Contents

[Q1. Write a program to find the factorial of a given number using recursion technique. 2](#_Toc435734244)

[Q2. Write a program to print only the magic numbers in given array of any size. 3](#_Toc435734245)

[Q3. Write a program to decrypt input strings using an input shift value. 3](#_Toc435734246)

[Q4. Write a program to find the saddle point of a matrix. 5](#_Toc435734247)

[Q5. Write a program to print the prime factors of numbers in an array. 6](#_Toc435734248)

[Q6. Write a program to input a number and print its equivalent roman numeral. 7](#_Toc435734249)

[Q7. Write a program to accept a string and print its piglatin equivalent. 8](#_Toc435734250)

[Q8. Write a program to input a birthday and print the its day of the year. 9](#_Toc435734251)

[Q9. Write a program to input an array of any size, sort it using bubble sort technique and search for a particular value in the array using binary search technique. 9](#_Toc435734252)

[Q10. Write a program to accept a number n and print the spiral form of a matrix from n2 to 1. 10](#_Toc435734253)

[Q11. Write a program to print a unique numbers in a range. 13](#_Toc435734254)

[Q12. Write a program to count the number of vowels in a string using recursive technique. 14](#_Toc435734255)

[Q13. Write a program to input a string, reverse it using recursive technique and check if it’s a palindrome. 15](#_Toc435734256)

[Q14. Write a program to create a text file, store some lines of text and read from it. 16](#_Toc435734257)

[Q15. Write a program to create a stack with linked list implementation, push, pop and display values in it. 17](#_Toc435734258)

[Q16. Write a program to create a stack as an array, push, pop and display values in it. 18](#_Toc435734259)

[Q17. Write a program to accept a date and convert it to its corresponding data number and vice versa. 20](#_Toc435734260)

[Q18. Write a program to find the difference in days between two dates. 21](#_Toc435734261)

[Q19. Write a program to create a queue as an array, insert, delete and display elements. 23](#_Toc435734262)

[Q20. Write a program to input 100 integers in an array, sort it using selection sort technique and print. 24](#_Toc435734263)

[Q21. Write a program to find the sum of the series x2/11 + x4/22 + x6/33 +…. using recursion technique. 26](#_Toc435734264)

[Q22. Write a program to compute the fine for returning a book late to a library. Show the implementation of Inheritance. 27](#_Toc435734265)

[Q23. Write a program to check if a give matrix is a wondrous square. 28](#_Toc435734266)

[Q24. Write a program to input integers into an array, using recursion, sort it with selection sort technique and print. 30](#_Toc435734267)

[Q25. Write a program to input two points with x and y coordinates and find the distance between them. 30](#_Toc435734268)

[Q26. Write a program to create a Double ended queue, method for insertion and deletion from both ends and a display method. 32](#_Toc435734269)

[Q27. Write a program to create a linked list implemented as a queue, insert, delete and display elements in it. 33](#_Toc435734270)

[Q28. Write a program to create a binary file which stores the marks obtained by students, read and write from the file. 34](#_Toc435734271)

[Q28. Write a program to create and abstract class Employee and a subclass Programmer to calculate salary. Implement the concept of inheritance. 36](#_Toc435734272)

[Q30. Write a program creating a super class Data and a subclass Process to print and count the words beginning with a vowel in a string. 37](#_Toc435734273)

## Q1. Write a program to find the factorial of a given number using recursion technique.

class factorial

{

int n, f;

factorial()

{

n=0;

f=1;

}

public int fact(int num)

{

if(num==0)

return 1;

else

return fact(num-1)\*num;

}

void getnumber(int x)

{

n=x;

f=fact(n);

System.out.println("Factorial of "+x+" is "+f);

}

public static void main(int n1)

{

factorial f1=new factorial();

f1.getnumber(n1);

}

}

## Q2. Write a program to print only the magic numbers in given array of any size.

import java.util.\*;

class Magic

{

public static void main(int size)

{

int sum=0,r;

Scanner s=new Scanner(System.in);

int num[]=new int[size];

System.out.println("Enter "+size+" numbers");

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

{

num[i]=s.nextInt();

}

System.out.print("The Magic numbers are: ");

for(int j=0;j<size;j++)

{

int n=num[j];

do

{

sum=0;

while(n>0)

{

r=n%10;

n=n/10;

sum=sum+r;

}

n=sum;

}while(sum>=10);

if(sum==1)

System.out.print(num[j]+" ");

}

}

}

## Q3. Write a program to decrypt input strings using an input shift value.

import java.util.\*;

class Decryption

{

public static void main()

{

Scanner s=new Scanner(System.in);

int sv,sv1,val,val1;

String str;

char ch,ch1,ch2,c,c1,c2;

System.out.println("Input coded text");

str=s.nextLine();

if(str.length()>=100)

{

System.out.println("Invalid entry");

System.exit(5);

}

System.out.println("Input shift value");

sv=(s.nextInt())-1;

if(sv<=0||sv>=26)

{

System.out.println("Invalid entry");

System.exit(5);

}

System.out.println("Decoded text");

String decod="";

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

{

ch=str.charAt(i);

val=(int)ch;

if(val+sv<=90)

{

ch1=(char)(val+sv);

decod=decod+ch1;

}

else if(val+sv>90)

{

val1=90-val;

sv1=(sv-val1)-1;

ch1=(char)(65+sv1);

decod=decod+ch1;

}

}

for(int j=0;j<decod.length();j++)

{

c=decod.charAt(j);

if(c=='Q')

{

c1=decod.charAt(j+1);

c2=decod.charAt(j-1);

if(c1=='Q'||c2=='Q')

{

decod=decod.replace('Q',' ');

continue;

}

}

}

System.out.println(decod);

System.out.println();

}

}

## Q4. Write a program to find the saddle point of a matrix.

import java.util.\*;

public class SaddlePoint

{

public static void main() {

Scanner s=new Scanner(System.in);

int row,lrow,prow,col,hcol,pcol,ARR[][],flag=0;

System.out.print("Enter the size of row : ");

row=s.nextInt();

System.out.print("Enter the size of column : ");

col=s.nextInt();

ARR=new int[row][col];

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

{

for(int j=0;j<col;j++)

{

ARR[i][j]=s.nextInt();

}

}

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

{

lrow=ARR[i][0];prow=0;

for(int j=0;j<col;j++)

if(ARR[i][j]<lrow)

{

lrow=ARR[i][j];

prow=j;

}

hcol=ARR[0][prow];pcol=0;

for(int k=0;k<row;k++)

if(ARR[k][prow]>hcol)

{

hcol=ARR[k][prow];

pcol=k;

}

if(lrow==hcol)

{

System.out.println("Saddle Point found at Position Row "+(prow+1)+" Coloumn "+(pcol+1)+" . Element Present = "+hcol);

flag=1;

}

}

}

}

## Q5. Write a program to print the prime factors of numbers in an array.

import java.util.\*;

class PrimeFactors

{

public static void main(int size)

{

Scanner s=new Scanner(System.in);

int num[]=new int[size];

System.out.println("Enter "+size+" numbers");

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

{

num[i]=s.nextInt();

}

for(int j=0;j<size;j++)

{

System.out.print("The Prime Factors of "+num[j]+" are : ");

int i=2;

while(num[j]>1)

{

if(num[j]%i == 0)

{

System.out.print(i+" ");

num[j]=num[j]/i;

}

else

i++;

}

System.out.println();

}

}

}

## Q6. Write a program to input a number and print its equivalent roman numeral.

import java.util.\*;

class RomanNumbers

{

public static void main()

{

Scanner s=new Scanner(System.in);

String num;

int l=0,sum=0;

char ch;

System.out.println("Enter a roman number");

num=s.nextLine();

l=num.length();

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

{

ch=num.charAt(i);

switch(ch)

{

case 'M' : sum=sum+1000;

break;

case 'D' : sum=sum+500;

break;

case 'C' : sum=sum+100;

break;

case 'L' : sum=sum+50;

break;

case 'X' : sum=sum+10;

break;

case 'V' : sum=sum+5;

break;

case 'I' : sum=sum+1;

break;

default :

System.out.println("Invalid input");

}

}

System.out.println(sum);

}

}

## Q7. Write a program to accept a string and print its piglatin equivalent.

class PigLatin

{

public static void main(String str)

{

str=str.toUpperCase();

char ch;

String str1="";

String str2="";

int i=0;

int ctr=1;

while(i<str.length()&&ctr==1)

{

ch=str.charAt(i);

if(ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')

{

str2=str.substring(0,i);

str1=str.substring(i);

ctr++;

}

i++;

}

str2=str1+str2+"AY";

System.out.println(str2);

}

}

## Q8. Write a program to input a birthday and print the its day of the year.

import java.util.\*;

class BirthDay

{

public static void main()

{

Scanner s=new Scanner(System.in);

int flag=0;

int day[]={31,28,31,30,31,30,31,31,30,31,30,31};

System.out.println("Enter your date of birth in dd mm form");

int dd=s.nextInt();

int mm=s.nextInt();

if(mm>12||dd>day[mm-1])

{

System.out.println("INVALID DATE");

System.exit(0);

}

else

{

System.out.println("VALID DATE");

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

flag=flag+day[i];

flag=flag-(day[mm-1]-dd);

System.out.println(flag);

}

}

}

## Q9. Write a program to input an array of any size, sort it using bubble sort technique and search for a particular value in the array using binary search technique.

class myarray

{

public static void main(int arr[],int val)

{

for(int j=0;j<arr.length;j++)

{

for(int k=0;k<arr.length-1;k++)

{

if(arr[k]==arr[k+1])

{

int temp=arr[k];

arr[k]=arr[k+1];

arr[k+1]=temp;

}

}

}

int pos=-1;

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

{

if(val==arr[i])

{

pos=i+1;

}

}

if(pos==-1)

System.out.println("-999");

else

System.out.println("Element found at position: "+pos);

}

}

## Q10. Write a program to accept a number n and print the spiral form of a matrix from n2 to 1.

import java.util.\*;

public class spiral

{

static void main()

{

Scanner in=new Scanner(System.in);

System.out.println("Enter a number");

int n=in.nextInt();

int a[][]=new int[n][n];

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

{

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

{

a[k][h]=0;

}

}

int b=n\*n;

int i=0;

int j=0;

int flag=0;

loop:while(b>0)

{

try

{

if(a[j][i]==0)

{

if(flag==0)

{

a[j][i]=b;

b--;

i++;

}

else if(flag==1)

{

a[j][i]=b;

b--;

j++;

}

else if(flag==2)

{

a[j][i]=b;

i--;

b--;

}

else if(flag==3)

{

a[j][i]=b;

j--;

b--;

}

}

else

{

if(flag==0)

{

i--;

j++;

}

else if(flag==1)

{

j--;

i--;

}

else if(flag==2)

{

i++;

j--;

}

else if(flag==3)

{

j++;

i++;

}

flag++;

if(flag==4)

flag=0;

}

}

catch(ArrayIndexOutOfBoundsException e)

{

if(flag==0)

{

i--;

j++;

}

else if(flag==1)

{

j--;

i--;

}

else if(flag==2)

{

i++;

j--;

}

else if(flag==3)

{

j++;

i++;

}

flag++;

if(flag==4)

flag=0;

continue loop;

}

}

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

{

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

{

System.out.print(a[k][h]+"\t");

}

System.out.println();

}

}

}

## Q11. Write a program to print a unique numbers in a range.

import java.util.\*;

class Unique

{

public static void main()

{

Scanner obj=new Scanner(System.in);

System.out.println("Enter 2 numbers m to be smaller than n");

int m=obj.nextInt();

int n=obj.nextInt();

int ctr=0;

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

{

int x=i;

String s =Integer.toString(x);

int l=s.length();

for(int j=0;j<l-1;j++)

{

for(int k=j+1;k<l;k++)

{

if(s.charAt(j)==s.charAt(k))

{

ctr=1;

break;

}

}

}

if (ctr==0)

System.out.print(x+" ");

ctr=0;

}

}

}

## Q12. Write a program to count the number of vowels in a string using recursive technique.

import java.util.\*;

class Check

{

Scanner s=new Scanner(System.in);

String str;

int l;

int w=0;

void InputString()

{

System.out.println("Enter a string");

str=" "+s.nextLine();

l=str.length()-1;

}

void Counter()

{

if(l>=0)

{

char ch=str.charAt(l);

if(ch==' ')

{

char ch1=str.charAt(l+1);

if(ch1=='A'||ch1=='E'||ch1=='I'||ch1=='O'||ch1=='U')

{

w++;

}

}

l--;

Counter();

}

}

void Disp()

{

InputString();

Counter();

System.out.println("number of words beginning with a captial vowel : "+w);

}

}

## Q13. Write a program to input a string, reverse it using recursive technique and check if it’s a palindrome.

import java.util.\*;

class Revstr

{

Scanner s=new Scanner(System.in);

String str="",revstr="";

int l;

char ch;

void getStr()

{

str=s.nextLine();

l=((str.length())-1);

}

void recReverse(int len)

{

if(len>=0)

{

ch=str.charAt(len);

revstr=revstr+ch;

len--;

recReverse(len);

}

}

void Check()

{

System.out.println("Original: "+str);

recReverse(l);

System.out.println("Reverse: "+revstr);

if(revstr.equals(str))

System.out.println("is Palindrome");

else

System.out.println("is not Palindrome");

}

void main()

{

System.out.println("Enter a string");

getStr();

Check();

}

}

## Q14. Write a program to create a text file, store some lines of text and read from it.

import java.util.\*;

import java.io.\*;

class TextFiles

{

public void write()

{

String str;

Scanner s=new Scanner(System.in);

try

{

FileWriter fw=new FileWriter("names.txt",true);

BufferedWriter bw=new BufferedWriter(fw);

PrintWriter pw=new PrintWriter(bw);

System.out.println("Enter 5 names");

for(int i=0;i<5;i++)

{

str=s.nextLine();

pw.println(str);

}

pw.close();

bw.close();

fw.close();

}

catch(IOException e)

{

System.err.println(e);

}

}

public void read()

{

String str;

Scanner s=new Scanner(System.in);

try

{

FileReader fr=new FileReader("names.txt");

BufferedReader br=new BufferedReader(fr);

while((str=br.readLine())!=null)

{

System.out.println(str);

}

br.close();

fr.close();

}

catch(IOException e)

{

System.err.println(e);

}

}

public void main()

{

write();

read();

}

}

## Q15. Write a program to create a stack with linked list implementation, push, pop and display values in it.

class StackAsList

{

int data;

StackAsList temp, next, start;

StackAsList()

{

data=0;

temp=next=start=null;

}

void push(int k)

{

StackAsList temp=new StackAsList();

temp.data=k;

if(start==null)

{

start=temp;

}

else

{

temp.next=start;

start=temp;

}

}

void display()

{

StackAsList temp=new StackAsList();

temp=start;

if(temp!=null)

{

while(temp.next!=null)

{

System.out.print(temp.data+"--->");

temp=temp.next;

}

System.out.print(temp.data);

}

else

System.out.print("Stack underflow");

}

int pop()

{

if(start==null)

{

System.out.println("Stack underflow");

return -1;

}

int dat=start.data;

start=start.next;

return dat;

}

}

## Q16. Write a program to create a stack as an array, push, pop and display values in it.

import java.util.\*;

class Stack

{

int stack[]=new int[3];

int top;

void push(int k)

{

if(top+1==3)

{

System.out.println("Overflow");

}

else

{

++top;

stack[top]=k;

}

}

void pop()

{

if(top==-1)

{

System.out.println("Underflow");

}

else

{

System.out.println("Value popped: "+stack[top]);

--top;

}

}

void display()

{

for(int i=top;i>-1;i--)

{

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

}

}

void main()

{

Scanner s=new Scanner(System.in);

int val,ch;

do

{

System.out.println("Enter choice");

System.out.println("1.push");

System.out.println("2.pop");

System.out.println("3.display");

System.out.println("4.exit");

ch=s.nextInt();

switch(ch)

{

case 1: System.out.println("Enter value to be pushed");

val=s.nextInt();

push(val);

break;

case 2: pop();

break;

case 3: display();

break;

case 4: System.exit(0);

break;

default: System.out.println("wrong input");

}

}while(ch!=4);

}

}

## Q17. Write a program to accept a date and convert it to its corresponding data number and vice versa.

import java.util.\*;

class Date

{

Scanner s=new Scanner(System.in);

int days[]={31,28,31,30,31,30,31,31,30,31,30,31};

int flag,f;

int dd, mm;

public Date()

{

f=0;

flag=0;

dd=0;

mm=0;

}

public Date(int nd, int nm)

{

f=0;

flag=0;

dd=nd;

mm=nm;

}

int datetodatenumber()

{

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

{

flag=flag+days[i];

}

flag=flag-(days[mm-1]-dd);

return flag;

}

Date datenumbertodate(int dn)

{

Date d=new Date();

int ctr=1;

int sum=days[0];

for(int i=1;i<12;i++)

{

if(sum>=dn)

{

d.mm=ctr;

break;

}

else

{

sum=sum+days[i];

ctr++;

}

}

int d1=(sum-(sum-dn))-days[ctr-1];

System.out.println(d1+" - "+d.mm);

return d;

}

}

## Q18. Write a program to find the difference in days between two dates.

import java.util.\*;

class DateDifference

{

static Scanner s=new Scanner(System.in);

int month[]={0,31,28,31,30,31,30,31,31,30,31,30,31};

public int isLeap(int y)

{

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

return 29;

else

return 28;

}

public boolean dateValidate(int d, int m, int y)

{

month[2]=isLeap(y);

if(m<0 || m>12 || d<0 || d>month[m] || y<0 || y>9999)

return false;

else

return true;

}

int dayno(int d, int m, int y)

{

int dn=0;

month[2]=isLeap(y);

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

{

dn=dn+month[i];

}

dn=dn+d;

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

{

if(isLeap(i)==29)

dn=dn+366;

else

dn=dn+365;

}

return dn;

}

public static void main()

{

DateDifference ob=new DateDifference();

System.out.print("Enter the 1st date in (dd/mm/yyyy) format: ");

String date1=s.nextLine().trim();

int p,q;

p=date1.indexOf("/");

int d1=Integer.parseInt(date1.substring(0,p));

q=date1.lastIndexOf("/");

int m1=Integer.parseInt(date1.substring(p+1,q));

int y1=Integer.parseInt(date1.substring(q+1));

System.out.print("Enter the 2nd date in (dd/mm/yyyy) format: ");

String date2=s.nextLine().trim();

p=date2.indexOf("/");

int d2=Integer.parseInt(date2.substring(0,p));

q=date2.lastIndexOf("/");

int m2=Integer.parseInt(date2.substring(p+1,q));

int y2=Integer.parseInt(date2.substring(q+1));

if(ob.dateValidate(d1,m1,y1)==true && ob.dateValidate(d2,m2,y2)==true)

{

int a=ob.dayno(d1,m1,y1);

int b=ob.dayno(d2,m2,y2);

System.out.print("Output : Difference = "+Math.abs(a-b)+" days.");

}

else

System.out.println("Invalid Date");

}

}

## Q19. Write a program to create a queue as an array, insert, delete and display elements.

import java.util.\*;

class Queue

{

int f=0,r=-1;

int q[]=new int[5];

void insert(int k)

{

if(r<q.length-1)

{

++r;

q[r]=k;

}

else

System.out.println("Overflow");

}

void delete()

{

if(r==-1&&f==0)

System.out.println("Underflow");

else

{

System.out.println("Value deleted: "+q[f]);

++f;

}

}

void display()

{

int i;

for(i=f;i<r;i++)

{

System.out.print(q[i]+"--->");

}

System.out.print(q[i]);

}

void main()

{

int val,ch;

Scanner s=new Scanner(System.in);

do

{

System.out.println();

System.out.println("Enter choice");

System.out.println("1.insert");

System.out.println("2.delete");

System.out.println("3.display");

System.out.println("4.exit");

ch=s.nextInt();

switch(ch)

{

case 1: System.out.println("Enter value to be inserted");

val=s.nextInt();

insert(val);

break;

case 2: delete();

break;

case 3: display();

break;

case 4: System.exit(0);

break;

default: System.out.println("wrong input");

}

}while(ch!=4);

}

}

## Q20. Write a program to input 100 integers in an array, sort it using selection sort technique and print.

import java.util.\*;

class sorter

{

Scanner s=new Scanner(System.in);

int arr[]=new int[100];

void readlist()

{

System.out.println("Enter 100 integers");

for(int i=0;i<100;i++)

{

arr[i]=s.nextInt();

}

}

void displaylist()

{

int i;

System.out.println("The 100 integers are: ");

for(i=0;i<99;i++)

{

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

}

System.out.print(arr[i]);

}

int indexofmin(int startindex)

{

int min=0;

for(int i=startindex;i<arr.length;i++)

{

min=i;

for(int j=i+1;j<arr.length;j++)

{

if(arr[j]<arr[min])

{

min=j;

}

}

}

return min;

}

public void secletionsort()

{

int min;

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

{

min=i;

for(int j=i+1;j<arr.length;j++)

{

if(arr[j]<arr[min])

{

min=j;

}

}

int temp=arr[i];

arr[i]=arr[min];

arr[min]=temp;

}

for(int k=0;k<arr.length;k++)

{

System.out.print(arr[k]+" ");

}

}

}

## Q21. Write a program to find the sum of the series x2/11 + x4/22 + x6/33 +…. using recursion technique.

import java.util.\*;

class sum\_series

{

Scanner s=new Scanner(System.in);

int x,n,p1,k,k1,pow,j;

double sum;

sum\_series()

{

j=0;

pow=2;

k=0;

k1=0;

x=0;

p1=1;

sum=0.0;

n=0;

}

void readLimit()

{

System.out.println("Enter n");

n=s.nextInt();

}

int getPower(int m,int p)

{

if(p==0)

return p1;

else

{

p1=p1\*m;

p=p-1;

return getPower(m,p);

}

}

void Sum()

{

System.out.println("Enter x");

x=s.nextInt();

readLimit();

for(int i=1,j=1;i<=n;i++,pow+=2,j++)

{

k=(getPower(x,pow));

System.out.println(i+" : "+k);

k1=(getPower(i,j));

System.out.println(i+" : "+k1);

sum=sum+(k/k1);

}

System.out.println(sum);

}

}

## Q22. Write a program to compute the fine for returning a book late to a library. Show the implementation of Inheritance.

class Library

{

String name,author;

double p;

Library(String n,String a, double np)

{

name=n;

author=a;

p=np;

}

void show()

{

System.out.println(name+" by "+author+" costs "+p);

}

}

class Compute extends Library

{

int d;

double r;

Compute(String n,String a, double np,int dd)

{

super(n,a,np);

d=dd;

r=0.0;

}

void fine()

{

if(d>7&&d<12)

r=2\*d;

else if(d>12&&d<17)

r=3\*d;

else if(d>=17)

r=5\*d;

}

void display()

{

show();

fine();

System.out.print("Number of excess days: "+(d-7)+" Total amount to be paid: "+((((2/100)\*p)\*(d-7))+r)+".");

}

}

## Q23. Write a program to check if a give matrix is a wondrous square.

import java.util.\*;

class WonderSquare

{

int arr[][],arr1[];;

int n,i,j,x=0,r,c;

int flag;

Scanner s=new Scanner(System.in);

public void take()

{

System.out.println("Enter the size of array:");

n=s.nextInt();

arr=new int[n][n];

arr1=new int[2\*n];

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

{

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

{

System.out.println("Enter the value:");

arr[i][j]=s.nextInt();

}

}

System.out.println("The matrix is");

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

{

r=0;

c=0;

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

{

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

r=r+arr[i][j];

c=c+arr[j][i];

}

System.out.println();

arr1[x]=r;

arr1[x+n-1]=c;

x++;

}

for(i=0;i< x;i++)

{

if(arr1[i]!= 0.5 \* n \* (n\*n + 1))

break;

}

if(i==x)

System.out.println("YES IT REPRESENTS A WONDROUS SQUARE.");

else

System.out.println("IT IS NOT A WONDROUS SQUARE.");

}

public static void main(String args[])throws Exception

{

WonderSquare ob=new WonderSquare();

ob.take();

}

}

## Q24. Write a program to input integers into an array, using recursion, sort it with selection sort technique and print.

import java.util.\*;

class SorterRecurison

{

Scanner s=new Scanner(System.in);

void selectionSort(int[] array, int startIndex)

{

if ( startIndex >= array.length - 1 )

return;

int minIndex = startIndex;

for ( int index = startIndex + 1; index < array.length; index++ )

{

if (array[index] < array[minIndex] )

minIndex = index;

}

int temp = array[startIndex];

array[startIndex] = array[minIndex];

array[minIndex] = temp;

selectionSort(array, startIndex + 1);

}

void main()

{

int n;

int arr[];

System.out.println("Enter size of the array");

int size=s.nextInt();

arr=new int[size];

System.out.println("Enter "+size+" numbers");

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

{

arr[i]=s.nextInt();

}

selectionSort(arr,0);

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

{

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

}

}

}

## 

## Q25. Write a program to input two points with x and y coordinates and find the distance between them.

class Point

{

int x;

int y;

public Point()

{

this.x = 0;

this.y = 0;

}

public Point(int x, int y)

{

this.x = x;

this.y = y;

}

public double distance(int x, int y) {

double d = Math.sqrt( Math.pow(this.x-x, 2) + Math.pow(this.y-y, 2) );

return d;

}

}

import java.util.\*;

class Distance

{

public static void main(String[] args){

Point first, second;

if(args.length==4){

first = new Point(Integer.valueOf(args[0]), Integer.valueOf(args[1]));

second = new Point(Integer.valueOf(args[2]), Integer.valueOf(args[3]));

}

else{

Scanner input = new Scanner(System.in);

System.out.println("Enter first point: ");

first = new Point(input.nextInt(),input.nextInt());

System.out.println("Enter second point: ");

second = new Point(input.nextInt(),input.nextInt());

}

double d = first.distance(second.x, second.y);

System.out.println("Distance between " +

"(" + first.x + "," + first.y + ")" +

" and " +

"(" + second.x + "," + second.y + ")" +

" is " + d);

System.out.println();

}

}

## Q26. Write a program to create a Double ended queue, method for insertion and deletion from both ends and a display method.

class dequeue

{

int s;

int dq[]=new int[s];

int f,r;

dequeue(int size)

{

dq=new int[size];

f=0;

r=-1;

}

void Insert\_front(int item)

{

if(f==0&&r==-1)

{

dq[f++]=item;

r++;

return;

}

else if(f>0)

{

dq[--f]=item;

return;

}

System.out.println("Insertion from front not possible");

}

void Insert\_rear(int item)

{

if(r!=(s-1))

{

dq[++r]=item;

return;

}

System.out.println("Insertion from rear not possible");

}

int Delete\_front()

{

int temp=dq[f];

f++;

return temp;

}

int Delete\_rear()

{

int temp1=dq[r];

r--;

return temp1;

}

void main()

{

for(int i=f;i<=r;i++)

{

System.out.println(dq[i]+" ");

}

}

}

## Q27. Write a program to create a linked list implemented as a queue, insert, delete and display elements in it.

public class QueueAsList

{

int data;

QueueAsList temp, link, front, rear;

QueueAsList()

{

data=0;

temp=link=front=rear=null;

}

void insert(int item)

{

QueueAsList temp=new QueueAsList();

temp.data=item;

temp.link=null;

if(front==null&&rear==null)

{

front=rear=temp;

}

else

{

rear.link=temp;

rear=temp;

}

}

void display()

{

QueueAsList temp=new QueueAsList();

temp=front;

if(temp!=null)

{

while(temp.link!=null)

{

System.out.print(temp.data+"--->");

temp=temp.link;

}

System.out.print(temp.data);

}

else

System.out.print("Queue underflow");

}

int delete()

{

if(front==null)

{

System.out.println("Queue underflow");

return -1;

}

int dat=front.data;

front=front.link;

return dat;

}

}

## Q28. Write a program to create a binary file which stores the marks obtained by students, read and write from the file.

import java.util.\*;

import java.io.\*;

class MarksWrite

{

public static void main()

{

String str;

int m1,m2,m3,roll;

Scanner s=new Scanner(System.in);

try

{

FileOutputStream fos=new FileOutputStream("marks.dat",true);

DataOutputStream dos=new DataOutputStream(fos);

int ctr=1;

System.out.println("Enter name, roll number and marks");

str=s.nextLine();

dos.writeUTF(str);

roll=s.nextInt();

dos.writeInt(roll);

m1=s.nextInt();

m2=s.nextInt();

m3=s.nextInt();

dos.writeInt(m1);

dos.writeInt(m2);

dos.writeInt(m3);

dos.close();

fos.close();

}

catch(IOException e)

{

System.err.println(e);

}

}

}

import java.util.\*;

import java.io.\*;

class MarksRead

{

public static void main()throws IOException

{

String str;

int m1,m2,m3,roll;

double p=0.0;

boolean eof=false;

Scanner s=new Scanner(System.in);

try

{

FileInputStream fis=new FileInputStream("marks.dat");

DataInputStream dis=new DataInputStream(fis);

while(eof!=true)

{

try

{

str=dis.readUTF();

roll=dis.readInt();

m1=dis.readInt();

m2=dis.readInt();

m3=dis.readInt();

System.out.println("Name: "+str+" Roll No: "+roll);

System.out.println("Marks1: "+m1+" Marks2: "+m2+" Marks3: "+m3);

p=((m1+m2+m3)\*100)/300;

System.out.println("Percentage: "+p);

}

catch(EOFException eo)

{

System.out.println();

System.out.println("END OF REPORT");

eof=true;

}

catch(IOException i)

{

System.err.println(i);

}

}

dis.close();

fis.close();

}catch(FileNotFoundException e)

{

System.err.println(e);

}

}

}

## Q28. Write a program to create and abstract class Employee and a subclass Programmer to calculate salary. Implement the concept of inheritance.

public abstract class Employee

{

String name;

int id;

Employee(String name, int id)

{

this.name=name;

this.id=id;

}

public abstract void calcSalary();

}

import java.util.\*;

public class Programmer extends Employee

{

Scanner s=new Scanner(System.in);

double bonus;

double salary;

Programmer(String name, int id, double salary)

{

super(name,id);

this.salary=salary;

bonus=0.0;

}

public void calcSalary()

{

System.out.println("Enter bonus");

bonus=s.nextDouble();

salary=salary+bonus;

System.out.println(salary);

}

}

## Q30. Write a program creating a super class Data and a subclass Process to print and count the words beginning with a vowel in a string.

class Data

{

protected String str;

Data()

{

str=" ";

}

void acceptstr(String st)

{

str=st;

}

void print()

{

System.out.println("String: "+str);

}

}

import java.util.\*;

class Process extends Data

{

int len=str.length();

void vowelWords()

{

String str1;

Scanner sc=new Scanner(System.in);

System.out.println("Enter a sentance");

acceptstr(sc.nextLine());

print();

str=str;

StringTokenizer s=new StringTokenizer(str," ");

char ch;

int ctr=0;

while(s.hasMoreTokens())

{

str1=s.nextToken();

ch=str1.charAt(0);

if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u')

{

System.out.print(str1+" ");

ctr++;

}

}

System.out.println();

System.out.println("Number of words starting with a vowel: "+ctr);

}

}