

/\* Problem 1: Write a java program to take input as a command line argument. Your name, course, university\_rollno and semester.

Display the information as.

Name:

UniversityRollNo:

Course:

Semester:\*/

```
public class Main
```

```
{ public static void main(String[] args) {
```

```
String name,course;
```

```
int rollno,sem;
```

```
System.out.println("Name: "+args[0]);
```

```
System.out.println("University RollNo"+args[1]);
```

```
System.out.println("Course: "+args[2]);
```

```
System.out.println("Semester: "+args[3]);
```

```
}
```

OUTPUT:

Name: shradhay dham

University Rollno: 2017033

Course: btech

Semester: 4

---

Name: john karlie

University Rollno: 2055434

Course: bba

Semester: 4

---

Name: tommy joplr

University Rollno: 2065437

Course: mca

Semester: 4

/\* Problem 2: Using the switch statement, write a menu-driven program to calculate the maturity amount of a bank deposit.

The user is given the following options: (i) Term Deposit (ii) Recurring Deposit For option (i) accept Principal (p), rate of interest (r)

and time period in years (n). Calculate and output the maturity amount (a) receivable using the formula  $a = p[1 + r / 100]n$ .

For option (ii) accept monthly installment (p), rate of interest (r) and time period in months (n).

Calculate and output the maturity amount (a) receivable using the formula  $a = p * n + p * n(n + 1) / 2 * r / 100 * 1 / 12$ .

For an incorrect option, an appropriate error message should be displayed. [ Use Scanner Class to take input ] \*/

```
import java.util.Scanner;

public class Main
{ void termdeposit()
{ int p,r,t;
Scanner ss=new Scanner(System.in);
System.out.println(" Principal: ");
p=Integer.parseInt(ss.nextLine());
System.out.println(" Rate of Interest: ");
r=Integer.parseInt(ss.nextLine());
System.out.println(" Time in months: ");
t=Integer.parseInt(ss.nextLine());
System.out.print("Output: "+ p*(1+(r/100))*t);
}

void recurringdeposit()
{ int p,r,t;
Scanner ss=new Scanner(System.in);
System.out.println(" Monthly Installment: ");
p=Integer.parseInt(ss.nextLine());
```

```
System.out.println(" Rate of Interest: ");
r=Integer.parseInt(ss.nextLine());
System.out.println(" Time in Years: ");
t=Integer.parseInt(ss.nextLine());
float a=((p*t)+p*(t*(t+1)/2)*(r/100)*(1/12));
System.out.println("Output: "+a);
}

public static void main(String[] args) {
System.out.print(" 1: Term Deposit 2: Recurring Deposit: ");
Main m=new Main();
Scanner obj=new Scanner(System.in);
int ch=Integer.parseInt(obj.nextLine());
switch(ch) {
case 1:
    m.termdeposit();
    break;
case 2:
    m.recurringdeposit();
    break;
default:
    System.out.print("Invalid Choice!!");
}}}
```

OUTPUT:

1: Term Deposit 2: Recurring Deposit: 1

Principal:4587

Rate of Interest:5

Time in months:12

Output:55044

---

1: Term Deposit 2: Recurring Deposit: 2

Principal:500

Rate of Interest:6

Time in months:5

Output:2500.0

---

1: Term Deposit 2: Recurring Deposit: 3

Invalid Choice!!

/\*Problem 3: Write a Program to find if the given numbers are Friendly pair or not (Amicable or not).

Friendly Pair are two or more numbers with a common abundance. \*/

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args) {

        Scanner obj=new Scanner(System.in);

        int n1=obj.nextInt();

        int n2=obj.nextInt();

        int result1=0,result2=0;

        for(int i=1;i<n1;i++)

        { if(n1 % i== 0)

        { result1= result1 +i;}

        }

        for(int i=1;i<n2;i++)

        { if(n2 % i == 0)

        { result2=result2+i;} }

        if(result1==n1&&result2==n2)

        System.out.print("Friendly Pair");

        else

        System.out.print("Not Friendly Pair");

    }}
```

OUTPUT:

5 3

Not Friendly Pair

---

7 5

Not Friendly Pair

---

220 284

Friendly Pair

/\*Problem 4: Write a Program to replace all 0's with 1 in a given integer. Given an integer as an input, all the 0's in the number has to be

replaced with 1. \*/

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args)

    {int n;

    Scanner obj=new Scanner(System.in);

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

    n=obj.nextInt();

    String s=Integer.toString(n);

    String str="";

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

    {

    if(s.charAt(i)=='0')

    str=str+'1';

    else str=str+s.charAt(i);

    }

    System.out.print(str);

    }

}
```



OUTPUT:

Enter an Integer: 10101001

1111111

---

Enter an Integer: 1110001

1111111

---

Enter an Integer: 10101

11111

/\*Problem 5: Write a Program for Printing an array into Zigzag fashion. Suppose you were given an array of integers, and you are

told to sort the integers in a zigzag pattern. In general, in a zigzag pattern, the first integer is less than the second integer, which is

greater than the third integer, which is less than the fourth integer, and so on. Hence, the converted array should be in the form of e1

< e2 > e3 < e4 > e5 < e6. \*/

```
import java.util.Scanner;

public class Main{

public static void main(String[] args){

int n;

Scanner obj=new Scanner(System.in);

System.out.println("Enter no of elements :");

n=obj.nextInt();

Integer arr[]=new Integer[n];

System.out.println("Enter elements :");

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

arr[i]=obj.nextInt();}

boolean flag=true;

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

{ if(flag)

{ if(arr[i]>arr[i+1])

{ int temp = arr[i];

arr[i] = arr[i+1];

arr[i+1] = temp;

}}

else

{ if (arr[i] < arr[i+1])

{ int temp= arr[i];
```

```
arr[i] = arr[i+1];  
arr[i+1] = temp;  
}}  
flag=!flag;  
}  
System.out.println("Elements are:");  
for(int i=0;i<n;i++) {  
System.out.print(arr[i]+" ");  
}}}
```

OUTPUT:

Enter no of elements : 5

Enter Elements: 1 5 4 3 8 5

Elements are:1 5 3 8 4 5

---

Enter no of elements : 6

Enter Elements: 4 3 6 4 2 1

Elements are: 3 4 1 4 6 2

---

Enter no of elements : 5

Enter Elements: 1 2 4 3 8

Elements are: 1 4 8 2 3

/\*Problem 6:Write a Program to rearrange positive and negative numbers in an array \*/

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args) {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter elements: ");
        int n=obj.nextInt();
        Integer arr[]=new Integer[n];
        System.out.print("Elements are : ");
        for(int i=0;i<n;i++){
            arr[i]=obj.nextInt();
            int k=0;
            for(int i=0;i<n;i++){
                if(arr[i]<0)
                { int temp=arr[i];
                    arr[i]=arr[k];
                    arr[k]=temp; k++;
                }
            }
            System.out.print("Arranged Elements are : ");
            for(int i=0;i<n;i++){
                System.out.print(arr[i]+" ");
            }
        }
    }
}
```

OUTPUT:

Enter no of elements : 4

Enter Elements: 1 -4 -5 3

Elements are: -5 -4 1 3

---

Enter no of elements : 5

Enter Elements: 0 -1 8 -5 0

Elements are: -5 -1 0 0 8

---

Enter no of elements : 6

Enter Elements: -3 -1 -2 0 1 5

Elements are: -3 -2 -1 0 1 5

/\*Problem 7: Program to find the saddle point coordinates in a given matrix. A saddle point is an element of the matrix, which is the

minimum element in its row and the maximum in its column.\*/

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args){

        Scanner obj=new Scanner(System.in);

        int n=obj.nextInt();

        Integer mat[][]=new Integer[n][n];

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

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

                mat[i][j]=obj.nextInt();

            }

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

                int min_row = mat[i][0], col_ind = 0;

                for (int j = 1; j < n; j++){

                    if (min_row > mat[i][j]){

                        min_row = mat[i][j];

                        col_ind = j;

                    }

                }

                int k;

                for (k = 0; k < n; k++){

                    if (min_row < mat[k][col_ind])

                        break;

                    if (k == n){

                        System.out.println("Value of Saddle Point " + min_row);

                        break;

                    }

                }

            }

        }

    }

}
```

OUTPUT:

3

1 2 3

4 5 6

7 8 9

Value of Saddle point: 7

---

2

4 5

3 5

Value of Saddle point: 4

---

2

5 6

1 0

Value of Saddle point: 5



/\*Problem 8: Program to find all the patterns of 0(1+)0 in the given string. Given a string containing 0's and 1's, find the total number

of 0(1+)0 patterns in the string and output it. 0(1+)0 - There should be at least one '1' between the two 0's. \*/

```
import java.util.Scanner;

public class Main{

static int find_pattern(char str[]){
char last = str[0];
int i = 1, counter = 0;
while (i < str.length){
if (str[i] == '1' && last == '0'){
while (str[i] == '1') i++;
if (str[i] == '0')
counter++; }
last = str[i]; i++; }
return counter;
}

public static void main (String args[]){
String string;
Scanner sc = new Scanner(System.in);
System.out.print("Enter the string : ");
string = sc.next();
char[] str = string.toCharArray();
System.out.print("Number of patterns : " + find_pattern(str));
}
}
```

OUTPUT:

Enter the string: 100100

Number of patterns: 1

---

Enter the string: 101110

Number of patterns: 1

---

Enter the string: 1010110

Number of patterns: 2

/\*Problem 9: Write a java program to delete vowels from given string using  
StringBuffer class\*/

```
import java.io.*;

public class Main{

    public static void main (String args[])throws IOException
    {
        String s;

        BufferedReader obj=new BufferedReader(new InputStreamReader(System.in));

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

        s = obj.readLine();

        String str="";

        for(int i=0;i<s.length();i++)
        {
            if(s.charAt(i)=='a' || s.charAt(i)=='e' || s.charAt(i)=='i' || s.charAt(i)=='o' || s.charAt(i)=='u')
                continue;

            str=str+s.charAt(i);
        }

        System.out.print(str);

    }
}
```

OUTPUT:

Enter the string: hello world

hll wrld

---

Enter the string: help me

hlp m

---

Enter the string: goodbye

gdby

/\* Problem 10: Write a java program to create a class named 'Bank ' with the following data members: Name of depositor Address of

depositor Account Number Balance in account Class 'Bank' has a method for each of the following: 1 - Generate a unique account

number for each depositor For first depositor, account number will be 1001, for second depositor it will be 1002 and so on 2 - Display

information and balance of depositor 3 - Deposit more amount in balance of any depositor 4 - Withdraw some amount from balance

deposited 5 - Change address of depositor After creating the class, do the following operations 1 - Enter the information (name, address,

account number, balance) of the depositors. Number of depositors is to be entered by user. 2 - Print the information of any depositor. 3 -

Add some amount to the account of any depositor and then display final information of that depositor 4 - Remove some amount from the

account of any depositor and then display final information of that depositor 5 - Change the address of any depositor and then display

the final information of that depositor 6 - Randomly repeat these processes for some other bank accounts. \*/

```
import java.util.Scanner ;

class Main{

String nm, addr ;

double bal ;

int accno;

static int num= 1001;

private void uniqueacc(){

accno=num++ ;}

Main()

{uniqueacc();}

public void input(){

Scanner sc= new Scanner(System.in) ;

System.out.print("Enter Name: ");
```

```
nm= sc.nextLine();
System.out.print("Enter Address: ");
addr= sc.nextLine();
System.out.print("Enter Balance: ");
bal = sc.nextDouble();
}

public void disp(){
System.out.println("Name="+nm+"\nAddress="+addr+ "\nBalance=" + bal+ "\nAccount no="
"+ accno); }

public void deposit(int amt)
{ bal+= amt; }

public void withdraw(int dep)
{if(bal>dep)
bal-= dep ;
else
System.out.println("Insufficient balance" ) ;}

public static void main( String s[]){
Main m=new Main();
m.input();
Scanner obj=new Scanner(System.in);
System.out.println("Enter Amount to deposit: ");
int amt=obj.nextInt();
m.deposit(amt);
System.out.println("Enter Amount to Withdraw: ");
int dr=obj.nextInt();
m.withdraw(dr);
m.disp();
}}
```

Enter Name: rohit

Enter Address: delhi

Enter Balance: 4535

Enter Amount to deposit: 66565

Enter Amount to withdraw: 4567

Name= rohit

Address= delhi

Balance= 66533.0

Account no=1001

---

Enter Name: elon

Enter Address: us

Enter Balance: 4678

Enter Amount to deposit: 65778

Enter Amount to withdraw: 456

Name= elon

Address= us

Balance= 70000.0

Account no=1056

---

Enter Name: riya

Enter Address: mumbai

Enter Balance: 3456

Enter Amount to deposit: 345

Enter Amount to withdraw: 234

Name= riya

Address= delhi

Balance= 3567

Account no=1452

/\*Problem 12: Write a Java program to create a class called ArrayDemo and overload arrayFunc() function. void arrayFunc(int [], int)

To find all pairs of elements in an Array whose sum is equal to a given number : Array numbers= [4, 6, 5, -10, 8, 5, 20], target=10 void

arrayFunc(int A[], int p, int B[], int q) Given two sorted arrays A and B of size p and q, Overload method arrayFunc() to merge elements

of A with B by maintaining the sorted order i.e. fill A with first p smallest elements and fill B with remaining elements\*/

```
import java.util.Scanner ;
import java.util.Arrays;
class Main{
public void arrayfunc(Integer arr[],int k){
Arrays.sort(arr);
int i=0,j=arr.length-1;
while(i<j){
if(arr[i]+arr[j]==k)
{
System.out.print(arr[i]+" "+arr[j]);
System.out.println();
}
if(arr[i]+arr[j]>k)
j--;
else
i++;} }
public void arrayfunc(Integer a[],int p,Integer b[],int q){
int i=0,j=0,k=0;
Integer res[]=new Integer[p+q];
while (i<p && j<q){
if (a[i] < b[j]){
```



```
res[k++] = a[i++];
}
else res[k++] = b[j++]; }
while (i< p)
res[k++] = a[i++];
while (j < q)
res[k++] = b[j++];
for(int x=0;x<p+q;x++){
System.out.print(res[x]+" ");
} }
public static void main( String s[]){
Main m=new Main();
Scanner obj=new Scanner(System.in);
int n=obj.nextInt();
Integer arr[]=new Integer[n];
for(int i=0;i<n;i++) {
arr[i]=obj.nextInt();
int k=obj.nextInt();
m.arrayfunc(arr,k);
int p=obj.nextInt();
Integer a[]=new Integer[p];
for(int i=0;i<p;i++) {
a[i]=obj.nextInt();
int q=obj.nextInt();
Integer b[]=new Integer[q];
for(int i=0;i<q;i++){
b[i]=obj.nextInt();
m.arrayfunc(a,p,b,q);
}}
```

OUTPUT:

7

4 3 9 2 7 5 1

5

1 4

2 3

-----

7

5 6 4 -10 5 20 8

10

-10 20

4 6

6 5

-----

7

4 6 5 -10 8 5 20

10

-10 20

4 6

5 5

/\*Problem: 13 Write a java program to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three

abstract methods namely rectangleArea() taking two parameters, squareArea() and circleArea() taking one parameter each. Now

create another class 'Area' containing all the three methods rectangleArea(),squareArea() and circleArea() for printing the area of

rectangle, square and circle respectively. Create an object of class Area and call all the three methods. \*/

```
abstract class Shape{
    abstract void rectangleArea(int a,int b);
    abstract void squareArea(int c);
    abstract void circleArea(int x);
}

class Area extends Shape{
    void rectangleArea(int a,int b){
        System.out.println(a*b);}
    void squareArea(int c){
        System.out.println(c*c);}
    void circleArea(int x){
        System.out.println(3.14*x*x);}
}

public class Main{
    public static void main (String[] args){
        Area obj=new Area();
        obj.rectangleArea(5,6);
        obj.squareArea(7);
        obj.circleArea(4);
    }
}
```

OUTPUT:

30

49

50.24

---

54

16

12.56

---

16

36

153.86

/\*Problem :14 Write a java program to implement abstract class and abstract method with following details: Create a abstract Base Class

Temperature Data members: double temp; Method members: void setTempData(double)  
abstract void changeTemp() Sub Class Fahrenheit

Data members: double ctemp; method member: Override abstract method changeTemp()  
to convert Fahrenheit temperature into degree

Celsius by using formula  $C = 5/9 * (F - 32)$  and display converted temperature Sub Class Celsius  
Data member: double ftemp; Method member:

Override abstract method changeTemp() to convert degree Celsius into Fahrenheit  
temperature by using formula  $F = 9/5 * c + 32$  and display  
converted temperature \*/

```
abstract class Temperature {  
    float temp;  
    void setTempData(float a){  
        temp=a;}  
    abstract void changeTemp();  
}  
class Fahrenheit extends Temperature{  
    float ctemp;  
    public void changeTemp(){  
        ctemp=((temp-32)*5)/9;  
        System.out.println("Fahrenheit:"+ctemp);  
    }  
class Celsius extends Temperature{  
    double ftemp;  
    public void changeTemp(){  
        ftemp=(9/5*(temp+32));  
        System.out.println("Celsius:"+ftemp);  
    }  
    public class Main{
```

```
public static void main(String[] args) {  
    Fahrenheit obj=new Fahrenheit();  
    obj.setTempData(95);  
    obj.changeTemp();  
    Celsius obj1=new Celsius();  
    obj1.setTempData(95);  
    obj1.changeTemp();  
}}
```

OUTPUT:

Fahrenheit: 35.0

Celcius: 127.0

---

Fahrenheit: 36.666668

Celcius: 130.0

---

Fahrenheit: 32.77778

Celcius: 123.0

/\*Program 15: Write a java program to create an interface that consists of a method to display volume () as an abstract method and

redefine this method in the derived classes to suit their requirements. Create classes called Cone, Hemisphere and Cylinder that

implements the interface. Using these three classes, design a program that will accept dimensions of a cone, cylinder and hemisphere

interactively and display the volumes. Volume of cone =  $(1/3)\pi r^2 h$  Volume of hemisphere =  $(2/3)\pi r^3$  Volume of cylinder =  $\pi r^2 h$ \*/

```
import java.util.Scanner;

interface dispvol{
    public void volume();}

class cone implements dispvol{
    int r, h;
    double pi=3.14;
    cone(int r, int h){
        this.r= r;
        this.h= h;}
    public void volume(){
        System.out.println("Volume of cone: "+ (pi*r*r*h)/3); }
}

class hemisphere implements dispvol{
    int r;
    double pi=3.14;
    hemisphere(int r){
        this.r= r;}
    public void volume(){
        System.out.println("Volume of      hemisphere:  "+ (2*pi*r*r*r)/3); }
}

class cylinder implements dispvol{
    int m, n;
```

```
double pi=3.14;
cylinder(int m, int n){
this.m= m;
this.n= n;}
public void volume(){
System.out.println("Volume of    cyliner: "+pi*m*m*n); }
}
public class Main {
public static void main(String[] args) {
Scanner sc= new Scanner(System.in);
int r = sc.nextInt();
int h= sc.nextInt();
cone obj= new cone(r,h);
obj.volume();
cylinder obj1= new cylinder(r,h);
obj1.volume();
hemisphere obj2= new hemisphere(r);
obj2.volume();
}}
```



OUTPUT:

3 5

Volume of cone:47.099

Volume of cylinder:141.2999

Volume of hemisphere:56.52

---

4 5

Volume of cone:83.7333

Volume of cylinder:251.200

Volume of hemisphere:133.973

---

6 7

Volume of cone:263.76

Volume of cylinder:791.28

Volume of hemisphere:452.16

/\*Program 16: Write a java program to accept and print the employee details during runtime. The details will include employee id, name, dept\_ Id. The program should raise an exception if user inputs incomplete or incorrect data. The entered value should meet the following conditions:

- (i) First Letter of employee name should be in capital letter.
- (ii) Employee id should be between 2001 and 5001
- (iii) Department id should be an integer between 1 and 5.

If the above conditions are not met then the application should raise specific exception else should complete normal execution.\*/

```
import java.util.Scanner;

class Myexception extends Exception {

    Myexception(String s) {
        super(s);}
}

class A {
    String name;
    int empid, dept_id;

    void getdata(int empid, String name, int dept_id)
    { this.empid = empid;
      this.name = name;
      this.dept_id = dept_id;
    }

    boolean checkname() {
        char x = name.charAt(0);
        return Character.isUpperCase(x);}

    boolean checkempid() {
        return empid > 2001 && empid < 5001;}

    boolean checkdept() {
```

```
return dept_id > 1 && dept_id < 5;}

void printdata() {

System.out.println("Name: " + name + "\nEmployee id: " + empid + "\nDepartment id: " +
dept_id); } }

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter Employee id: ");

int empid = sc.nextInt();

System.out.print("\nEnter name: ");

String name = sc.nextLine();

System.out.print("\nEnter department id:");

int dept_id = sc.nextInt();

A obj = new A();

try {

obj.getdata(empid, name, dept_id);

if (obj.checkname() && obj.checkempid() && obj.checkdept()) {

obj.printdata();}

else throw new Myexception("Wrong or incoplete data input");

}

catch (Myexception e) {

System.out.println("Exception: " + e.getMessage());}

}}
```

OUTPUT:

Enter employee id: 224

Enter name: shubhangi

Enter department id: 2002

Exception: Wrong or incomplete data input

---

Enter employee id: 203

Enter name: sonam

Enter department id: 2045

Exception: Wrong or incomplete data input

---

Enter employee id: 211

Enter name: roshan

Enter department id: 2023

Exception: Wrong or incomplete data input

/\*Program 17: Create a class MyCalculator which consists of a single method power(int, int). This method takes two integers, n and p, as

parameters and finds  $n^p$ . If either n or p is negative, then the method must throw an exception which says "n and p should be

non-negative".

Input Format

Each line of the input contains two integers, n and p .

Output Format

Each line of the output contains the result ,if neither of n and p is negative. Otherwise the output contains "n and p should be

non-negative".

Sample Input 3 5

2 4

0 0

-1 -2

-1 3

Sample Output

243 16 java.lang.Exception: n and p should not be zero. java.lang.Exception: n or p should not be negative. java.lang.Exception: n or p

should not be negative.

Explanation

In the first two cases, both n and p are positive. So, the power function returns the answer correctly. In the third case, both n and p are

zero. So, the exception, "n and p should not be zero." is printed.

In the last two cases, at least one out of n and p is negative. So, the exception, "n or p should not be negative." is printed for these two

cases.\*/

```
import java.util.Scanner;
```

```
class MyCalculator {
```

```
public long power(int n, int p) throws Exception
```

```
{ if(n == 0 && p == 0) throw new Exception("n and p should not be zero.");  
else if(n < 0 || p < 0) throw new Exception("n or p should not be negative.");  
else return (long)(Math.pow(n,p));  
}}  
  
public class Main {  
    public static final MyCalculator my_calculator = new MyCalculator();  
    public static final Scanner in = new Scanner(System.in);  
    public static void main(String[] args) {  
        while (in .hasNextInt()) {  
            int n = in .nextInt();  
            int p = in .nextInt();  
            try {  
                System.out.println(my_calculator.power(n, p));  
            } catch (Exception e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

OUTPUT:

3 5

243

0 0

java.lang.Exception: n and p should not be zero

---

3 5

243

-8 -2

java.lang.Exception: n and p should not be negative

---

3 5

243

-8 0

java.lang.Exception: n and p should not be negative and zero

/\*Program 18: Write a java file handling program to count and display the number of palindrome present in a text file "myfile.txt".

Example: If the file "myfile.txt" contains the following lines, My name is NITIN Hello aaa and bbb word How are You ARORA is my friend

Output will be => 4 // NITIN, aaa, bbb, ARORA \*/

```
import java.io.*;
import java.util.*;
public class Main{
    static int ispalindrome(String word){
        String rev="";
        char ch;
        for(int i=0;i<word.length();i++) {
            ch=word.charAt(i);
            rev=ch+rev;}
        if(rev.equals(word)) return 1;
        else return 0;}
    public static void main(String[] args)throws IOException{
        File f=new File("PalindromeCheck.txt");
        FileReader fr=new FileReader(f);
        BufferedReader br=new BufferedReader(fr);
        String str;
        String [] words=null;
        int count=0;
        while((str = br.readLine()) != null){
            words=str.split(" ");
            for(String word:words){
                int res=ispalindrome(word);
                if(res==1){
                    count++;
```



```
System.out.println(word);  
}}}  
System.out.println(count);  
}}
```

OUTPUT:

NITIN

bbb

aaa

ARORA

4

/\*Program 19: Write a program MultiThreads that creates two threads-one thread with the name CSthread and the other thread named

ITthread. Each thread should display its respective name and execute after a gap of 500 milliseconds. Each thread should also display a

number indicating the number of times it got a chance to execute.\*/

```
public class MultiThreads{
    public void Dname(String s){
        for(int i=1;i<=4;i++){
            try{
                Thread.sleep(500);}
            catch(Exception e){
                System.out.println(e);}
            System.out.println(s+ " Executed : "+i+" times ");
        }
    }
    public static void main(String args[]) throws Exception{
        MultiThreads o =new MultiThreads();
        Thread CSthread =new Thread(){
            public void run(){
                o.Dname("CSthread");}
        };
        Thread ITthread =new Thread(){
            public void run(){
                o.Dname("ITThread");}
        };
        CSthread.start();
        CSthread.join();
        ITthread.start();
    }
}
```

OUTPUT:

CSThread Executed: 1 times

CSThread Executed: 2 times

CSThread Executed: 3 times

CSThread Executed: 4 times

ITThread Executed: 1 times

ITThread Executed: 2 times

ITThread Executed: 3 times

ITThread Executed: 4 times

/\*Program 20: Write a java program for to solve producer consumer problem in which a producer produce a value and consumer consume

the value before producer generate the next value.\*/

```
class RationShop{
    boolean ip=false;
    int item;
    synchronized public void producer(int x){
        if(ip){
            try {
                System.out.println("Producer is waiting for consumer");
                wait();}
            catch(InterruptedException e){}
        }
        item=x;
        System.out.println(item+" is produced");
        ip=true;
        notify();}
    synchronized public void consumer(){
        if(!ip){
            try{
                System.out.println("Consumer is waiting for producer"); wait();}
            catch(InterruptedException e){}
        }
        System.out.println(item+" is consumed"); ip=false;
        notify();
    }}
class Produce extends Thread{
    RationShop rs;
    public void run(){
```

```
for(int i=1;i<=5;i++){  
    rs.producer(i);}  
}  
  
public Produce(RationShop rs){  
    this.rs=rs;}  
}  
  
class Consumer extends Thread{  
    RationShop rs;  
  
    public void run(){  
        for(int i=1;i<=5;i++){  
            rs.consumer();  
        }  
    }  
  
    public Consumer(RationShop rs){  
        this.rs=rs;}  
    }  
  
class Shop{  
    public static void main(String args[]){  
        RationShop rs=new RationShop();  
        Produce pr= new Produce(rs);  
        Consumer cr= new Consumer(rs);  
        pr.start();  
        cr.start();  
    }  
}
```

OUTPUT:

Producer produced-0

Producer produced-1

Consumer consumed-0

Producer produced-2

---

Producer produced-0

Producer produced-1

Consumer consumed-0

Producer produced-2

Consumer consumed-1

Consumer consumed-2

---

Producer produced-0

Producer produced-1

Consumer consumed-0

Producer produced-2

Consumer consumed-1

Consumer consumed-2

Producer produced-3

Producer produced-4

Consumer consumed-3

/\*Program 21: Write a method removeEvenLength that takes an ArrayList of Strings as a parameter and that removes all of the strings

of even length from the list.(Use ArrayList)\*/

```
import java.util.*;

class Main{

public static void main(String[] args) {

ArrayList<String> array = new ArrayList<String>();

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of string you want to insert in ArrayList");

int n = sc.nextInt();

System.out.println("Enter the String into ArrayList");

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

String s = sc.next();

array.add(s);

}

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

String word = array.get(i);

if (word.length() % 2 == 0) {

array.remove(i);

i--;

}}

System.out.println("\n ArrayList after removal of even length Strings");

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

System.out.println(array.get(i));    }

sc.close();

}}
```

OUTPUT:

Enter number of string you want to insert in Arraylist:

4

Enter the string into Arraylist:

this is an array

Arraylist after removal of even length string:

array

---

Enter number of string you want to insert in Arraylist:

4

Enter the string into Arraylist:

there are nine student

Arraylist after removal of even length string:

are

student

---

Enter number of string you want to insert in Arraylist:

6

Enter the string into Arraylist:

all professor are from same the city

Arraylist after removal of even length string:

all

professor

are the



/\*Program 22: Write a method swapPairs that switches the order of values in an ArrayList of Strings in a pairwise fashion. Your method

should switch the order of the first two values, then switch the order of the next two, switch the order of the next two, and so on.

For example, if the list initially stores these values: {"four", "score", "and", "seven", "years", "ago"} your method should switch the first

pair, "four", "score", the second pair, "and", "seven", and the third pair, "years", "ago", to yield this list: {"score", "four", "seven", "and",

"ago", "years"} If there are an odd number of values in the list, the final element is not moved.

For example, if the original list had been: {"to", "be", "or", "not", "to", "be", "hamlet"} It would again switch pairs of values, but the final

value, "hamlet" would not be moved, yielding this list: {"be", "to", "not", "or", "be", "to", "hamlet"}\*/

```
import java.util.*;

class Main {

    public static void main(String[] args) {

        ArrayList<String> array = new ArrayList<String>();

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number of string you want to insert in ArrayList");

        int n = sc.nextInt();

        System.out.println("Enter the String into ArrayList");

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

        {String s = sc.next(); array.add(s);}

        for(int i=0;i<array.size()-1;i=i+2)

        {String s = array.get(i);

        array.set(i,array.get(i+1));

        array.set(i+1,s);}

        System.out.println("\n ArrayList after removal of even length Strings");

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

        {System.out.print(array.get(i) + " ");}
```

```
sc.close();  
}}
```

OUTPUT:

Enter number of string you want to insert in ArrayList:

6

Enter the string into ArrayList:

asdv das d sdf sdfsg sgd

ArrayList after removal of even length string:

das asdv sdf d sgd sdfsg

---

Enter number of string you want to insert in ArrayList:

3

Enter the string into ArrayList:

abc aaaa vcdv

ArrayList after removal of even length string:

aaaa abc vcdv

---

Enter number of string you want to insert in ArrayList:

2

Enter the string into ArrayList:

vbv vhiuh

ArrayList after removal of even length string:

vhiuh vbv

/\*Program 23: Write a method called alternate that accepts two Lists of integers as its parameters and returns a new List containing

alternating elements from the two lists, in the following order:

- First element from first list
- First element from second list
- Second element from first list
- Second element from second list
- Third element from first list
- Third element from second list
- ...

If the lists do not contain the same number of elements, the remaining elements from the longer list should be placed consecutively at

the end. For example, for a first list of (1, 2, 3, 4, 5) and a second list of (6, 7, 8, 9, 10, 11, 12), a call of alternate(list1, list2) should return

a list containing (1, 6, 2, 7, 3, 8, 4, 9, 5, 10, 11, 12). Do not modify the parameter lists passed in.\*/

```
import java.util.*;

class Main {

    public static ArrayList<Integer> alternate(ArrayList<Integer> a1,
        ArrayList<Integer> a2){ ArrayList<Integer> ans = new ArrayList<Integer>();
        int i=0,j=0;
        while(i<a1.size() && j<a2.size()){
            ans.add(a1.get(i));
            ans.add(a2.get(j));
            i++;
            j++;}
        while(i<a1.size()){
            ans.add(a1.get(i));
            i++; }
        while(j<a2.size()){
```

```
ans.add(a2.get(j));
j++;}
return ans; }

public static void main(String[] args) {
    ArrayList<Integer> array = new ArrayList<Integer>();
    ArrayList<Integer> array2 = new ArrayList<Integer>();
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter number of integer you want to insert in ArrayList");
    int n1 = sc.nextInt();
    System.out.println("Enter the integer into ArrayList1");
    for(int i=0;i<n1;i++){
        int x = sc.nextInt();
        array.add(x);}
    System.out.println("Enter number of integer you want to insert in ArrayList");
    int n2 = sc.nextInt();
    System.out.println("Enter the integer into ArrayList1");
    for(int i=0;i<n2;i++){
        int x = sc.nextInt();
        array2.add(x);}
    ArrayList<Integer> ans = alternate(array,array2);
    System.out.println("\nArrayList after removal of even length Strings");
    for(int i=0;i<ans.size();i++){
        System.out.print(ans.get(i) + " ");}
    sc.close();
}}
```

OUTPUT:

Enter number of integer you want to insert in ArrayList:

3

Enter the integer into ArrayList:

1 2 3

Enter number of integer you want to insert in ArrayList:

3

Enter the integer into ArrayList:

4 5 6

ArrayList after removal of even length string:

1 4 2 5 3 6

---

Enter number of integer you want to insert in ArrayList:

2

Enter the integer into ArrayList:

4 6

Enter number of integer you want to insert in ArrayList:

3

Enter the integer into ArrayList:

1 7 5

ArrayList after removal of even length string:

4 1 6 7 5

---

Enter number of integer you want to insert in ArrayList:

2

Enter the integer into ArrayList:

5 9

Enter number of integer you want to insert in ArrayList:

4

Enter the integer into ArrayList:

7 2 6 9

ArrayList after removal of even length string:

5 7 9 2 6 9