ASSIGNMENT-2

NAME	:	L. Jaithra
REG NO	:	192210259
COURSE CODE	:	CSA0914
COURSE NAME	:	programming in java for Raspberry Pi

Submitted To:-

Dr. Hemavathi R

Aim:- To write a Java program to Reverse a number.

Pseudocode:-

```
• Take an integer input from the user.
• Initialize 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 reversenumber{
    public static void main(string[] args){
        int num= 1234;
        int reversed =0;
        while (num>0) {
            int lastDigit=num%10;
            reversed=reversed*10+lastDigit;
            num/=10;
        }
        System.out.println("Reversed number:"+
                                        reversed);
    }
}
```

Input

Enter anumber:

1234

Output:-

Reversed Number

4321

2. **Aim:-** To write a java program to check Armstrong number or not using ~~java~~ while loop.

Pseudocode:-

```
• Take an integer input from the user
• Calculate the number of digits in the input number.
• Initialize 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 numDigits = 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 countDigits (int num)
{
   int count=0
   while (num>0) {
      num /=10;
      count++;
   }
   return count;
 }
}
```

Input:-

Enter a Number
153

Output:-

Armstrong number.

Aim:- To Write a java program to calculate the GCD of Two numbers.

Pseudocode:-

- Take two integers from user n_1 and n_2
- If n_2 is 0 return n_1 as the GCD
- otherwise calculate the remainder of n_1 divided n_2 and store it in a temporary variable temp.
- Replace n_1 with n_2 and n_2 with temp.

• Final value of n_1 is the GCD.

Program:-

```
public class GCD{
  Public static void main(String[] args){
    int n1 = 12;
    int n2 = 15;
    int gcd = calculateGCD(n1, n2)
    System.out.println("The GCD of "+ n1 + "and"+
  }                     "n2"+ "is"+ gcd);
 public static int calculateGCD(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
```

Aim:- To write a java program to merge two sorted arrays.

Pseudocode:-

• Inititilize 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 and k to 0
- while i is less than arr1, j is less than arr2
- print the result.

~~pseudocode~~
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= mergeSortedArrays(arr1,arr2)
   System.out.println("Merged array:"+ java.util.
                           tostring(result));
 }
 public static int[] mergesortedArray(int arr1[],
                                     int arr2[])
 {
  int result[] =new int[arr1.length+arr2.length];
  while(i<arr1.length && j<arr2.length){
    if(arr1[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++];
        }
        return result;
    }
}
```

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 of 1
• Return the 'char frequency Hashmap
```

Program:-

```
import java.util.HashMap;
import java.util-Map;
public class characterFrequency{
    public static void main(String[] args){
```

```
string input="hello world";
map<character,Integer>charfrequency=
    countcharacterFrequency(input);
system.out.println("character frequency:"+
                    charfrequency);
}
public static map<character,Integer>countchara
    -terFreqcueny(String input) {
map<character, Integer>charfrequency=new
                         HashMap<>();
for(char c:input.tocharArray()){
  if(charfrequency.containkey(c)){
     Char frequency.put(c,charfrequency.get(c)+1);
  }
  else{
   char frequency.put(c,1);
  }
 }
 return charfrequency;
}
}
```

Output:-

```
character Frequency={h=1, e=1, l=3, o=2, w=1, r=1,
                     d=1}.
```