**Juan Sebastián Bernal Jiménez - 1022434946  
Guillermo Alejandro Cano Rodriguez - 1121888697  
Jhon Edison Prieto Artunduaga - 1007163058  
Jonatan Javier Valero Comayan - 1007491800**  
 **Estructuras de Datos – Grupo 3**  
 **4 de mayo de 2023**

**PARCIAL EJERCICIOS ESTRUCTURAS SECUENCIALES - UNCODE**

1. **Serpiente:**

import java.util.ArrayList;

import java.util.Scanner;

public class Main {

private static ArrayList<Integer> treeArr= new ArrayList<>();

public static int serpentSum(ArrayList<Integer> tree){

int sum=0;

int zig=0;

int eat=1;

int currInd=0;

int levTrack=0;

int prevLevInd=-1;

while(currInd<tree.size()){

for(int i=0;i<Math.pow(2,levTrack) && currInd<tree.size() && currInd>prevLevInd;i++){

if(eat==1) sum+=tree.get(currInd);

eat=(eat+1)%2;

if(i!=(int) Math.pow(2,levTrack)-1) {

if (zig == 0) currInd++;

else {

currInd--;

}

}

}

if(zig==0){

currInd= (currInd+1)\*2>=tree.size() ? tree.size()-1 : (currInd+1)\*2;

}

else {currInd=2\*currInd+1;}

zig=(zig+1)%2;

prevLevInd+=Math.pow(2,levTrack);

levTrack++;

}

return sum;

}

public static void main(String[] args) {

Scanner scanner= new Scanner(System.in);

String input= scanner.nextLine();

Scanner inputScan= new Scanner(input);

while(inputScan.hasNextInt()){

treeArr.add(inputScan.nextInt());

}

int cal=scanner.nextInt();

int sum=serpentSum(treeArr);

String output= sum>=cal ? "Sobrevive" : "Muere";

System.out.print(output);

}

}

1. **Máxima sabiduría**

import java.util.ArrayList;

import java.util.Scanner;

public class Main {

private static ArrayList<Integer> treeArr= new ArrayList<>();

public static Integer[] findMaxLev(ArrayList<Integer> tree){

Integer[] max=new Integer[2];

int levTrack, levSum, maxLev, currInd, maxSum;

maxSum=levTrack= levSum=maxLev=currInd =0;

while(currInd<tree.size()){

for(int i=0;i<Math.pow(2,levTrack) && currInd<tree.size();i++){

levSum += treeArr.get(currInd);

currInd++;

}

if(levSum>maxSum){

maxSum=levSum;

maxLev=levTrack;

}

levSum=0;

levTrack++;

}

max[0]=maxLev+1;

max[1]=maxSum;

return max;

}

public static void main(String[] args) {

Scanner scanner= new Scanner(System.in);

String zones= scanner.nextLine();

Scanner zonesScan= new Scanner(zones);

while(zonesScan.hasNextInt()){

treeArr.add(zonesScan.nextInt());

}

Integer[] max=findMaxLev(treeArr);

System.out.print(max[1]);

}

}

1. **¿ Digan en qué casos de prueba fallaron, cuál creen que era la causa y cómo la solucionaron?**

En nuestros casos de prueba, encontramos que los problemas más fáciles de corregir fueron los relacionados con "**presentation** **error**" y "**compilation** **error**". El primero se debía a la presencia de espacios o saltos de línea no deseados, mientras que el segundo se originaba por errores de indentación. A continuación, el problema de "wrong answer" era un poco más frecuente, pero no se trataba tanto de respuestas incorrectas como de la omisión de elementos debido a punteros nulos. Sin embargo, el problema más recurrente fue el de "**Runtime** **Error**". Como mencioné anteriormente, esto se debió al uso incorrecto de punteros dentro de las sentencias de control "for". Por otro lado, el recurso de casos de prueba nos ayudarían a resolver los problemas de manera más sencilla, sin embargo la aclaración del profesor contribuyó bastante, puesto que se trataba de un error sencillo.