**JAVA WEEK-2**

**CHIRAN JEEVI**

**2019103013**

Q1) In our University, student email addresses end with @student.annauniv.edu, while professor email addresses end with @prof.annauniv.edu. Write a program that first asks the user how many email addresses they will be entering, and enter those user given addresses in an array of String. The program should print out a message indicating the number of student addresses and professor addresses.

CODE:

import java.util.Scanner;

public class OneP2 {

    public static *void* main(String[] *args*)

    {

        Scanner input =  new Scanner(System.in);

*int* i, j, n;

        String stud  = new String("@student.annauniv.edu");

        String prof  = new String("@prof.annauniv.edu");

*int* studNum=0, profNum=0;

        System.out.print("\n Enter Total no. of Email Addresses: ");

        n = input.nextInt();

        input.nextLine();

        String [] s = new String[n];

        System.out.println();

        for (i=0; i<n; i++)

        {

            System.out.print("Enter Email #"+(i+1)+": ");

            s[i] = input.nextLine();

*int* len = s[i].length();

            if (len > 18) {

                if ( prof.equals(s[i].substring(len-(prof.length()), len)) )

                    profNum++;

            }

            if (len > 20) {

                if ( stud.equals(s[i].substring(len-(stud.length()), len)) )

                    studNum++;

            }

        }

        System.out.println("Number of Student Email Addresses   : " + studNum);

        System.out.println("Number of Professor Email Addresses : " + profNum);

        System.out.println("Number of Personal Email Addresses  : "+(n-studNum-profNum));

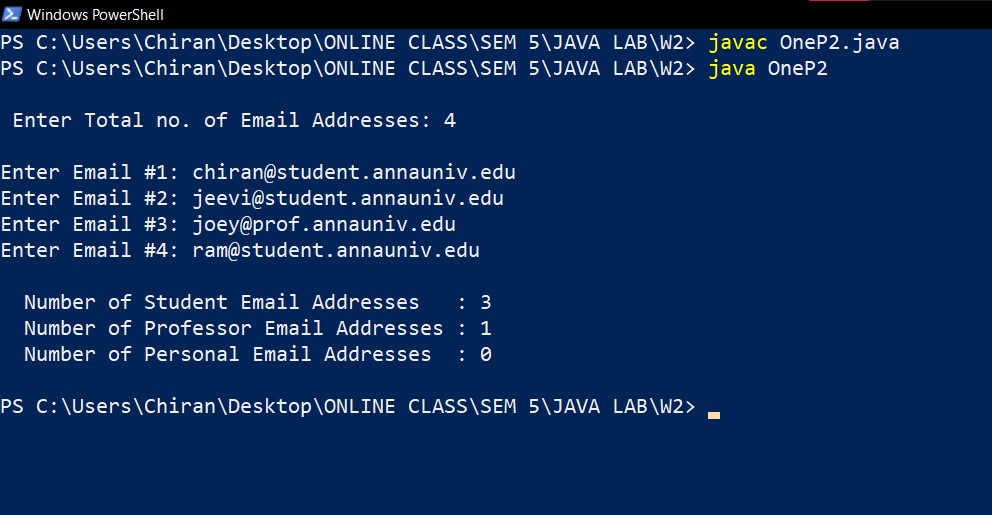
        System.out.println();

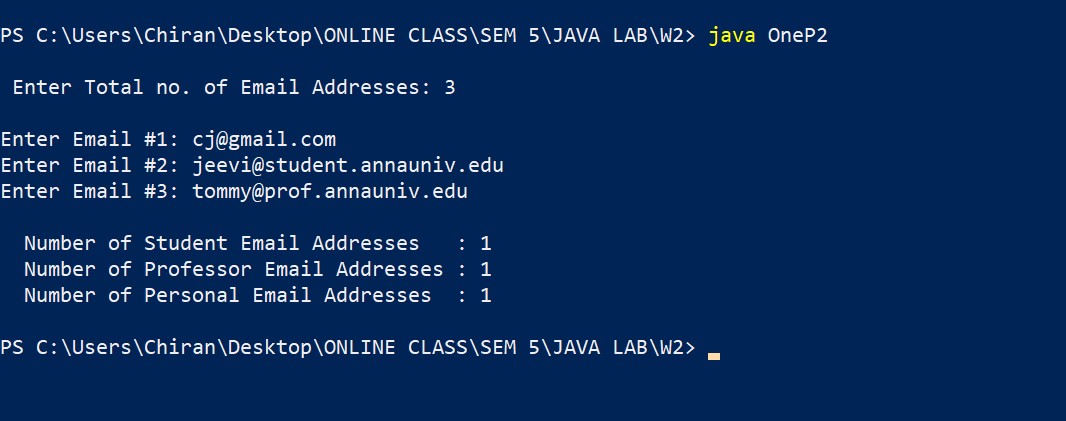
        input.close();

    }

}

OUTPUT:





2) Write a static method given as: **public static void palindromicPrime( int n).** The method has to print the palindromic prime within the range of integer specified by the user, n. Example: 131, 313,757 are palindromic primes

import java.util.Scanner;

public class Two

{

    public static *void* main(String[] *args*)

    {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter the Integer: ");

*int* n = input.nextInt();

        palindromicPrime(n);

    }

    public static *void* palindromicPrime( *int* *n*)

    {

        Scanner input = new Scanner(System.in);

*int* i, j, k=0;

*boolean* prime, palindrome;

*int* [] arr = new *int*[1000];

        arr[k] = 2;

*int* count=1;

        for (i=3; i<*n*; i++)

        {

            prime = true;

            for (j=2; j<i; j++)

            {

                if (i%j == 0) {

                    prime=false;

                    break;

                }

            }

            if (prime)

            {

                if (revv(i)) {

                    arr[++k] = i;

                    count++;

                }

            }

        }

        System.out.println("\n PALINDROME INTEGERS: \n");

        for (j=0; j<count; j++)

            System.out.println(" " + arr[j]);

    }

    public static *boolean* revv (*int* *num*)

    {

*int* result=0, reminder;

*int* original=*num*;

        while (*num* != 0) {

            reminder = *num*%10;

            result = result\*10 + reminder;

*num* = *num*/10;

        }

        if (result == original)

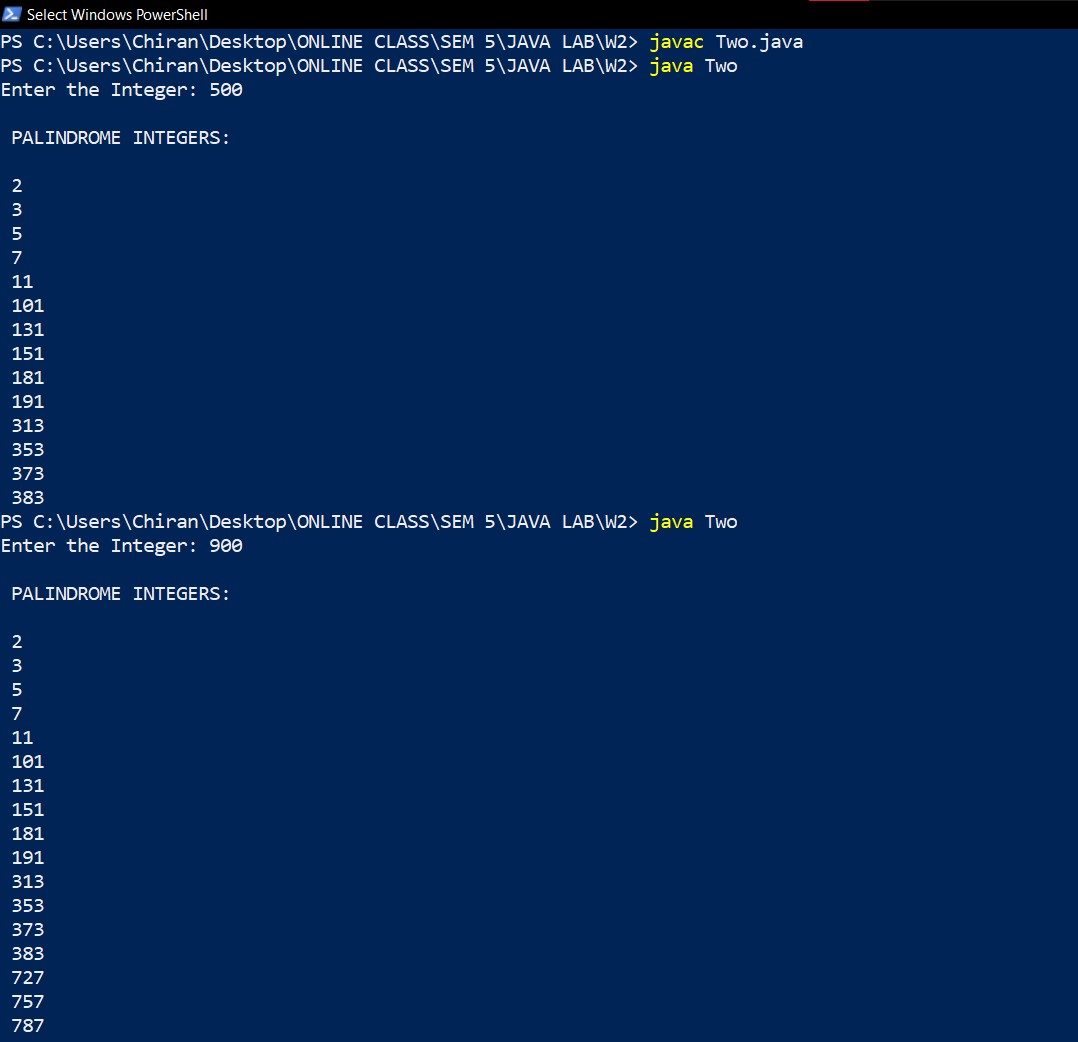
            return true;

        return false;

    }

}

OUTPUT:



3) Write a method equalTo to check whether the given two double values are equal and returns Boolean value

import java.util.Scanner;

public class Three {

    public static *void* main(String[] *args*)

    {

        Scanner input = new Scanner(System.in);

*double* num1, num2;

        System.out.print("\n Enter 1st Num: ");

        num1 = input.nextDouble();

        System.out.print(" Enter 2nd Num: ");

        num2 = input.nextDouble();

        if (equalTo(num1, num2))

            System.out.println("\t The 2 values are Equal!\n");

        else

            System.out.println("\t The 2 values are NOT Equal\n");

    }

    public static *boolean* equalTo (*double* *n1*, *double* *n2*) {

        if (*n1*==*n2*)

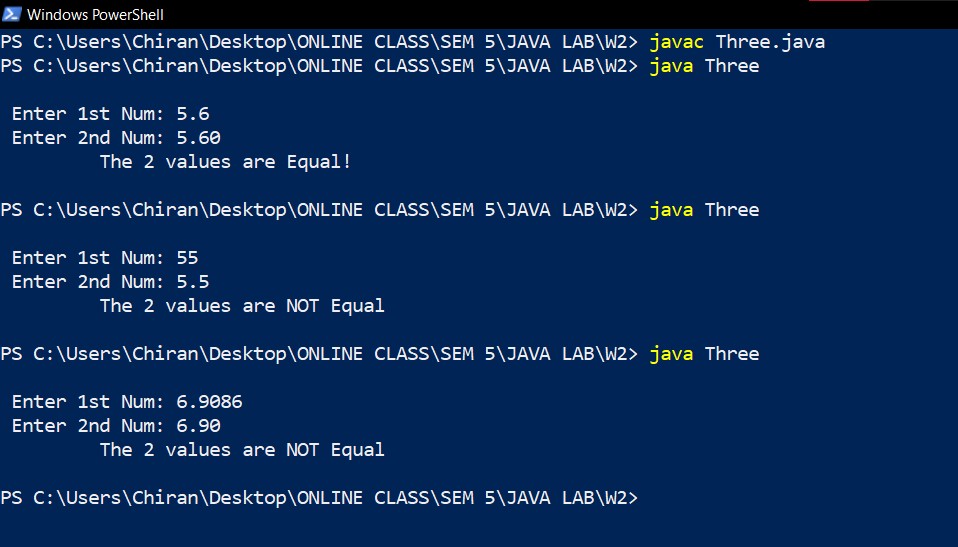
            return true;

        return false;

    }

}

OUTPUT:



**LEVEL 2**

Overload the equalTo method to find whether the given two strings are anagrams. Two words are anagrams if they contain the same letters in any order. For example, silent and listen are anagrams

import java.util.Scanner;

import java.util.Arrays;

public class Four

{

    public static *void* main(String[] *args*)

    {

        Scanner input = new Scanner(System.in);

*double* num1, num2;

        String s1 = new String();

        String s2 = new String();

*int* option;

        System.out.print("\n Enter Choice: (1)Double Check (2)Anagram Check : ");

        option = input.nextInt();

        if (option==1)

        {

            System.out.print("\n Enter 1st Num: ");

            num1 = input.nextDouble();

            System.out.print(" Enter 2nd Num: ");

            num2 = input.nextDouble();

            if (equalTo(num1, num2))

                System.out.println("\n\t The 2 values are Equal!\n");

            else

                System.out.println("\n\t The 2 values are NOT Equal\n");

        }

        else if (option==2)

        {

            input.nextLine();

            System.out.print("\n Enter 1st String: ");

            s1 = input.nextLine();

            System.out.print(" Enter 2nd String: ");

            s2 = input.nextLine();

            if ( equalTo(s1, s2) )

                System.out.println("\n\t YES THEY ARE ANAGRAMS!\n");

            else

                System.out.println("\n\t NO THEY ARENT ANAGRAMS\n");

        }

    }

    public static *boolean* equalTo (*double* *n1*, *double* *n2*)

    {

        if (*n1*==*n2*)

            return true;

        return false;

    }

    public static *boolean* equalTo (String *s1*, String *s2*)

    {

        if (*s1*.length()==*s2*.length())

        {

*int* i, j, len = *s1*.length();

*int* a1[] = new *int*[len];

*int* a2[] = new *int*[len];

*boolean* ans;

            for (i=0; i<len; i++) {

                a1[i] = *s1*.charAt(i);

                a2[i] = *s2*.charAt(i);

            }

            Arrays.sort(a1);

            Arrays.sort(a2);

            ans = Arrays.equals(a1, a2);

            return ans;

        }

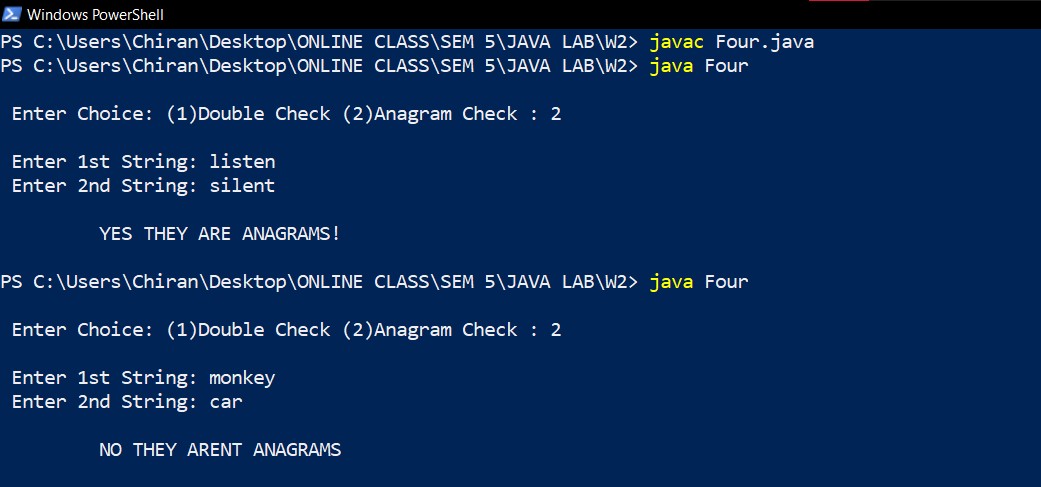
        else

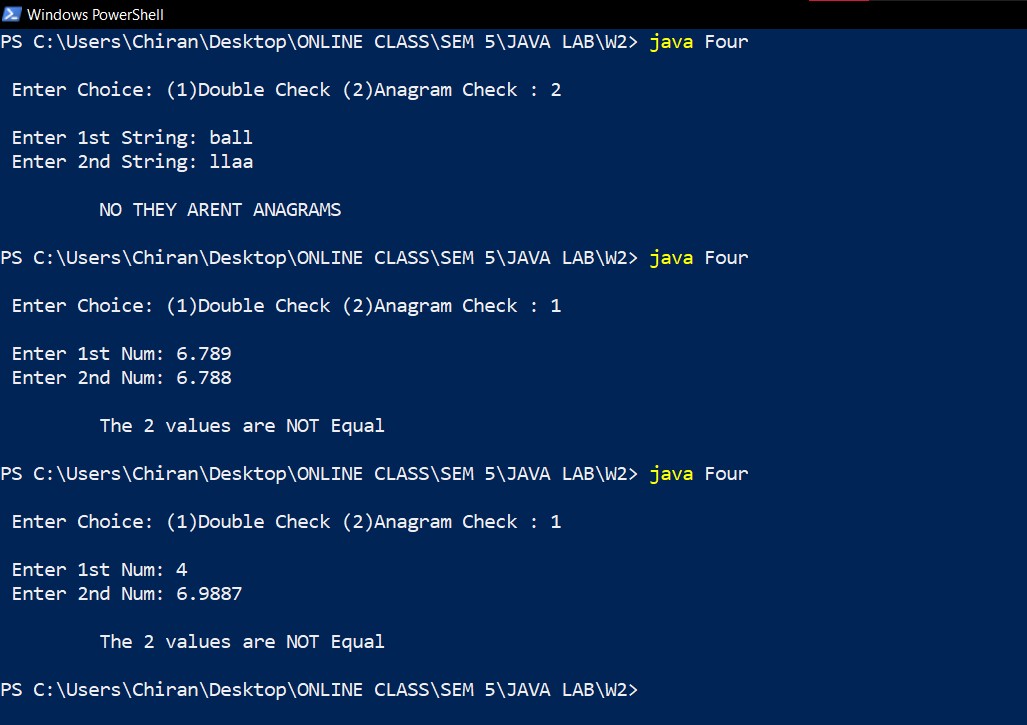
            return false;

    }

}

OUTPUT:





**LEVEL 3 – SPOT**

Define a static final String array FLAVOURS that consists of Ice cream flavours.

**static final String[] FLAVOURS** = {‘"Chocolate", "Strawberry", "Vanilla Fudge Swirl", "Mint Chip", "Mocha Almond Fudge", "Mango", "Praline Cream", "Lichi"}.

Define a method called flavourSet **public static String [] flavourSet (String flavor [])** that takes the array of flavours and returns the user preferred flavours as an array in the decreasing order of preference.

import java.util.Scanner;

import java.util.Arrays;

public class Five

{

    static final String[] flavours =

{

        "Chocolate", "Strawberry", "Vanilla Fudge Swirl", "Mint Chip",

        "Mocha Almond Fudge", "Mango", "Praline Cream", "Lichi"

    };

    public static *void* main(String[] *args*)

    {

*int* i, j, n = flavours.length;

        String result [] = new String[n];

        result = flavourSet(flavours);

        System.out.println("\n USER PREFERENCE: \n");

        for (i=0; i<n; i++)

        {

            System.out.println(result[i]);

        }

        System.out.println("\n");

    }

    public static String [] flavourSet (String *flavor* [])

    {

        Scanner input = new Scanner(System.in);

*int* i, j, n = flavours.length;

*int* loves[] = new *int*[n];

        String pref [] = new String[n];

        System.out.println("\n");

        for (i=0; i<n; i++) {

            System.out.print(" " + flavours[i] + " ");

        }

        System.out.println("\n\n"+"Enter Your Preferences (0-Most Favorite, 7-Least Favorite): \n");

        for (i=0; i<n; i++)

        {

            System.out.print(" " + flavours[i] + " Preference: ");

            loves[i] = input.nextInt();

        }

        for (j=0; j<n; j++)

            pref[loves[j]] = flavours[j];

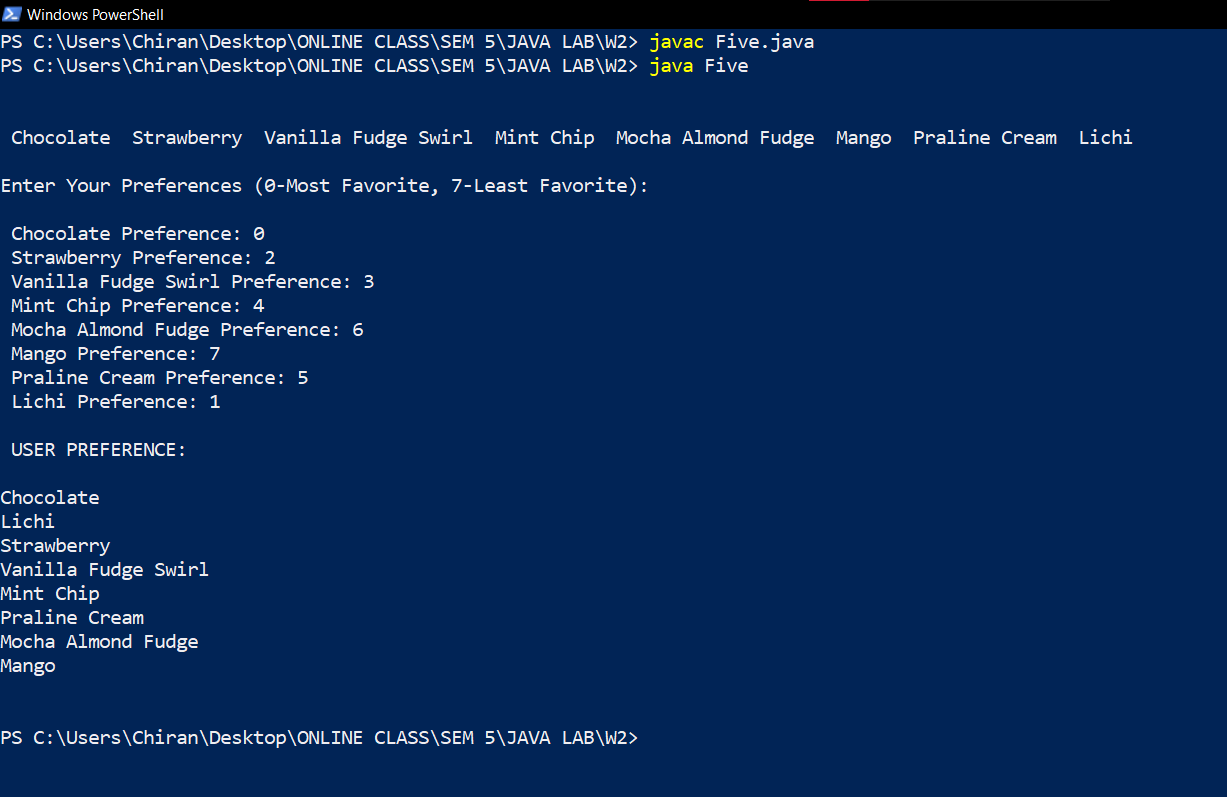
        input.close();

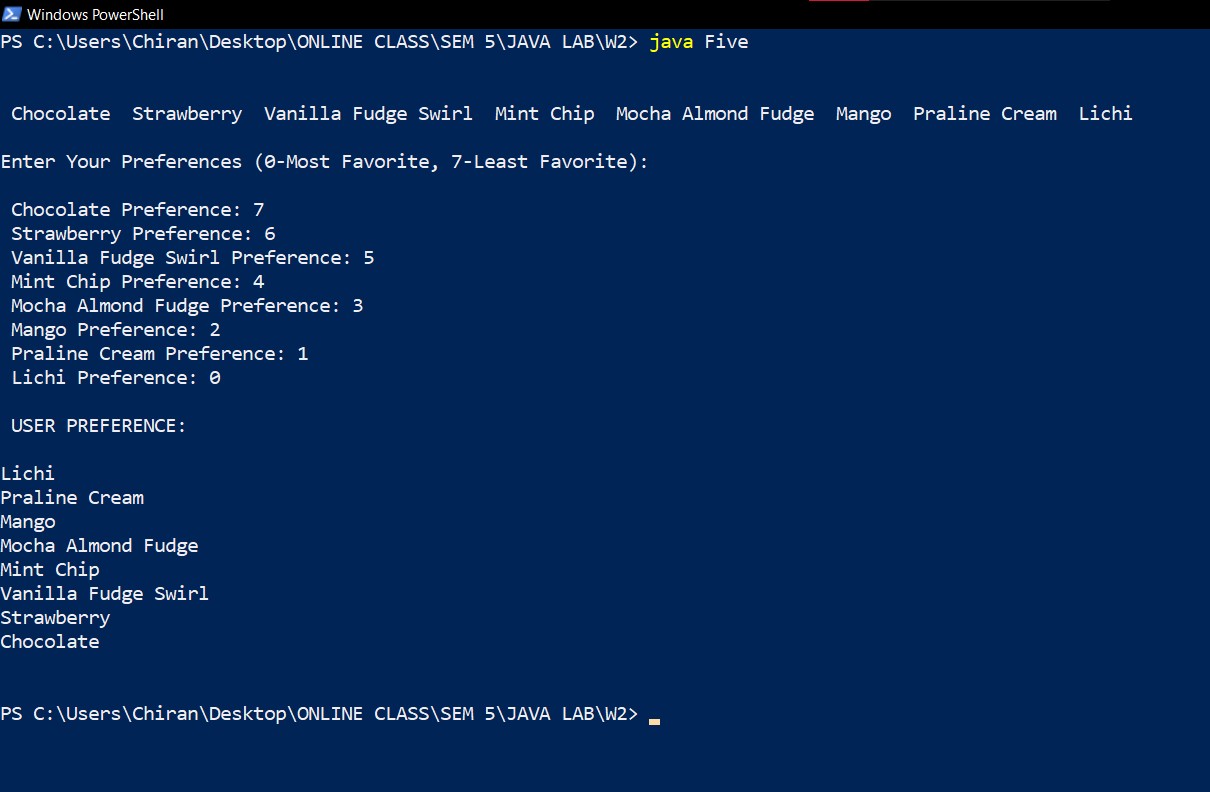
        return pref;

    }

}

OUTPUT:





|  |  |
| --- | --- |
|  | **Marks** |
| **Preparatory Exercises** |  |
| **Observation** |  |
| **Spot** |  |
| **Total** |  |