ASSIGNMENT-02.

COURSE CODE: CSA0914.

COURSE NAME: programming in Java

Name : V.Bhavani

Regno : 192210198.

Submitted to:

Dr. Hemavathi R.

1

Aim : To write a java program to reverse a number

PSeudocode :

```
* Take an integer input from the user.
* Initialise a variable reversed to 0.
* While the input is greater than zero
    * Take the last digit of the input number by finding
      the remainder of the number when divided by 10.
    * Add the last digit to the reversed after shifting
      its current value to the left by one digit
* print the reversed number.
```

program:

```
public class reverse number {
  public Static void main (String[] args){
    int num = 1234;
    int reversed = 0;
    while (num>0) {
      int last digit = num%10;
      reversed = reversed * 10 + lastdigit;
      num/=10;
    }
    System.out.println ("reversed number:"+reversed);
  }
}
```

Input:

Enter a number:

1234

Output:

reversed numbers

4321

② Aim: To write a Java program to Check Armstrong number or not using while Loop.

pseudocode:

```
* Take an integer input from the user.
* Calculate the number of digits in the input number
* Initialise a variable 'Sum' to 0.
* for each digit in the input number.
    → Raise the digit to the power of the number of digits.
    → Add the result to the sum variable.
* Check if the Sum is equal to original number
* print the result
```

Program:

```
public class Armstrongnumber {
    public static void main(String[] args){
        int num = 153;
        int numdigit = countdigits(num);
        int Sum =0;

        int temp = num;
        while (temp >0) {
            int digit = temp % 10;
            Sum += (int) math.pow(digit, numdigits);
            temp /=10;
        }
    if (Sum ==num)
    {
     System.out.println("Armstrong number:");
    }
else
  {
  System.out.println ("not an armstrong number:");
  }
}
public static int count digits(int num)
{
 int count=0;
 while (num>0){
   num/=10;
   count++;
  }
  return count;
  }
}
```

Input:

Enter a number : 153

output:

Armstrong number

3) Aim: To write a java program to calculate the gcd of two numbers.

Pseudocode:

```
* Take two integers from the user n1 and n2
* If n2 is 0 return n1 as the gcd.
* otherwise calculate the remainder of n1 divided by
  n2 and store it in a temporary variables temp.
* Replace n1 with n2 and n2 with temp.
* final values of n1 is the gcd.
```

program:-

```
public class gcd {
    public static void main (String() args) {
      int n1 = 12, n2 = 15)
      int gcd = calculate gcd (n1, n2)
     System.out.println (the gcd of "+n1 + "and" + n2 +
                                   "is" +gcd);
    }
   public static int calculate gcd ( int n1 , int n2) {
   while (n2 ≠ 0) {
      int temp = n1 % n2 ;
        n1 = n2 ;
        n2 = temp;
     }
      return n1 ;
    }
  }.
```

Input:

```
Enter first number : 12
Enter second number : 15
```

output:

```
The gcd of 12 and 15 is 3.
```

4) Aim:-

To write a java program to merge two sorted arrays;

Pseudocode:-

```
* Initialize the variables.
* Create a new array result that is equal to the
    size of both the arrays.
* Initialize three indices i to 0, j to 0 k to 0
* while i is less than arr1, j is less than arr2
```

* print the result.

program:

```
public class mergesorted arrays {
    public static void main(String()args) {
     int arr1 () = { 1, 3, 5, 7 }
     int arr2 () = { 2 ,4, 6, 8 }
     int result ( = merge sortedArrays ( arr1, arr2);
    System. out. println ("merged array !" + java. util. toString
   }                                          (result));
  public static int () mergedsortedarray( int arr1(), int arr())
  {
  int result () = new int (arr1. length + arr2. length);
 While ( i< arr1. length && j <arr2. length) {
   if (arr(i) ≤ arr2 (j)){
    result [k++]= arr1(i++);
   }
   else {
    result (k++) = arr2[j++];
   }
  }
  While ( i <arr1. length) {
       result [k++] = arr1 (i++);
  }
  While (j<arr2, length) {
   result (k++)= arr2(j++);
  }
```

output:

```
[1,2,3,4,5,6,7,8]
```

⑤ Aim: To write a java program to count the frequency of characters in the string.

pseudocode:

```
* Take a string 'input' as input.
* Create a hashmap char frequency to store the frequency of each character.
* Initialize an empty hashmap 'char frequency'.
* Iterate through each character 'c' in the input string.
    → If c is already a key 'char frequency' increment value by 1.
    → otherwise add 'c' as a new key in char frequency with a value by 1
* Return the 'char frequency hashmap.
```

program:-

```
import java.util.Hashmap;
import java.util.map;
    public class character frequency {
        public static void main(String[] args) {
         String input = "hello world";
         map<character, integer> char frequency
```