

Imagen de Unique Path II Ejercicio1

Java

Autocomplete

```
1 class Solution {
2     public int uniquePathsWithObstacles(int[][] obstacleGrid) {
3         int m = obstacleGrid.length;
4         int n = obstacleGrid[0].length;
5         int temp[] = new int[n];
6         if (obstacleGrid[0][0] == 1)
7             return 0;
8         temp[0] = 1;
9
10        for (int i = 1; i < n; i++) {
11            temp[i] = 0;
12            if (temp[i-1] == 1 && obstacleGrid[0][i] == 0) {
13                temp[i] = 1;
14            }
15        }
16    }
17 }
```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase

Run Code Result

Debugger

Accepted

Runtime: 0 ms

Your input

[[0,0,0],[0,1,0],[0,0,0]]
[[0,1],[0,0]]

Output

2
1

Expected

2
1

Console

Use Example Testcases

Run

Imagen Book Shop Ejercicio 2

```
43 public static void main(String args[]) {
44     Scanner scanner = new Scanner(System.in);
45     //caso prueba
46     /*int p = scanner.nextInt();
47     int cap = scanner.nextInt();
48     int[] costo = new int[p];
49     int[] pag = new int[p];
50     for (int i = 0; i < p; i++) {
51         costo[i] = scanner.nextInt();
52         pag[i] = scanner.nextInt();
53     }
54     System.out.println(bookShop(cap, pag, costo));
55     */
56
57     int n=4, cap=10;
58     int [] costo= {4,8,5,3};
59     int[] pag= {5, 12 ,8,1 };
60     System.out.println("cantidad maxima de paginas:"+bookShop(cap, pag, costo));
61
62 }
```

<terminated> BookShopEjercicio2 [Java Application] C:\Users\MUNOZ\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.w

cantidad maxima de paginas:13

Imagen Number Long Ejercicio 3

re

☆ Premium

+

Java Autocomplete i {}

```
1 class Solution {
2     public int lengthOfLIS(int[] nums) {
3         int []aux=new int[nums.length]; //aux toma valores maximos de cadena creciente
4         aux[0]=1;
5         for(int i=1;i<nums.length;i++){
6             int cont=1;
7             for(int j=0;j<i;j++){
8                 //compara valores crecientes, si es menor aumentara contador,
9                 //y luego actualiza el auxiliar con el contador
10                if(nums[j]<nums[i]){
11                    cont=Math.max(cont,aux[j]+1);
12                }
13                aux[i]=cont;
14            }
15        }
16        return aux[nums.length-1];
17    }
18 }
```

testcase Run Code Result Debuqger

Accepted Runtime: 0 ms

Your input

[10,9,2,5,3,7,101,18]
[0,1,0,3,2,3]
[7,7,7,7,7,7,7]

Output

4
4
1

Expected

4
4
1

Imagen Rectangle Cutting Ejercicio 4

```

17 private static int rectangleCutt(int a, int b) {
18     int cont = 0;
19     int temp;
20     //entra si es un rectangulo mayor a 1x1
21     while(a+b>1) {
22         if(a==b)
23             return cont;
24         if(Math.min(a, b)==1) {
25             cont=cont+Math.abs(a-b);
26             a=0;b=0;
27         }
28         else { //restamos el max menos el minimo de a,b en lo que resulta (max-min)x(min) el nuevo rectangulo
29             temp=Math.min(a, b);
30             b=Math.max(a, b)-temp;
31             a=temp;
32             cont++;
33         }
34     }
35     return cont;
36 }
37 public static void main(String args[]) {
38     Scanner scanner = new Scanner(System.in);
39     //casos prueba
40     //int a=3,b=5;
41     //int a=10,b=8;
42     //int a=21,b=29;
43     //int a=13,b=21;
44     int a=21,b=28;
45     System.out.println("cortes max de un rectangulo de "+a+" y "+b+" es :"+rectangleCutt(a,b));
46 }
47

```

Imagen Maximal Square Ejercicio 5

Store LeetCode Challenge + GIVEAWAY! Premium

Java Autocomplete

```
1 class Solution {
2     public int maximalSquare(char[][] matrix) {
3         int n,m;
4         n=matrix.length;
5         m=matrix[0].length;
6         int[][] B=new int[n][m];
7         for(int i=0;i<n;i++) {
8             if(matrix[i][0]=='1')
9                 B[i][0]=1;
10            else
11                B[i][0]=0;
12        }
13        for(int j=0;j<m;j++) {
14            if(matrix[0][j]=='1')
```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input

```
[[["1","0","1","0","0"],["1","0","1","1","1"],["1","1","1","1","1"],["1","0","0","1","0"]],
[["0","1"],["1","0"]],
[["0"]]]
```

Output

```
4
1
0
```

Expected

```
4
1
0
```

Console Use Example Testcases Run Code