NAME: NITHISH KUMAR V R

ID: vrsathish15@gmail.com

Coding Challenges 2.1

1. **The farmer problem: A farmer is asking you to tell him how many legs can be counted among all his animals. The farmer breeds three spicies: chicken, cows and dogs. The farmer counter his animals and gives you the subtotal for each species, you have to write a method to find the total number of legs for the given animal list.**

**PROGRAM:**

import java.util.Scanner;

class Farmer

{

public static void main(String[] arg)

{

int Chicken,Cows,Dogs,total\_legs;

Scanner sc=new Scanner(System.in);

System.out.println("Enter Animals count");

System.out.print("chickens:");

Chicken=sc.nextInt();

System.out.print("cows:");

Cows=sc.nextInt();

System.out.print("Dogs:");

Dogs=sc.nextInt();

total\_legs=AnimalLegs(Chicken,Cows,Dogs);

System.out.println("Total number of legs for the given animal list : "+ total\_legs);

}

static int AnimalLegs(int x,int y,int z)

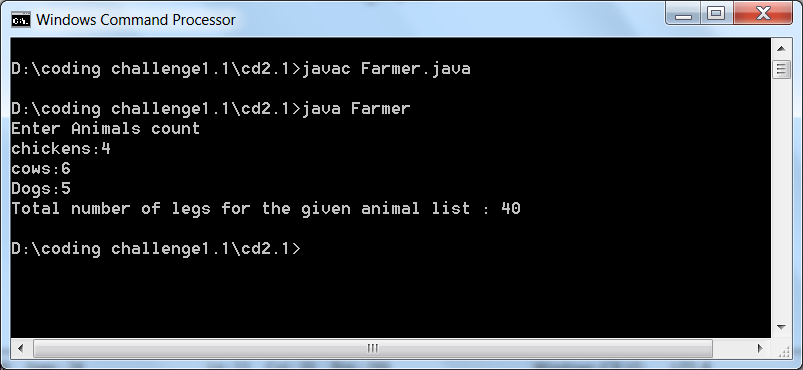
{

return (x\*2+y\*2+z\*4);

}

}

**OUTPUT:**

****

**5. Sam wants to select a username in order to register on a website. The rules for selecting a username are:**

**1. The minimum length of the username must be 5 characters and the maximum may be 10.**

**2. It should contain at least one letter from A-Z or a-z**

**3. It should contain at least one digit from 0-9**

**4. It should contain at least one character from amongst @#\*=**

**5. It should not contain any spaces**

**6. The first character should be an alphbet.**

**Write a program which accepts 4 usernames (one username per line) as input and checks whether each of them satisfy the above mentioned conditions. If a username satisfies the conditions, the program should print PASS (in uppercase) If a username fails the conditions, the program should print FAIL (in uppercase)**

**If suppose the following usernames are given to the program,**

**find the output:**

**1234@a**

**ABC3a#@**

**1Ac@**

**ABC 3a#@**

**PROGRAM:-**

import java.util.regex.\*;

import java.util.Scanner;

public class username {

public static void main(String[] args) {

String[] Array = new String[4];

Scanner in = new Scanner(System.in);

String pattern = "((?=.\*[0-9])(?=.\*[A-Z])(?=.\*[@#\*=])(?=[\\S]+$).{5,10})";

for(int i=0; i<Array.length;i++){

Array[i] = in.nextLine();

}

System.out.println("");

for(int i=0; i<Array.length;i++){

if(Array[i].matches(pattern)){

System.out.println("PASS");

}

else {

System.out.println("FAIL");

}

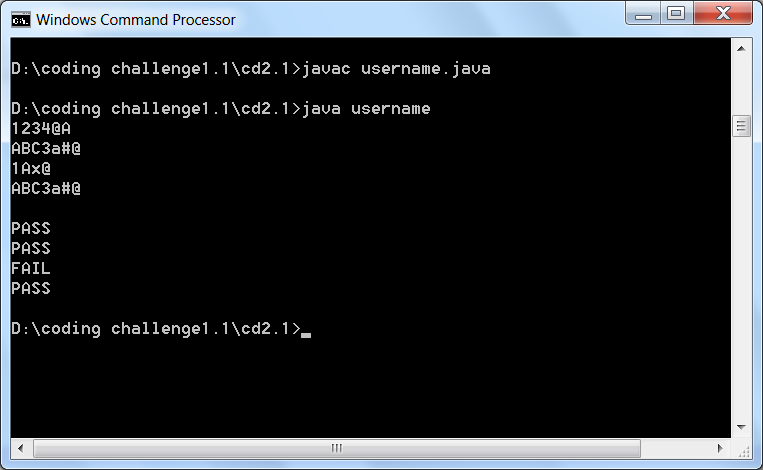
}

in.close();

}

}

OUTPUT:-



**6. Programming Challenges Question - 1**

**` Write a program that accepts 10 student records (roll number and score) and prints them in decreasing order of scores. In case there are multiple records pertaining to the same student, the program should choose a single record containing the highest score. The program should be capable of accepting a multi-line input. Each subsequent line of input will contain a student record, that is, a roll number and a score (separated by a hyphen). The output should consist of the combination of roll number and corresponding score in decreasing order of score. INPUT to program**

**1001-40**

**1002-50**

**1003-60**

**1002-80**

**1005-35**

**1005-55**

**1007-68**

**1009-99**

**1009-10**

**1004-89**

**PROGRAM:-**

import java.io.\*;

import java.util.\*;

public class rollnum{

public static void main(String[] args) {

String Data[]= new String[10];

int splitData[][]= new int[10][2];

InputStreamReader sr= new InputStreamReader(System.in);

BufferedReader br= new BufferedReader(sr) ;

try{

for(int i=0;i<10;i++)

{

Data[i]=br.readLine();

String str[]= Data[i].split("-");

splitData[i][0]=Integer.parseInt(str[1]);

splitData[i][1]=Integer.parseInt(str[0]);

}

Arrays.sort(splitData, new Comparator<int[]>() {

public int compare(int[] o1, int[] o2) {

return o2[0] - o1[0];

}

});

System.out.println("The Score");

Byte flag=0;

for(int i=0;i<10;i++)

{

flag=0;

for(int j=i-1;j>=0;j--)

{

if(splitData[i][1]==splitData[j][1])

{

flag=1;

break;

}

}

if(flag==1)

continue;

else

System.out.println(splitData[i][1]+"-"+splitData[i][ 0]);

}

}

catch(Exception ex)

{

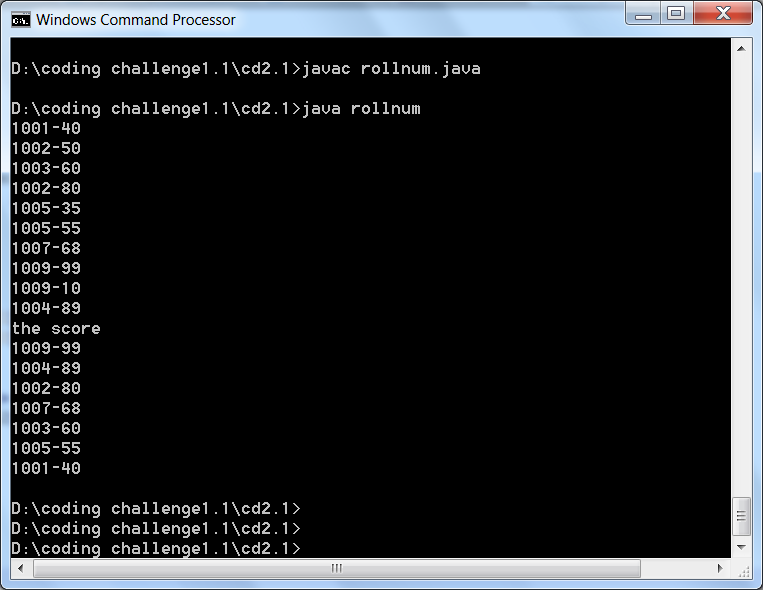
System.out.println(ex.toString());

}

}

}

**OUTPUT:-**



**7. Programming Challenges Question - 3**

**a frog hops in a particular way such that:  
  
1. He hops 20cm in the first hop, 10cm in the second hop and 5cm in the third hop.  
2. After three hops Kermit rests for a while and then again follows the same hopping pattern.  
  
Calculate the total distance travelled by Kermit (in centimeters) for the provided number of hops. Exactly 4 numbers of hops will be provided to the program (one number per line) as per the below example.  
  
Suppose the following number of hops is provided to the program:  
  
4  
6  
3  
5**

**PROGRAM:-**

import java.util.Scanner;

public class frog\_hops {

public static void main(String[] args) {

int a,b,c,d;

Scanner scan = new Scanner(System.in);

System.out.println("enter the inputs:");

a = scan.nextInt();

b = scan.nextInt();

c = scan.nextInt();

d = scan.nextInt();

int[] array = {a,b,c,d};

int k = 0;

for(int i=0; i<array.length; i++){

int n = array[i];

int x = n/3;

int y = n%3;

int z = x\*35;

switch(y){

case 0: k = 0 + z; break;

case 1: k = 20 + z; break;

case 2: k = 30 + z; break;

case 3: k = 35 + z; break;

}

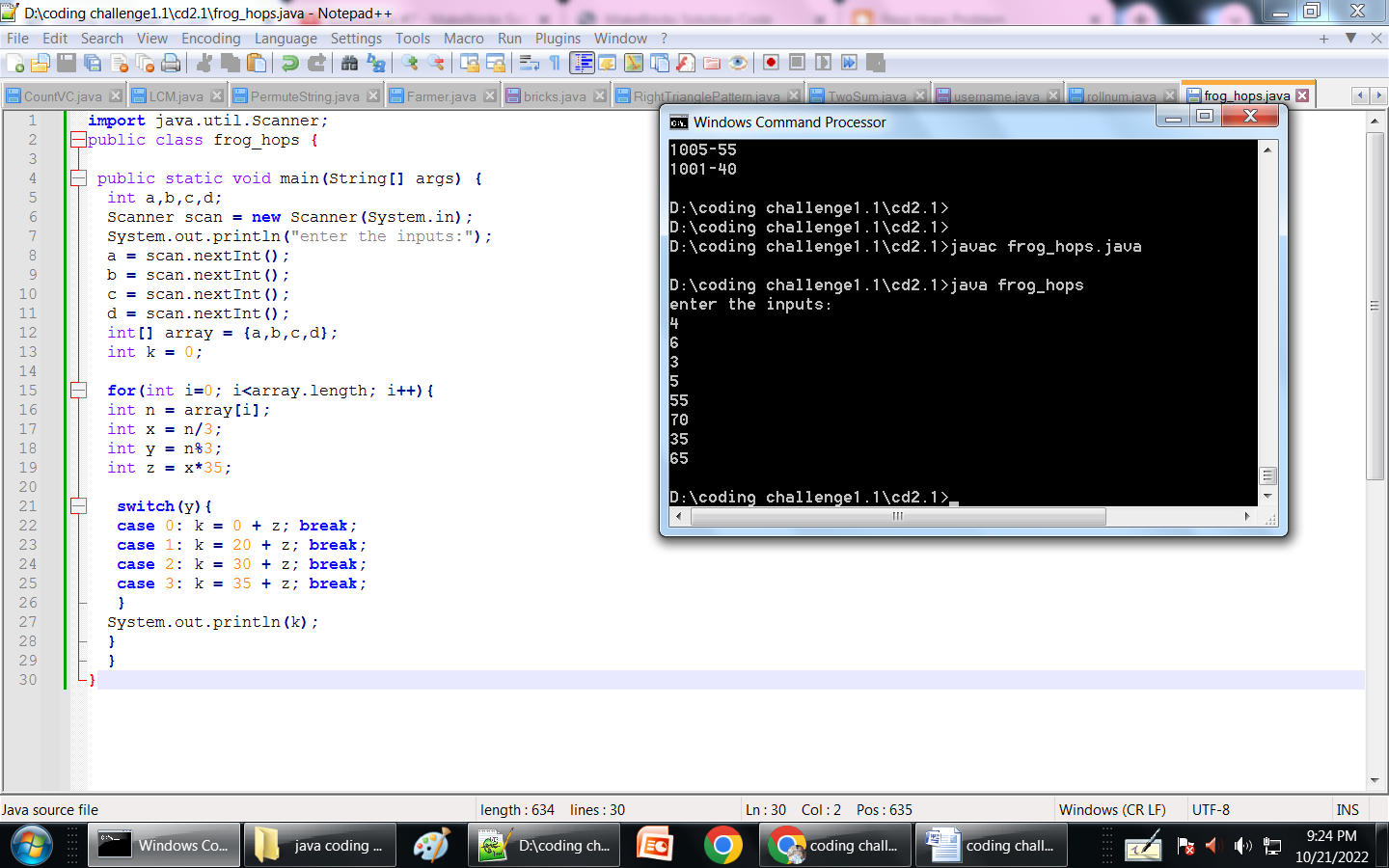
System.out.println(k);

}

}

}

**OUTPUT:-**

****

8. **Write a program which will take the year (yyyy) and the numeric sequence of the month (0-11) as its input.   
The program will return the day on which the 28th of that particular month and year falls. The input can consist of two year-month combinations, one combination per line.  
  
The numeric sequence of months is as follows:  
0 – Jan  
1 – Feb  
2 – March  
and so on......  
  
The format for supplying the input is:  
1999-5  
Where 1999 is the year and 5 is the numeric sequence of the month (corresponding to June). The program should display the day on which June 28, 1999 fell, and in this case the output will be MONDAY.   
  
The output should be displayed in uppercase letters.  
Suppose the following INPUT sequence is given to the program:  
1999-5  
1998-6  
Then the output should be:  
MONDAY  
TUESDAY**

**PROGRAM:-**

import java.io.\*;

import java.text.DateFormatSymbols;

import java.util.\*;

public class YEARTODAY {

public static void main(String[] args) {

try{

String data[]= new String[2];

InputStreamReader sr= new InputStreamReader(System.in);

BufferedReader br= new BufferedReader(sr);

for(int i=0;i<2;i++)

{

data[i]=br.readLine();

Calendar cal=Calendar.getInstance();

cal.set(Integer.parseInt(data[i].split("-")[0]), Integer.parseInt(data[i].split("-")[1]), 28);

String dayName = new DateFormatSymbols().getWeekdays()[cal.get(Calendar.DAY\_OF\_WEEK)];

System.out.println(dayName.toUpperCase());

}

}

catch(Exception ex)

{

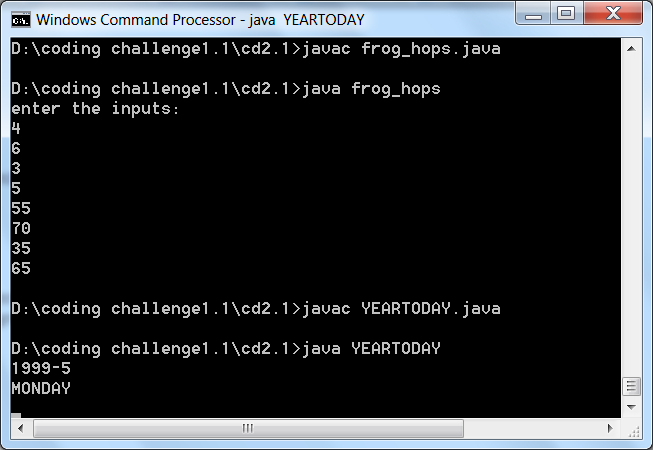
System.out.println(ex.toString());

}

}

}

**OUTPUT:-**

****

**10.** Write a program to find the GCD of the given numbers: 45, 270.

**PROGRAM:**

import java.util.Scanner;

public class GCD

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the two numbers: ");

int x = sc.nextInt();

int y = sc.nextInt();

System.out.println("The GCD of two numbers is: " + findGCD(x,y));

}

static int findGCD(int x, int y)

{

int r=0, a, b;

a = (x > y) ? x : y;

b = (x < y) ? x : y;

r = b;

while(a % b != 0)

{

r = a % b;

a = b;

b = r;

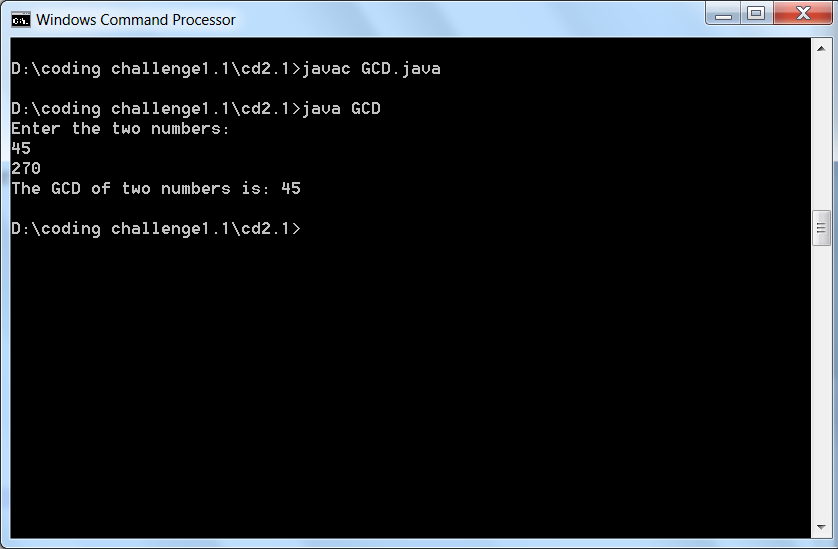
}

return r;

}

}

**OUTPUT:**

****