문제

## 문제

7  
 3 8  
 8 1 0  
 2 7 4 4  
4 5 2 6 5

위 그림은 크기가 5인 숫자 삼각형의 한 모습이다.

맨 위층 7부터 시작해서 아래에 있는 수 중 하나를 선택하여 아래층으로 내려올 때, 이제까지 선택된 수의 합이 최대가 되는 경로를 구하는 프로그램을 작성하라. 아래층에 있는 수는 현재 층에서 선택된 수의 대각선 왼쪽 또는 대각선 오른쪽에 있는 것 중에서만 선택할 수 있다.

삼각형의 크기는 1 이상 500 이하이다. 삼각형을 이루고 있는 각 숫자는 모두 정수이며, 범위는 0 이상 99 이하이다.

## 입력

첫째 줄에 삼각형의 크기 n(1≤n≤500)이 주어지고, 둘째 줄부터 n+1줄까지 숫자 삼각형이 주어진다.

## 출력

첫째 줄에는 최대가 되는 합을 출력한다.

## 예제 입력 복사

5  
7  
3 8  
8 1 0   
2 7 4 4  
4 5 2 6 5

## 예제 출력 복사

30

**answer - good**

import java.util.Scanner;

public class T\_1932 {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int size = scan.nextInt();

int[] input= new int[size];

int[] max=new int[size];

input[0] = scan.nextInt();

int check;

for(int i=1;i<size;i++){

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

check=scan.nextInt();

if(j==0){

max[j]=input[j]+check;

}else if(j==i){

max[j]=check+input[j-1];

}else{

max[j]=check +max(input[j-1],input[j]);

}

}

System.arraycopy(max, 0, input, 0, max.length);

}

int mx=0;

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

if(input[i]>mx){

mx=input[i];

}

}

System.out.println(mx);

}

public static int max(int x,int y){

return x>y ? x:y;

}

}

answer 2 - 입력과 출력 변경

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.util.Arrays;

import java.util.StringTokenizer;

public class T\_1932\_2 {

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

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

BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));

int size = Integer.parseInt(br.readLine());

int[] input= new int[size];

int[] max=new int[size];

input[0] = Integer.parseInt(br.readLine().trim());

int check;

StringTokenizer st;

for(int i=1;i<size;i++){

st= new StringTokenizer(br.readLine());

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

check= Integer.parseInt(st.nextToken());

if(j==0){

max[j]=input[j]+check;

}else if(j==i){

max[j]=check+input[j-1];

}else{

max[j]=check +max(input[j-1],input[j]);

}

}

System.arraycopy(max, 0, input, 0, max.length);

}

int mx=0;

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

if(input[i]>mx){

mx=input[i];

}

}

// bw.write(Arrays.toString(max));

bw.write(mx+"");

bw.flush();

bw.close();

}

public static int max(int x,int y){

return x>y ? x:y;

}

}