**JAVA CODES:**

// System.out.println("Hello World"); //ln - line space ln

// System.out.print("Hello World\n"); // /n- where we want we get line space

// System.out.print("Hello World");

// int a =10;

// String b = "hi prasanth";

// System.out.println(a);

// System.out.println(b);

// int a = sc.nextInt();

// float b = sc.nextFloat();

// float sum = a + b;

// System.out.println(sum);

// float r = sc.nextFloat();

// float area = 3.14f \* r \* r; //\*\*\*without specifying 'f' it takes it has double

// System.out.println(area);

// float num = sc.nextInt(); // \*\*\* Type conversion int to float

// System.out.println(num);

// float num1 = 98.9999; // \* Error

// float num = 98.999f; // f

// int num2 = (int) num; // Type Casting

// System.out.println(num2);

// float a = sc.nextFloat();

// float b = sc.nextFloat();

// float c = sc.nextFloat();

// float avg = (a+b+c)/3;

// System.out.println(avg);

// int a=5;

// int b=5;

// System.out.println("add" + (a+b));

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

// int a = sc.nextInt();

// // if(a>0)

// // {System.out.println("+");}

// // else if(a<0)

// // {

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

// // }

// // else {System.out.println("0");}

// switch(a)

// {

// case 1 : System.out.println("mon");

// break;

// case 2 : System.out.println("t");

// break;

// case 3 : System.out.println("w");

// break;

// case 4 : System.out.println("th");

// break;

// case 5 : System.out.println("fr");

// break;

// case 6 : System.out.println("sa");

// break;

// case 7 : System.out.println("s");

// break;

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

// }

—---------------------------------

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

// {

// System.out.println("Enter the range:");

// Scanner sc = new Scanner(System.in);

// int n = sc.nextInt();

// System.out.println("Primes in the range are:");

// primeinrange(n);

// }

—-------------------------------

// public static void primeinrange(int n)

// {

// for(int i=2;i<=n;i++)

// {

// if(isPrime(i))

// {

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

// }

// }

// }

—---------------------------------

// public static boolean isPrime(int n)

// {

// boolean isPrime = true;

// if(n==2)

// {

// return true;

// }

// for( int i=2;i<=Math.sqrt(n);i++)

// {

// if(n%i==0)

// {

// return false;

// }

// }

// return isPrime;

// }

// System.out.println("Enter the num u wanna change:");

// Scanner sc = new Scanner(System.in);

// int n =sc.nextInt();

// decbin(n);

// }

—-------------------------------

// public static void decbin(int n)

// {

// int rem=0;

// int bin=0;

// int pow =0;

// while(n>0)

// {

// rem = n%2;

// bin = (rem\*(int)Math.pow(10, pow))+bin;

// pow ++;

// n = n/2;

// }

// System.out.println(""+bin);

—-----------------------

// public static void palindrome(int n)

// {

// int rem =0;

// int nnum =0;

// int a=n;

// while(n>0)

// {

// rem = n%10;

// nnum = nnum\* 10+ rem;

// n=n/10;

// }

// if(nnum==a){

// System.out.println("yes it is palindrome"+nnum);}

// else{

// System.out.println("dengai");

// }

// }

—------------------------

// public static int sumofint(int n)

// {

// int rem =0;

// int sum=0;

// while(n>0)

// {

// rem=n%10;

// sum = sum +rem;

// n/=10;

// }

// return sum;

// }

—-------------------------------

public static void holl\_rect(int rows , int cols)

// {

// for(int i =1; i<=rows ; i++)

// {

// for (int j = 1; j<=cols ; j++)

// {

// if (i==1||i==rows||j==1||j==cols)

// {

// System.out.print("\*");

// }

// else{

// System.out.print(" ");

// }

// }

// System.out.println("");

// }

// }

\*\*\*\*\*\*

// public static void butterfly(int n)

// {

// for( int i=1; i<=n;i++)

// {

// for(int j=1;j<=i;j++)

// {

// System.out.print("\*");

// }

// for(int j=1;j<=(2\*(n-i));j++)

// {

// System.out.print(" ");

// }

// for (int j=1;j<=i;j++)

// {

// System.out.print("\*");

// }

// System.out.println();

// }

// for(int i=n;i>=1;i--)

// {

// for(int j=1;j<=i;j++)

// {

// System.out.print("\*");

// }

// for(int j=1;j<=(2\*(n-i));j++)

// {

// System.out.print(" ");

// }

// for (int j=1;j<=i;j++)

// {

// System.out.print("\*");

// }

// System.out.println();

// }

// }

—-----------------------------

//public static void rhombus(int n)

// {

// for(int i=1;i<=n;i++)

// {

// for (int j =1; j<=(n-i);j++)

// {

// System.out.print(" ");

// }

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

// {

// System.out.print("\*");

// }

// System.out.println();

// }

// }

—------------------------------

// public static void diamond(int n)

// {

// for (int i=1;i<=n;i++)

// {

// for(int j=1;j<=(n-i);j++)

// {

// System.out.print(" ");

// }

// for(int j=1;j<=(2\*i-1);j++)

// {

// {System.out.print("\*");}

// }

// System.out.println();

// }

// for(int i=n;i>=1;i--)

// {

// for(int j=1;j<=(n-i);j++)

// {

// System.out.print(" ");

// }

// for(int j=1;j<=(2\*i-1);j++)

// {

// System.out.print("\*");

// }

// System.out.println();

// }

// }

—--------------------------

// public static void pyramid(int n)

// {

// for(int i=1;i<=n;i++)

// {

// for(int j=1;j<=(n-i);j++)

// {

// System.out.print(" ");

// }

// for(int j=1;j<=i;j++)

// {

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

// }

// System.out.println();

// }

// }

—------------------------------

import java.util.\*;

// import java.lang.Math;

// public class Java\_Lec

// {

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

// {

// Scanner sc = new Scanner(System.in);

// System.out.println("Enter the space needed for the array:");

// int n = sc.nextInt();

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

// int num[]=new int[n];

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

// {

// num[i]= sc.nextInt();

// }

// Arraypair(num);

// }

—------------------------------

// public static void Arraypair(int num[]) {

// int total =0;

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

// {

// int crr= num[i];

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

// {

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

// total++;

// }

// System.out.println();

// }

// System.out.print("total=" +total);

// }

// }

—-------------------------------

// public static void bubblesort(int num[])

// {

// for (int turn=0;turn<num.length-1;turn++)

// {

// one swap element must be added

// for(int j=0;j<num.length-1-turn;j++)

// {

// if(num[j]>num[j+1])

// {

// swap

// int temp = num[j];

// num[j]=num[j+1];

// num[j+1]=temp;

// }

// }

// }

// }

—--------------------------

// public static void selectionsort(int num[])

// {

// for (int i=0;i<num.length-1;i++)

// {

// int minPos=i;

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

// {

// if(num[minPos]>num[j])

// greater than symbol for descending order

// {

// minPos = j;

// }

// //swap

// int temp = num[minPos];

// num[minPos]=num[i];

// num[i]=temp;

// }

// }

// }

—---------------------------------

// public static void insertionsort(int num[]) {

// for (int i=1;i<num.length;i++)

// {

// int curr = num[i];

// int prev = i-1;

// //Finding out the correct position to insert

// while(prev>=0 && num[prev]> curr) // greater than symbol for descending prder

// {

// num[prev+1]=num[prev];

// prev--;

// }

// //insertion

// num[prev+1]=curr;

// }

// }

—------------------------------

public static void Countingsort(int arr[] )

// {

// int largest = Integer.MIN\_VALUE;

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

// {

// largest = Math.max(largest,arr[i]);

// }

// int count[]=new int[largest+1];

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

// {

// count[arr[i]]++;

// }

// //sorting

// int j=0;

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

// {

// while(count[i]>0)

// {

// arr[j]=i;

// j++;

// count[i]--;

// }

// }

// }

—----------------------

import java.util.\*;

import java.lang.Math;

public class Java\_Lec

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the space needed for the array:");

int n = sc.nextInt();

int m = sc.nextInt();

int j;

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

int arr[][]=new int[n][m];

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

{

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

{

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

}

}

printSpiral(arr);

}

public static void printSpiral(int arr[] [])

{

int startRow=0;

int startCOl=0;

int endRow=arr.length-1;

int endCol=arr[0].length-1;

while(startRow<=endRow && startCOl<=endCol)

{

//top

for (int j = startCOl; j<=endCol; j++)

{

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

}

//right

for(int i =startRow+1;i<=endRow;i++)

{

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

}

//bottom

for(int j=endCol-1;j>=startCOl;j--)

{

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

}

//left

for(int i=endRow-1;i>=startRow+1;i--)

{

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

}

startCOl++;

startRow++;

endCol--;

endRow--;

}

}

}

—----------------------------

// public static int diagnolsum(int arr[] [])

// {

// // int sum =0 ;

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

// // {

// // for (int j=0;j<arr[02].length;j++)

// // if(i==j)

// // {

// // sum+=arr[i][j];

// // }

// // else if(i+j == arr.length- 1)

// // {

// // sum+=arr[i][j];

// // }

// // }

// // return sum; // not optimized and TC is n^2)

// int sum =0;

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

// {

// sum+=arr[i][i];

// if (i!=arr.length-i-13)

// sum+=arr[i][arr.length-1-i]; // i+j=n-1;

// }

// return sum;

// }

—-------------------

public static boolean sortedmatrixsearch(int arr[] [],int key)

// {

// int i =0 ;

// int j=arr[0].length-1;

// while (i < arr.length && j>=0)

// {if(arr[i][j]==key)

// {

// System.out.print("found key at("+i+","+j+")");

// return true;

// }

// else if (key<arr[i][j])

// j --;

// else

// i++;}

// System.out.println("key not found");

// return false;

// }

—----------------------

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

String name = sc.nextLine();

indi(name);

}

public static void indi(String name)

{

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

{

System.out.print(name.charAt(i));

}

}

—--------------------

public static boolean indi(String name)

// {

// int n=name.length();

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

// {

// if(name.charAt(i)==name.charAt(n-i-1))

// return true;

// }

// return false;

// }

—----------------------------------------

// public static void indi(String name)

// {

// int n=name.length();

// int sum=0,x=0,y=0;

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

// {

// if(name.charAt(i)=='N')

// {

// y = y+1; //y++

// }

// else if(name.charAt(i)=='S')

// {

// y = y-1; //y- -

// }

// else if(name.charAt(i)=='E')

// {

// x = x+1; // x++

// }

// else if(name.charAt(i)=='W')

// {

// x = x-1; //x--

// }

// }

// sum=(x\*x)+(y\*y);

// System.out.println(sum + " "+Math.sqrt(sum));

// }

—-------------------------------

public static void main(String[] args)

// {

// StringBuilder sb= new StringBuilder("");

// for (char ch='A';ch <='z';ch++)

// {

// sb.append(ch);

// }

// System.out.println(sb);

// }

—---------------------------------

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

// {

// System.out.println("enter the text");

// Scanner sc = new Scanner(System.in);

// String str =sc.nextLine() ;

// System.out.println( Upercase(str));

// }

// public static String Upercase(String str)

// {

// int n = str.length();

// StringBuilder sb= new StringBuilder();

// char ch = Character.toUpperCase(str.charAt(0));

// sb.append(ch); // for first character

// for( int i =1;i<n;i++)

// {

// if(str.charAt(i)==' '&& i < n-1)

// {

// sb.append(str.charAt(i));

// i++;

// sb.append(Character.toUpperCase(str.charAt(i))); // imp line

// }

// else

// {

// sb.append(str.charAt(i));

// }

// }

// return sb.toString();

// }

—----------------------------------

Recursion :

public static void main(String[] args)

{

int n=10;

printinc(n);

}

public static void printinc(int n)

{

if(n==1)

{

System.out.println(n);

return;

}

printinc(n-1); // imp \*\*\*

System.out.println(n); // just swap the step for reverse order

}

—-------------------------------

public static void main(String[] args)

{

int n=10;

System.out.println(printinc(n));

}

public static int printinc(int n)

{

if(n==0)

{

return 0 ;

}

int fn = n + printinc(n-1);

return fn ;

}

—-----------------------------

//not optimised

public static int printinc(int x,int n)

{

if(n==0)

{

return 1;

}

int prev=printinc(x,n-1);

int sum = x\*prev;

return sum;

}

—-------------------

//optimized o(log n)

public static int printinc(int x,int n)

{

if(n==0)

{

return 1;

}

int prev=printinc(x,n/2);

prev = prev\*prev;

if(n%2!=0)

{

prev=x\*prev;

}

return prev;

}

—-----------

// tiling problem \*\*\*\* very imp

public static int printinc(int n)

{

if(n==0||n==1)

return 1;

int fnm1 = printinc(n-1);

int fnm2 = printinc(n-2);

int totalways = fnm1+fnm2;

return totalways;

}

—---------------

// remove duplicate string

public class Java\_Lec

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

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

String str= sc.nextLine();

printinc(str,0, new StringBuilder(""),new boolean[26]);

}

public static void printinc(String str,int inx,StringBuilder st,boolean map[])

{

if(inx==str.length())

{

System.out.println(st);

return;

}

char curchar = str.charAt(inx);

if(map[curchar -'a']==true)

{

printinc(str, inx+1, st, map);

}

else

{

map[curchar-'a']=true;

printinc(str, inx+1, st.append(curchar), map);

}

}

—-------------------------

// pairing with friends

public static int printinc(int n)

{

if(n==1||n==2)

{

return n;

}

int fnm1= printinc(n-1);

int fnm2=printinc(n-2);

int pairways=(n-1)\*fnm2;

int totalways=fnm1+pairways;

return totalways;

}

—---------------------------

// binary string very imp\*\*\*\*\*\*\*\* pairing problem friend can sit or stand

public static void main(String[] args)

{

printinc(3,0,"");

}

public static void printinc(int n,int lastplace,String str)

{

if(n==0)

{

System.out.println(str);

return;

}

printinc(n-1, 0, str+"0");

if(lastplace==0)

printinc(n-1, 1, str+"1");

}

—-----------------------

// merge sort very imp \*\*\*\*\*\*\*

import java.util.\*;

import java.lang.Math;

public class Java\_Lec

{

public static void printa(int arr[])

{

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

{System.out.println(arr[i]);}

System.out.println();

return;

}

public static void mergesort(int arr[], int si , int ei)

{

if(si>=ei)

{return;}

int mid = si+(ei-si)/2;

mergesort(arr, si, mid);

mergesort(arr, mid+1, ei);

merge(arr,si,mid,ei);

}

public static void merge(int arr[],int si,int mid , int ei)

{

int temp[] = new int [ei-si+1];

int i =si;

int j= mid+1;

int k =0;

while (i<=mid &&j<=ei)

{

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

{

temp[k]=arr[i];

i++;

}

else

{

temp[k]=arr[j];

j++;

}

k++;

}

while (i<=mid)

{

temp[k++]=arr[i++];

}

while(j<=ei)

{

temp[k++]=arr[j++];

}

for(k=0,i=si;k<temp.length;k++,i++)

arr[i]=temp[k];

}

public static void main(String[] args)

{

int arr[]={2,5,8,20,56,45};

mergesort(arr, 0, arr.length-1);

printa(arr);

}

}

// Backtracking

public static void backtracksubset(String str , String ans,int i)

{

if(i==str.length())

{

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

return;

}

backtracksubset(str, ans+str.charAt(i), i+1);

backtracksubset(str, ans, i+1);

}

}

—-------------------------------------

//nqueens very imp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import java.util.\*;

import java.lang.Math;

public class Java\_Lec

{

public static void main(String[] args)

{

int n=5;

char board[][] = new char[n][n];

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

{

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

{

board[i][j]='x';

}

}

nqueens(board,0);

}

public static void nqueens(char board[][],int row)

{

if(row==board.length)

{

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

printarr(board);

return;}

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

{

if(issafe(board,row,i))

{board[row][i]='Q';

nqueens(board, row+1);

board[row][i]='x';}

}

}

public static boolean issafe(char board[][],int row,int col)

{

//vertical

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

{

if(board[i][col]=='Q')

return false;

}

//right dia

for (int i=row-1,j=col+1;i>=0&&j<board.length;i--,j++)

{

if(board[i][j]=='Q')

return false;

}

//left dia

for (int i =row-1, j=col-1;i>=0 && j>=0;i--,j--)

{

if(board[i][j]=='Q')

return false;

}

return true;

}

public static void printarr(char board[][])

{

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

{

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

{

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

}

System.out.println();

}

return;

}

}

—---------------------------------

// nqueens all possibilities

import java.util.\*;

import java.lang.Math;

public class Java\_Lec

{

public static void main(String[] args)

{

int n=2;

char board[][] = new char[n][n];

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

{

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

{

board[i][j]='x';

}

}

nqueens(board,0);

}

public static void nqueens(char board[][],int row)

{

if(row==board.length)

{

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

printarr(board);

return;}

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

{

board[row][i]='Q';

nqueens(board, row+1);

board[row][i]='x';

}

}

public static void printarr(char board[][])

{

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

{

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

{

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

}

System.out.println();

}

return;

}

}

—-----------------