**Assignment – 03**

1. **Display Fibonacci Series Using for Loop**

class Main {

public static void main(String[] args) {

int n = 10, firstTerm = 0, secondTerm = 1;

System.out.println("Fibonacci Series till " + n + " terms:");

for (int i = 1; i <= n; ++i) {

System.out.print(firstTerm + ", ");

// compute the next term

int nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

secondTerm = nextTerm;

}

}

}

**Output:**

Fibonacci Series till 10 terms:0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

**2)Display Fibonacci series using while loop**

class Main {

public static void main(String[] args) {

int i = 1, n = 10, firstTerm = 0, secondTerm = 1;

System.out.println("Fibonacci Series till " + n + " terms:");

while (i <= n) {

System.out.print(firstTerm + ", ");

int nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

secondTerm = nextTerm;

i++;

}

}

}

Output:

Fibonacci Series till 10 terms:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

1. **Display Fibonacci series up to a given number**

class Fibonacci {

public static void main(String[] args) {

int n = 100, firstTerm = 0, secondTerm = 1;

System.out.println("Fibonacci Series Upto " + n + ": ");

while (firstTerm <= n) {

System.out.print(firstTerm + ", ");

int nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

secondTerm = nextTerm;

}

}

}

**Output:**

Fibonacci Series Upto 100: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89,

**4)GCD for both positive and negative numbers**

#include <stdio.h>

int main()

{

int j1, j2;

printf("Enter two integers: ");

scanf("%d %d",&j1,&j2);

// if user enters negative number, sign of the number is changed to positive

j1 = ( j1 > 0) ? j1 : -j1;

j2 = ( j2 > 0) ? j2 : -j2;

while(j1!=j2)

{

if(j1 > j2)

j1 -= j2;

else

j2 -= j1;

}

printf("GCD = %d",j1);

return 0;

}

Output:

Enter a number : 81Enter a number -153GCD of 81 and -153 is 9

1. **LCM using while Loop and if Statement**

#include <stdio.h>

int main()

{

int number\_1, number\_2, minMultiple;

printf("Enter two positive integers: ");

scanf("%d %d", &number\_1, &number\_2);

// maximum number between number\_1 and number\_2 is stored in minMultiple

minMultiple = (number\_1 > number\_2) ? number\_1 : number\_2;

// Always true

while(1)

{

if( minMultiple%number\_1==0 && minMultiple % number\_2==0 )

{

printf("The LCM of %d and %d is %d.", number\_1, number\_2,minMultiple);

break;

}

++minMultiple;

}

return 0;

}

Output:

Enter two positive integers: 72

120LCM

The LCM of 72 and 120 is 360.

1. **Calculate LCM using GCD**

#include <stdio.h>

// Recursive function to return gcd of a and b

int gcd(int a, int b)

{

if (a == 0)

return b;

return gcd(b % a, a);

}

// Function to return LCM of two numbers

int lcm(int a, int b)

{

return (a / gcd(a, b)) \* b;

}

// Driver program to test above function

int main()

{

int a = 15, b = 20;

printf("LCM of %d and %d is %d ", a, b, lcm(a, b));

return 0;

}

Output:

LCM of 15 and 20 is 60

1. **Display uppercase alphabet using for loop**

#include <stdio.h>

int main() {

char c;

for (c = 'A'; c <= 'Z'; ++c)

printf("%c ", c);

return 0;

}

Output:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

1. **Display lowercase alphabet using while loop**

#include <stdio.h>

int main()

{

//loop counter or a variable that

//will store initial alphabet,

//from where we will print the alphabets

char alphabet;

//assigning 'a' as initial alphabet

alphabet='a';

//print statement

printf("Lowercase alphabets:\n");

//loop statement, that will check the condition

//and print the alphabets from 'a' to 'z'

while(alphabet<='z')

{

//printing the alphabets

printf("%c ",alphabet);

//increasing the value by 1

alphabet++;

}

return 0;

}

**Output:**

Lowercase alphabets:

a b c d e f g h i j k l m n o p q r s t u v w x y z

1. **Count Number of Digits in an Integer using while loop**

#include <stdio.h>

int main() {

long long n;

int count = 0;

printf("Enter an integer: ");

scanf("%lld", &n);

// iterate at least once, then until n becomes 0

// remove last digit from n in each iteration

// increase count by 1 in each iteration

do {

n /= 10;

++count;

} while (n != 0);

printf("Number of digits: %d", count);

}

**Output:**

Enter an integer: 9812

Number of digits: 4

1. **Count Number of Digits in an Integer using for loop**

#include <stdio.h>

int main() {

long long n;

int count = 0;

printf("Enter an integer: ");

scanf("%lld", &n);

// iterate at least once, then until n becomes 0

// remove last digit from n in each iteration

// increase count by 1 in each iteration

do {

n /= 10;

++count;

} while (n != 0);

printf("Number of digits: %d", count);

}

**Output:**

Enter an integer: 00009812

Number of digits: 4

1. **Reverse a Number using a while loop in Java**

public class ReverseNumberExample1

{

public static void main(String[] args)

{

int number = 987654, reverse = 0;

while(number != 0)

{

int remainder = number % 10;

reverse = reverse \* 10 + remainder;

number = number/10;

}

System.out.println("The reverse of the given number is: " + reverse);

}

}

**Output:**

The reverse of the given number is: 456789