**ASSINGMENT 03:**

**Question 01(by tail recursive):**

import java.util.\*;

public class Main {

public static void tail(int arr[],int x){

if (x==1){

return; }

for (int i=0;i<x-1;i++){

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

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

}

tail(arr,x-1);

}}

public static void main(String[] args) {

int arr[]={2,1,3,5,6,7};

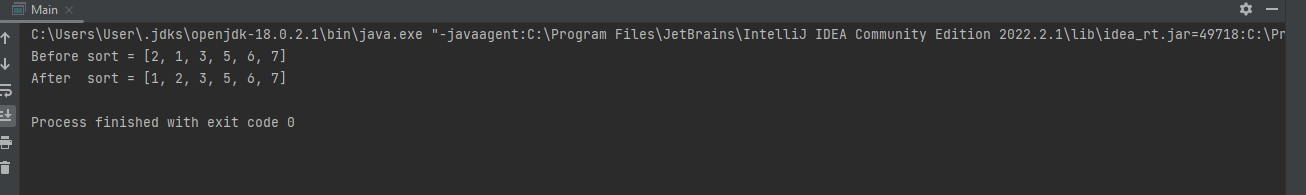
Arrays.sort(arr);

System.out.println("Before sort = [2, 1, 3, 5, 6, 7]");

System.out.println("After sort = "+Arrays.toString(arr));

}

}



**Question 01(by normal recursive ):**

import java.util.\*;

public class Main{

public static void normal(int arr[],int x){

if (x==1){

for (int i=0;i<x-1;i++){

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

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

}

normal(arr,x-1);

}

}

else{

return; }

}

public static void main(String[] args) {

int arr[]={2 ,1, 3, 5, 6, 7};

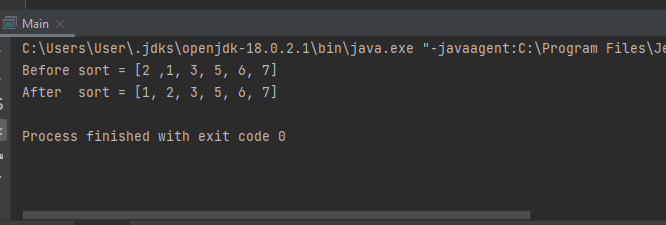
Arrays.sort(arr);

System.out.println("Before sort = [2 ,1, 3, 5, 6, 7]");

System.out.println("After sort = "+Arrays.toString(arr));

}

}



**Question 02:**

**RAT MAZE :**

public class Main{

final int N = 4;

void print(int sol[][]){

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

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

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

System.out.println();

}

}

boolean isSafe(int maze[][], int x, int y) {

return (x >= 0 && x < N && y >= 0 && y < N && maze[x][y] == 1);

}

boolean solveMaze(int maze[][]) {

int sol[][] = { { 0, 0, 0, 0,0 },

{ 0, 0, 0, 0,0 },

{ 0, 0, 0, 0,0},

{ 0, 0, 0, 0,0}

};

if (solveMaze1(maze, 0, 0, sol) == false) {

System.out.print("Solution doesn't exist");

return false;

}

print(sol);

return true;

}

boolean solveMaze1(int maze[][], int x, int y,int sol[][]){

if (x == N - 1 && y == N - 1) {

sol[x][y] = 1;

return true;

}

if (isSafe(maze, x, y) == true) {

sol[x][y] = 1;

if (solveMaze1(maze, x + 1, y, sol))

return true;

if (solveMaze1(maze, x, y + 1, sol))

return true;

sol[x][y] = 0;

return false;

}

return false;

}

public static void main(String args[]) {

Main rat = new Main();

int maze[][] = { { 1, 0, 0, 0},

{ 1, 1, 1, 1 },

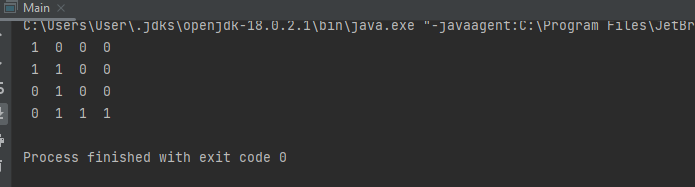
{ 0, 1, 0, 0 },

{ 0, 1, 1, 1 } };

rat.solveMaze(maze);

}

}



**NQUEENS :**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter the value of n");

int n=sc.nextInt();

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

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

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

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

if(solveNQueens(board,0,n)) display(board,n);

else System.out.println("No solution exists");

}

public static void display(char board[][], int n) {

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

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

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

System.out.println();

}

}

public static boolean isSafe(char board[][], int row, int column, int n) {

int i,j;

for(i=0;i<column;i++) {

if(board[row][i]=='Q') return false;

}

for(i=row,j=column; i>=0 && j>=0;i--,j--) {

if(board[i][j]=='Q') return false;

}

for(i=row,j=column; i<n && j>=0;i++,j--) {

if(board[i][j]=='Q') return false;

}

return true;

}

public static boolean solveNQueens(char board[][], int column, int n) {

if(column>=n) return true;

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

if(isSafe(board,i,column,n)) {

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

if(solveNQueens(board, column+1,n)) return true;

board[i][column]='-';

}

}

return false;

}

}

**Question 03:**

import java.util.\*;

public class Main{

static void fun1(int n) {

if (n!=16) {

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

func2(n +1);

}

}

static void func2(int n) {

if (n!=16) {

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

fun1(n + 1);

}

}

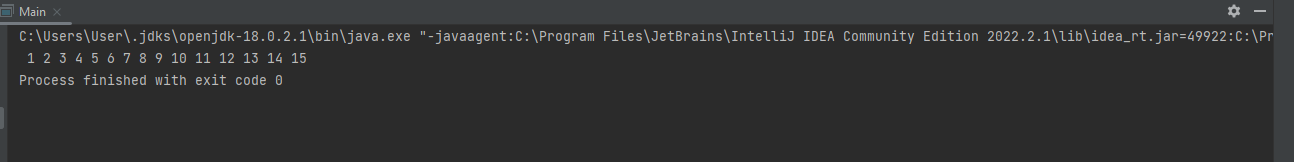
public static void main(String[] args) {

int n = 1;

fun1(n);

}

}



**Question 04(a):**

import java.util.\*;

public class Qno4Lab3 {

public static void generatesequence(int a,int b,int series) {

if(b==series) {

return; }

int c=a+b;

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

generatesequence(c, b+1,series);

}

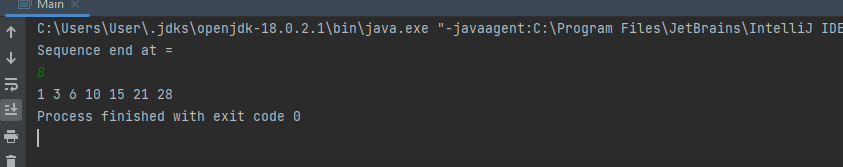
public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number to the end the series");

int series=sc.nextInt();

generatesequence(0, 1,series);

} } 

**Question 04(b):**

public class Qno4PartBLab3 {

public static void generatesequenceA(int a,int b,int series) {

if(b==series){

return; }

int c=a+b;

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

generatesequenceB(c, b+1,series);

}

public static void generatesequenceB(int c,int d,int series){

if(d==series){

return; }

int e=c+d;

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

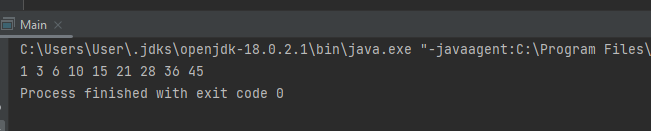
generatesequenceA(e, d+1,series);

}

public static void main(String[] args) {

generatesequenceA(0, 1, 10);

} }



**Question 05:**

public class Main{

public static void print(String s, int x, String y){

if(s.length()==0){

System.out.println(y);

return ;

}

int i=5;

while (s.length()>i){

char k=s.charAt(i);

String b= s.substring(0,i)+s.substring(i+1);

print(b,x+1,y+k);

}

}

public static void main(String[] args) {

String str = "FAST ";

print(str,0,"");

}

}