**ST REPORT**

**Name : Dev Prakash**

**USN : 1BM18IS147**

1. **Equivalence testing**

**Triangle problem**

#include<stdio.h>

#include<stdlib.h>

int main()

{ int a,b,c,c1,c2,c3;

char istriangle;

char choice='y';

do{ printf("\nEnter three sides of a triangle\n");

scanf("%d%d%d",&a,&b,&c);

c1=(a>=1&&a<=10);

c2=(b>=1&&b<=10);

c3=(c>=1&&c<=10);

if(!c1)

printf("\n The value of a=%d is not in the range of the permitted value",a);

if(!c2)

printf("\n The value of b=%d is not in the range of the permitted value",b);

if(!c3)

printf("\n The value of c=%d is not in the range of the permitted value",a);

if(a<b+c&&b<a+c&&c<a+b)

istriangle='y';

else

istriangle='n';

if(istriangle=='y')

{

if((a==b)&&(b==c))

printf("\n Equilateral triangle");

else if((a!=b)&&(a!=c)&&(b!=c))

printf("\n Scalene triangle");

else

printf("\n Isosceles triangle");

}

else

printf("\n Cannot form a triangle");

printf("\nEnter more test cases??(Y/N)");

scanf(" %c",&choice);

}while(choice=='y'||choice=='Y');

return 0;

}

**Test Data**: Enter the 3 Integer Value ( a , b And c )

**Pre-condition**: 1 ≤ a ≤ 10 , 1 ≤ b ≤ 10 and 1 ≤ c ≤ 10 and a < b + c , b < a + c and c < a + b

**Brief Description**: Check whether given value for a equilateral, isosceles , Scalene triangle or Not a triangle

**Equivalence classes**:

D1 = {<a, b, c>: a = b = c}

D2 = {<a, b, c>: a = b, a ≠ c}

D3 = {<a, b, c>: a = c, a ≠ b}

D4 = {<a, b, c>: b = c, a ≠ b}

D5 = {<a, b, c>: a ≠ b, a ≠ c, b ≠ c}

D6 = {<a, b, c>: a ≥ b + c}

D7 = {<a, b, c>: b ≥ a + c}

D8 = {<a, b, c>: c ≥ a + b}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Weak Normal Equivalence Class Testing** | | | | | | | |
| **Case id** | **Description** | **Input data** | | | **Expected output** | **Actual output** | **Status** |
| a | b | c |
| 1 | Enter value for a , b and c | 5 | 5 | 5 | Equilateral triangle | Equilateral triangle | Pass |
| 2 | Enter value for a , b and c | 2 | 2 | 3 | Isosceles triangle | Isosceles triangle | Pass |
| 3 | Enter value for a , b and c | 3 | 4 | 5 | Scalene triangle | Scalene triangle | Pass |
| 4 | Enter value for a , b and c | 4 | 1 | 2 | Not a triangle | Not a triangle | Pass |
| **Weak Robust Equivalence Class Testing** | | | | | | | |
| 5 | Enter one invalid input and two valid value for a, b and c | -1 | 5 | 5 | Value of a is not in the range of permitted values | Value of a is not in the range of permitted values | Pass |
| 6 | Enter one invalid input and two valid value for a, b and c | 5 | -1 | 5 | Value of b is not in the range of permitted values | Value of b is not in the range of permitted values | Pass |
| 7 | Enter one invalid input and two valid value for a, b and c | 5 | 5 | -1 | Value of c is not in the range of permitted values | Value of c is not in the range of permitted values | Pass |
| 8 | Enter one invalid input and two valid value for a, b and c | 11 | 5 | 5 | Value of a is not in the range of permitted values | Value of a is not in the range of permitted values | Pass |
| 9 | Enter one invalid input and two valid value for a, b and c | 5 | 11 | 5 | Value of b is not in the range of permitted values | Value of b is not in the range of permitted values | Pass |
| 10 | Enter one invalid input and two valid value for a, b and c | 5 | 5 | 11 | Value of c is not in the range of permitted values | Value of c is not in the range of permitted values | Pass |
| **Strong Robust Equivalence Class Testing** | | | | | | | |
| 11 | Enter one invalid input and two valid value for a , b and c | -1 | 5 | 5 | Value of a is not in the range of permitted values | Value of a is not in the range of permitted values | Pass |
| 12 | Enter one invalid input and two valid value for a , b and c | 5 | -1 | 5 | Value of b is not in the range of permitted values | Value of b is not in the range of permitted values | Pass |
| 13 | Enter one invalid input and two valid value for a , b and c | 5 | 5 | -1 | Value of c is not in the range of permitted values | Value of c is not in the range of permitted values | Pass |
| 14 | Enter two invalid input and one valid value for a , b and c | -1 | -1 | 5 | Value of a,b is not in the range of permitted values | Value of a,b is not in the range of permitted values | Pass |
| 15 | Enter two invalid input and one valid value for a , b and c | 5 | -1 | -1 | Value of b,c is not in the range of permitted values | Value of b,c is not in the range of permitted values | Pass |
| 16 | Enter two invalid input and one valid value for a , b and c | -1 | 5 | -1 | Value of a,c is not in the range of permitted values | Value of a,c is not in the range of permitted values | Pass |
| 17 | Enter all invalid inputs | -1 | -1 | -1 | Value of a,b,c is not in the range of permitted values | Value of a,b,c is not in the range of permitted values | Pass |

**Next date problem**

#include<stdio.h>

int check(int day,int month)

{

if((month==4||month==6||month==9 ||month==11) && day==31)

return 1;

else

return 0;

}

int isleap(int year)

{

if((year%4==0 && year%100!=0) || year%400==0)

return 1;

else

return 0;

}

int main()

{

int day,month,year,tomm\_day,tomm\_month,tomm\_year;

char flag;

do

{

flag='y';

printf("\nenter the today's date in the form of dd mm yyyy\n");

scanf("%d%d%d",&day,&month,&year);

tomm\_month=month;

tomm\_year= year;

if(day<1 || day>31)

{

printf("value of day, not in the range 1...31\n");

flag='n';

}

if(month<1 || month>12)

{

printf("value of month, not in the range 1....12\n");

flag='n';

}

else if(check(day,month))

{

printf("value of day, not in the range day<=30");

flag='n';

}

if(year<=1812 || year>2013)

{

printf("value of year, not in the range 1812.......2013\n");

flag='n';

}

if(month==2)

{

if(isleap(year) && day>29)

{

printf("invalid date input for leap year");

flag='n';

}

else if(!(isleap(year))&& day>28)

{

printf("invalid date input for not a leap year");

flag='n';

}

}

}

while(flag=='n');

switch (month)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:if(day<31)

tomm\_day=day+1;

else

{

tomm\_day=1;

tomm\_month=month+1;

}

break;

case 4:

case 6:

case 9:

case 11: if(day<30)

tomm\_day=day+1;

else

{

tomm\_day=1;

tomm\_month=month+1;

}

break;

case 12: if(day<31)

tomm\_day=day+1;

else

{

tomm\_day=1;

tomm\_month=1;

if(year==2013)

{

printf("the next day is out of boundary value of year\n");

tomm\_year=year+1;

}

else

tomm\_year=year+1;

}

break;

case 2:

if(day<28)

tomm\_day=day+1;

else if(isleap(year)&& day==28)

tomm\_day=day+1;

else if(day==28 || day==29)

{

tomm\_day=1;

tomm\_month=3;

}

break;

}

printf("next day is : %d %d %d",tomm\_day,tomm\_month,tomm\_year);

return 0;

}

**2. Boundary value testing:**

**Commision problem**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online Java Compiler.

Code, Compile, Run and Debug java program online.

Write your code in this editor and press "Run" button to execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.Scanner;

public class Main

{

public static void main(String arg[])

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter locks,stocks and barrel qty:");

int locks = sc.nextInt();

int stocks = sc.nextInt();

int barrels = sc.nextInt();

int lockPrice = 45;

int stockPrice = 30;

int barrelPrice = 25;

if(locks<=70&&stocks<=80&&barrels<=90){

double amount=(locks\*lockPrice)+(stockPrice\*stocks)+(barrelPrice\*barrels);

double commission = 0.0;

//System.out.print("Enter commissionPercentage:");

//double commissionPercentage=sc.nextDouble();

if(amount > 1800){

commission = 0.10\*1000.0;

commission+= 0.15 \* 800.0;

commission+= 0.20 \* (amount - 1800.0);

}

else if(amount > 1000.0){

commission = 0.10\* 1000.0;

commission += 0.15\*(amount-1000.0);

}

else{

commission = 0.10 \* amount;

}

System.out.println("commission:"+commission);

}

else{

System.out.println("Request qty exceeds the available qty.");

}

}

}

**Test cases:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Input data | | | Expected output | | Actual output | | | |
| Description | Total locks | Total stocks | Total barrels | Sales | Commission | Sales | Commission | Status | Comment |
| Enter random values to calculate commission nearly less than 1000 | 11 | 10 | 8 | 995 | 99.5 | 995 | 99.5 | PASS | Border point |
| Enter random values to calculate commission for sales greater than 1000 | 10 | 11 | 9 | 1005 | 100.75 | 1005 | 100.75 | PASS | Border point+ |
| Enter random values to calculate commission for sales nearly less than 1800 | 18 | 17 | 19 | 1795 | 219.25 | 1795 | 219.25 | PASS | Border point- |
| Enter random values to calculate commission for sales nearly greater than 1800 | 18 | 19 | 17 | 1805 | 221 | 1805 | 221 | PASS | Border point+ |

**Triangle problem**

#include<stdio.h>

#include<stdlib.h>

int main()

{

int a,b,c,c1,c2,c3;

char istriangle;

char choice='y';

do{

printf("\nEnter three sides of a triangle\n");

scanf("%d%d%d",&a,&b,&c);

c1=(a>=1&&a<=10);

c2=(b>=1&&b<=10);

c3=(c>=1&&c<=10);

if(!c1)

{ printf("\n The value of a=%d is not in the range of the permitted value",a);

exit(0);}

if(!c2)

{ printf("\n The value of b=%d is not in the range of the permitted value",b);

exit(0);}

if(!c3)

{ printf("\n The value of c=%d is not in the range of the permitted value",a);

exit(0);}

if(a<b+c&&b<a+c&&c<a+b)

istriangle='y';

else

istriangle='n';

if(istriangle=='y')

{

if((a==b)&&(b==c))

{

printf("\n Equilateral triangle");

}

else if((a!=b)&&(a!=c)&&(b!=c))

{

printf("\n Scalene triangle");

}

else

{

printf("\n Isosceles triangle");

}

}

else

{

printf("\n Cannot form a triangle");

}

printf("\nEnter more test cases??(Y/N)");

scanf(" %c",&choice);

}while(choice=='y'||choice=='Y');

return 0;

}

**Test Data**: Enter the 3 Integer Value (a, b and c)

**Pre-condition**: 1 ≤ a ≤ 10, 1 ≤ b ≤ 10 and 1 ≤ c ≤ 10 and a < b + c, b < a + c and c < a + b

**Brief Description**: Check whether given value for an equilateral, isosceles, Scalene triangle or can't from a triangle

Test Cases:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Case id** | **Description** | **Input data** | | | **Expected output** | **Actual output** |
| a | b | c |
| 1 | Enter the min value for a, b and c | 1 | 1 | 1 | Equilateral triangle | Equilateral triangle |
| 2 | Enter min value for 2 items and min+1 for any one item | 1 | 1 | 2 | Can’t form a triangle | Can’t form a triangle |
| 3 | Enter min value for 2 items and min+1 for any one item | 1 | 2 | 1 | Can’t form a triangle | Can’t form a triangle |
| 4 | Enter min value for 2 items and min+1 for any one item | 2 | 1 | 1 | Can’t form a triangle | Can’t form a triangle |
| 5 | Enter normal value for 2 items and 1 item is min value | 5 | 5 | 1 | Isosceles triangle | Isosceles triangle |
| 6 | Enter normal value for 2 items and 1 item is min value | 5 | 1 | 5 | Isosceles triangle | Isosceles triangle |
| 7 | Enter normal value for 2 items and 1 item is min value | 1 | 5 | 5 | Isosceles triangle | Isosceles triangle |
| 8 | Enter the normal value for a, b, and c | 5 | 5 | 5 | Equilateral triangle | Equilateral triangle |
| 9 | Enter the normal value for 2 items and 1 item is max value | 5 | 5 | 10 | Can’t form a triangle | Can’t form a triangle |
| 10 | Enter the normal value for 2 items and 1 item is max value | 5 | 10 | 5 | Can’t form a triangle | Can’t form a triangle |
| 11 | Enter the normal value for 2 items and 1 item is max value | 10 | 5 | 5 | Can’t form a triangle | Can’t form a triangle |
| 12 | Enter the max value for 2 items and max-1 for any one item | 10 | 10 | 9 | Isosceles triangle | Isosceles triangle |
| 13 | Enter the max value for 2 items and max-1 for any one item | 10 | 9 | 10 | Isosceles triangle | Isosceles triangle |
| 14 | Enter the max value for 2 items and max-1 for any one item | 9 | 10 | 10 | Isosceles triangle | Isosceles triangle |
| 15 | Enter max value for a, b and c | 10 | 10 | 10 | Equilateral triangle | Equilateral triangle |

**3. Data flow**

**Data flow testing for all variables (mean, median, sd, variance)**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online Java Compiler.

Code, Compile, Run and Debug java program online.

Write your code in this editor and press "Run" button to execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

public class Main

{

// Function for calculating mean

public static double findMedian(double a[], int n)

{

Arrays.sort(a);

// check for even case

if (n % 2 != 0)

return (double)a[n / 2];

return (double)(a[(n - 1) / 2] + a[n / 2]) / 2.0;

}

public static double findMean(double a[], double n)

{

int sum = 0;

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

sum += a[i];

return (double)sum / (double)n;

}

public static double findVariance(double a[], int n)

{

double variance = 0.0;

for(int i=0;i<n;++i){

variance+= ((a[i]- findMean(a,n))\*(a[i]- findMean(a,n)));

}

variance = variance/(n-1);

return variance;

}

public static double standardDeviation(double var)

{

return Math.sqrt(var);

}

// Driver code

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

//double a[n];

double[] a = new double[n];

for(int i =0;i<n;++i){

a[i] = sc.nextDouble();

}

// Function call

System.out.println("Mean = " + findMean(a, n));

System.out.println("Median = " + findMedian(a, n));

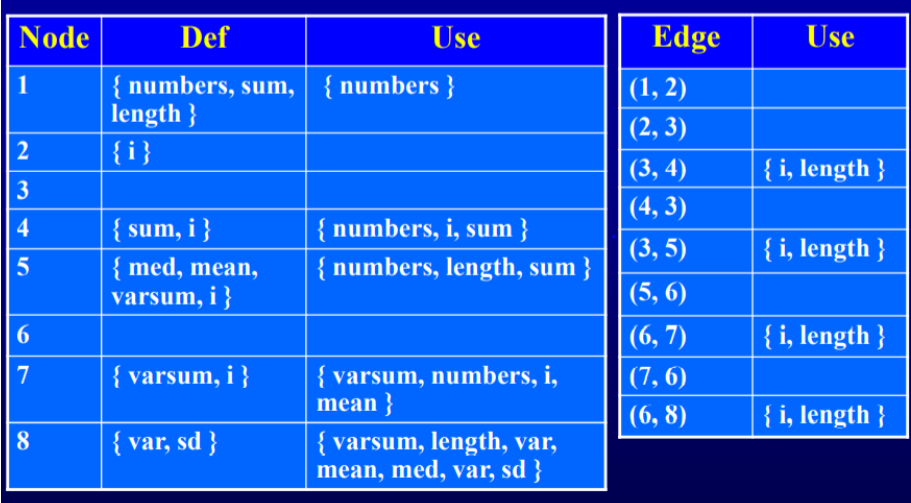
System.out.println("Variance=" + findVariance(a,n));

System.out.println("Std deviation="+standardDeviation(findVariance(a,n)));

}

}

Test cases:



**Data flow testing for cellular service problem**

#include<iostream>

#include<stdlib.h>

using namespace std;

double calculateBill(int usage)

{ double bill=0; //1

if(usage>0) //2

{ bill=40; //3

}

if(usage>100) //4

{ if(usage<=200) //5

{ bill+=(usage-100)\*0.5;//9

}

else

{ bill+=50+(usage-200)\*0.1;//6

if(bill>=100)//7

{ bill=bill\*0.9; //8

} } }

return bill; //1 }

int main()

{ int usage;

while(true)

{ cout<<"\n Enter usage(Enter -1 to exit): ";

cin>>usage;

if(usage==-1)

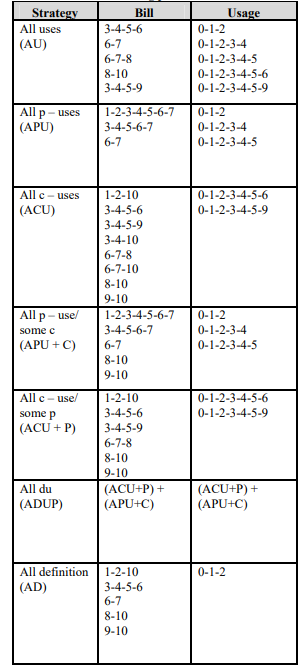
exit(0);

cout<<"\nThe total bill is: "<<calculateBill(usage); //0

}//1

return 0; }

Test cases:



**4.Object Oriented**

**Currency converter**

**package** testing;

**import** javax.swing.\*;

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** Convertor {

**public** **static** String *country*="";

**public** JFrame f;

**public** JLabel l1;

**public** JTextArea t1;

**public** **void** convertor()

{

f=**new** JFrame("Currency Convertor");

l1= **new** JLabel("US Dollar Amount");

l1.setBounds(50,50,300,50);

l1.setFont(**new** Font("Helvetica Neue", Font.*BOLD*, 18));

t1=**new** JTextArea();

t1.setBounds(100,100,200,30);

t1.setFont(**new** Font("Helvetica Neue", Font.*BOLD*, 18));

JRadioButton b1=**new** JRadioButton("Brazil");

b1.setBounds(50,130,100,80);

JRadioButton b2=**new** JRadioButton("Canada");

b2.setBounds(50,180,100,80);

JRadioButton b3=**new** JRadioButton("European Community");

b3.setBounds(50,230,250,80);

JRadioButton b4=**new** JRadioButton("Japan");

b4.setBounds(50,280,100,80);

b1.addActionListener(actionListener);

b2.addActionListener(actionListener);

b3.addActionListener(actionListener);

b4.addActionListener(actionListener);

JButton but1=**new** JButton("Convert");

but1.setBounds(230,500,100,40);

JButton but2=**new** JButton("Clear");

but2.setBounds(100,500,100,40);

JButton but3=**new** JButton("Quit");

but3.setBounds(370,500,100,40);

but1.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e)

{

// Converting to double

**double** d1 = Double.*parseDouble*(t1.getText());

JLabel t2=**new** JLabel("Equivalent In ");

t2.setBounds(50,350,300,50);

t2.setFont(**new** Font("Serif", Font.*BOLD*, 20));

f.add(t2);

String[] currency= {"Brazilian real","Canadian Dollar","Euro","Yen"};

**int** index=4;

Double curr=0.0;

**if**(*country*.equals("Canada"))

{

index=1;

curr=1.21\*d1;

}

**else** **if**(*country*.equals("Brazil"))

{

index=0;

curr=5.22\*d1;

}

**else** **if**(*country*.equals("Japan"))

{

index=3;

curr=108.85\*d1;

}

**else**

{

index=2;

curr=0.82\*d1;

}

JLabel t3=**new** JLabel(currency[index]);

t3.setBounds(175,350,250,50);

t3.setFont(**new** Font("Helvetica Neue", Font.*BOLD*, 20));

f.add(t3);

JTextArea t4=**new** JTextArea(curr.toString());

t4.setBounds(100,420,200,30);

t4.setEditable(**false**);

t4.setFont(**new** Font("Helvetica Neue", Font.*BOLD*, 20));

f.add(t4);

f.validate();

f.repaint();

}

});

but2.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

f.dispose();

**new** Convertor().convertor();

}

});

but3.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

System.*exit*(0);

}

});

f.add(l1);

f.add(t1);

f.add(b1);

f.add(b2);

f.add(b3);

f.add(b4);

f.add(but1);

f.add(but2);

f.add(but3);

f.setSize(600,700);

f.setLayout(**null**);

f.setVisible(**true**);

}

ActionListener actionListener = **new** ActionListener() {

**public** **void** actionPerformed(ActionEvent actionEvent) {

JRadioButton button = (JRadioButton)actionEvent.getSource();

*country*=button.getText();

}

};

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

{

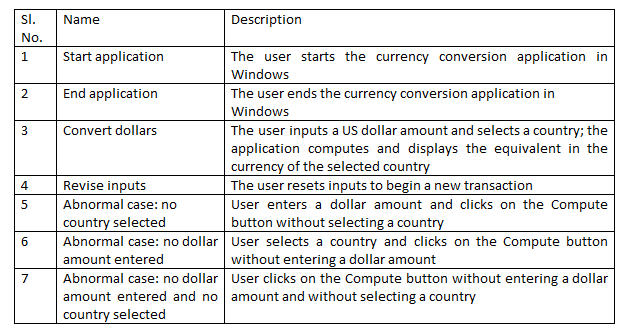
Convertor obj=**new** Convertor();

obj.convertor();

}

}

Test Cases:



**Next date calculator:**

import java.awt.event.\*;

import java.util.\*;

public abstract class CalendarProgram{

int currentPos;

CalendarProgram(){}

CalendarProgram(int pCurrentPos){

currentPos = pCurrentPos;

}

public void setCurrentPos(int pCurrentPos) {

currentPos = pCurrentPos;

}

abstract boolean increment();

public static void main(String[] args) {

Date date = new Date(12,31,1995);

date.increment();

date.printDate();

}

}

class Year extends CalendarProgram{

private int year;

Year(int year){

super();

setCurrentPos(year);

}

public int getYear() {

return currentPos;

}

public boolean increment() {

currentPos+=1;

return true;

}

public boolean isLeap() {

if(((currentPos%4==0) &&!(currentPos%100==0))||(currentPos%400==0)){

return true;

}

else {

return false;

}

}

}

class Month extends CalendarProgram{

private Year y;

private int cMonth;

private int[] stringIndex = {31,28,31,30,31,30,31,31,30,31,30,31};

Month(int pCur,Year y){

super();

setMonth(pCur,y);

}

public void setMonth(int pcur,Year pyear) {

setCurrentPos(pcur);

y = pyear;

}

public int getMonth() {

return currentPos;

}

public int getMonthSize() {

if(y.isLeap()) {

stringIndex[1]=28;

}

else {

stringIndex[1]=29;

}

return stringIndex[currentPos-1];

}

public boolean increment() {

currentPos+=1;

if(currentPos>12) {

return false;

}

else {

return true;

}

}

}

class Day extends CalendarProgram{

private Month month;

Day(int pDay,Month pMonth){

setDay(pDay,pMonth);

}

public void setDay(int pday,Month mMonth)

{

setCurrentPos(pday);

//currentPos = pday;

month = mMonth;

}

public int getDay() {

return currentPos;

}

public boolean increment() {

System.out.println("Increment day");

currentPos+=1;

if(currentPos<=month.getMonthSize()) {

return true;

}

else {

return false;

}

}

}

class Date {

private Day d;

private Month m;

private Year y;

Date(int pMonth,int pDay,int pYear){

super();

this.y = new Year(pYear);

this.m = new Month(pMonth,this.y);

this.d = new Day(pDay,this.m);

}

public void increment() {

if(!d.increment()) {

if(!m.increment()) {

y.increment();

System.out.println("Called date increment");

this.m.setMonth(1, y);

this.d.setDay(1,this.m);

}

else {

this.m.setMonth(m.getMonth(), y);

this.d.setDay(1,this.m);

}

}

else {

System.out.println("Called date increment else");

//d.increment();

this.d.setDay(d.getDay(), m);

}

}

public void printDate() {

System.out.println(m.getMonth()+"/"+d.getDay()+"/"+y.getYear());

}

}

TEST CASES:

Entered date: 29-02-1995

Next date(actual output) : 1-03-1995

Expected output: 1-03-1995

**5.Selenium**

**Pgm 1**

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class First {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "E:\\college\\ST\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.navigate().to("https://google.com");

driver.findElement(By.name("q")).sendKeys("Bengaluru to Manali");

driver.findElement(By.name("q")).click();

driver.findElement(By.name("q")).sendKeys(Keys.ENTER);

}

}