == 1)
{
   printf("%d", Count);
   break ;
}
      x = x >> 1 ;
}
      return 0;
```

(5) WAP to check whether two given numbers are co-prime numbers or not

Ans-

```
int main()
{ int a, b, i, hcf = 1;
printf("enter 2 numbers\n");
Scanf("%d %d", &a, &b);

int min = a<b ? a : b;
for(i=1; i<=min; i++)
{
  if(a%i==0) && (b%i==0)
    hcf = i;
}
}

if(hcf=1)
{ printf("%d and %d are co-prime numbers", a, b);
}
else
{ printf("%d and %d are not co-prime numbers", a, b);

}
```

(9) Write a program to check whether a given number is an Armstrong number or not

```
#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; // contains last digit
result += remainder * remainder * remainder;
// removing last digit from the original number
originalNum /= 10;
}
if (result == num)
  printf("%d is an Armstrong number", num);
else
  printf("%d is not an Armstrong number", num);
}
```

(10) Write a program to print all Armstrong numbers under 1000

```
#include<stdio.h>
int main()
{  int num, count = 1, rem, Sum;
while (count <= 1000)
{
   num = count;
   Sum = 0;
   while (num)
   {
      rem = num % 10;
      Sum = Sum + (rem * rem * rem);
      num = num / 10;
}
   if (count == Sum)
{ printf("%d is a Armstrong number \n", count);
}
```