26/8/24

CSA0979 - JAVA PROGRAMMING

ASSIGNMENT-2

S. Mani Sankar

192224196

1. Reverse Number:

AIM:

To Reverse a Number in Java program of given digits.

Pseudo code:

→ Initialize a variable to store the reversed num, '0'
→ use while loop to extract each digit from org num by rem!0
→ Append & Remove the last digit from org num by divd it by 10
→ Repeat the steps. 2-4 untitil org Num becomes '0'

PROGRAM:

```
import java.util.Scanner;
Public class Rev Num {
    Public static void main (String [] args){
        Scanner Sconner = New Scanner (System.in);
        System.out.print ("Enter a num" );
        int number = sconner . nextInt();
        int revNum = 0;
        while (Number != 0) {
            int digit = Number % 10;
            reversed number = reversed number * 10 + digit;
            Number / = 10;
        }
        System.out.println (" Rev numb: " + Reversed Num);
    }
}
```

Output:

12345 o/p. 54321

2. Amstrong Number:

AIM:

To write a Java program for Amstrong Number

Pseudo code:

1. Initialize a variable as org Num
2. use while loop for this as syntax dig = N %.10
3. Then sum = sum + (digit * digit * digit) . N=Num /10.
4. Then display the output.

Program:

```
Import java.util.scanner;
Public class Amstrong Num {
    Public static void main (string [] args) {
        Scanner sCanner = new Scanner (system.in);
        System.out.print ("Enter a Num : ");
        int Num = Scanner . next Int ();
        int Original Number = Num ;
        while ( Num != 0 ) {
            int digit = Num %.10;
            Sum += digit * digit * digit;
            Num /=10;
        }
    if (sum == org Num) {
        System . out . println ( org Num + " is Amstrong Num.");
    } else {
        System . out . println ( org Num + " is not Amstrong ");
    }
}
```

Output:

153 is a Armstrong Number

3. GCD:

AIM:

To write a Java program for GCD calculation.

Pseudo code:

1. Initialize the variable as Num1, Num2.
2. Use while loop as (b != 0) condition statement
3. Then use the temp as the temporary variable
4. Diplay the output.

Program:

```
Import  Java.util.Scanner;
public  class GCD {
        public static void main (String[] args) {
            Scanner scanner = new Scanner (System.in);
            System.out.print ("Enter the 1st Number: ");
            int num1 = scanner.nextInt();
            System.out.print ("Enter the 2nd Num: ");
            int num2 = scanner.nextInt();
            int gcd = find GCD (num1, num2);
            System.out.println ("GCD is" + gcd);
        }
    public static int findGCD (int a, int b) {
            while (b != 0) {
                int temp = b;
                b = a % b;
                a = temp;
            }
        return a;
    }
}
```

Output:

12, 18 GCD = 6

4. merge two sort:

AIM:

To write a Java program for merge two sort.

Pseudo code:

1. Initialize two pointer, one for each array
2. Compare the elements using the if else condition.
3. Create result array and while loop syntax
4. Display output of sorted.

program:

```
import java.util.Arrays;
public class merge sorted arrays {
    public static void main (String [] args() {
        int [] arr1 = { 1, 2, 3 }
        int [] arr2 = { 2, 4, 6 }
        int [] merged array = merge Arrays (arr1, arr2);
        System.out.println ("merged array" + Arrays.toString (mergedArray));
    }
int i=0, j=0, k=0;
while (i < n1 && j <= n2) {
    if (arr1[i] <= arr2[j]) {
        result[k++] = arr1[i++];
    } else {
        result[k++] = arr2[j++];
    }
}
return result;
}
}
```

output:

[1, 2, 3, 4, 5, 6]

5. Frequnes of character:

Aim:

To calculate the Frequency of the character in Java program.

Pseudo code:

1. Initialise the variable as Hashmap to store frequency of each character
2. iterate over each character in the string
3. If it is already in the map increment the count, If not, add it with a count of 1
4. print the contents of the 'Hashmap'

Program:

```
Import java.util.Hashmap;
Import java.util.map;
public class Characterfrequency {
    public static void main (string[] args) {
        string str = "hello";
    map < character, integer > charFrequency = Count Frequency (str);
    for (map.Entry < character, Integer > entry : charFreq.entrySet()){
    System.out.println (entry.getKey() + ":" + entry.getValue())}
    }
}
    for (Char c : str.toCharArray()) {
        if (frequency map.containsKey (c)) {
        frequency map.put (c, freq map.get(c)
    } else {
        frequency map.put (c, 1);
    }
}
    return frequency map;
}
```

output:

Hello

h : 1
e : 1
l : 2
o : 1