

Fundamentos de Algoritmia
Grados en Ingeniería Informática. Grupos A, C, DG, E
Examen Convocatoria Extraordinaria, 22 de junio de 2022.

Nombre: _____ Grupo: _____

Laboratorio: _____ Puesto: _____ Usuario de DOMjudge: _____

Normas de realización del examen

1. Debes programar soluciones para cada uno de los tres ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección <http://exacrc>.
2. Escribe comentarios que expliquen tu solución, justifiquen por qué se ha hecho así y ayuden a entenderla. Calcula la complejidad de todas las funciones que implementes.
3. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen. El nombre de usuario y contraseña que has estado utilizando durante la evaluación continua **no** son válidos.
4. Escribe tu **nombre y apellidos** en un comentario en la primera línea de cada fichero que subas al juez.
5. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayas realizado de cada ejercicio.

1. (2 puntos) Estamos de excursión y debemos atravesar una quebrada que tiene un río en el fondo: primero descendemos, luego cruzamos el río y por último ascendemos. Dado un vector cuyos valores representan la altura del punto correspondiente del camino, se desea obtener la longitud del tramo correspondiente a cruzar el río, teniendo en cuenta que los tramos de descenso y ascenso son estrictamente decreciente y creciente respectivamente, y que el tramo que corresponde a cruzar el río consta de uno o más valores iguales.

Se pide:

1. Utilizar la técnica de divide y vencerás para resolver el problema. Puedes utilizar las funciones que consideres necesarias.
2. Justificar adecuadamente el coste del algoritmo.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de dos líneas: en la primera se indica el número de elementos del vector y en la segunda sus valores. La entrada finaliza con el valor -1 .

El número de elementos del vector es mayor o igual que cero.

Salida

Para cada caso de prueba se escribe en una línea diferente la longitud del segmento del vector cuyos elementos son todos iguales al mínimo del vector.

Entrada de ejemplo

```
7
10 9 5 5 7 8 9
8
20 3 3 3 3 7 9 19
6
50 43 30 27 16 12
5
6 6 6 8 9
-1
```

Salida de ejemplo

```
2
4
1
3
```

2.(2 puntos) Dada una secuencia de números enteros positivos de longitud mayor que cero, queremos encontrar el tramo más largo de unos seguidos de doses. El tramo comenzará con un 1. En el tramo, el valor siguiente a un 1 debe ser un 2 y el valor siguiente a un 2 debe ser un 1. El tramo finalizará con un 2. Se pide:

1. Define un predicado $tramo12(v, p, q)$ que sea cierto si los valores entre p (incluido) y q (excluido) forman un tramo de unos seguidos de doses.
2. Utilizando el predicado anterior, especifica una función que reciba un vector con los valores de la secuencia y devuelva la longitud del tramo más largo de unos seguidos de doses.
3. Implementa la función anterior. El coste de la implementación debe ser lineal en la longitud del vector.
4. Indica un invariante y una función de cota que permita probar la corrección del algoritmo implementado y justifica que el coste del algoritmo es lineal en la longitud del vector.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de dos líneas. En la primera se indica el número n de elementos de la secuencia y en la siguiente se indican los valores de la misma. La entrada termina con una secuencia con cero elementos.

Salida

Para cada caso de prueba se escribe en una línea la longitud del tramo máximo.

Entrada de ejemplo

```
2
1 2
2
2 1
3
6 1 2
3
1 2 6
3
1 6 2
10
1 2 1 2 2 1 2 1 2 1
11
3 4 5 6 1 2 4 5 1 2 1
0
```

Salida de ejemplo

```
2
0
2
2
0
4
2
```

3. (3 puntos) Para un desfile de la Pasarela Neptuno, los n modelos tienen que elegir entre n trajes diferentes y combinarlos con n pares de zapatos. Cada prenda se utilizará exactamente una vez. Las preferencias de cada modelo por cada traje y par de zapatos son conocidas y se encuentran recogidas en dos matrices cuadradas $T[1..n, 1..n]$ y $Z[1..n, 1..n]$ de números positivos de forma que $T[i, j]$ representa la preferencia del modelo i por el traje j y análogamente para $Z[i, j]$. Los modelos son muy conscientes de su imagen y saben perfectamente que, por mucho que les guste un traje, no pueden combinarlo con cualquier par de zapatos por lo que la preferencia de un modelo i por una combinación compuesta por el traje j y los zapatos k viene dada por $T[i, j] * Z[i, k]$. Se pide:
1. Escribir un algoritmo que utilice vuelta atrás para encontrar la asignación de trajes y zapatos a los modelos de modo que se maximice la suma de las preferencias de todos los modelos con la combinación asignada. Describe claramente el árbol de exploración y los marcadores utilizados.
 2. Utilizar estimaciones para podar el árbol de búsqueda siempre que sea posible. Para ello puede ser necesario precalcular algunos valores antes de comenzar la exploración.

Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá en una línea el número de modelos (que coincide con el número de trajes y pares de zapatos). A continuación, en una matriz de dimensión $n \times n$ se indica el valor de la preferencia de cada modelo por cada traje. Por último, en otra matriz de dimensión $n \times n$ se indica el valor de la preferencia de cada modelo por cada par de zapatos.

Salida

Para cada caso de prueba, el programa escribirá el valor que maximiza las sumas de las preferencias de todos los modelos con la combinación asignada.

Entrada de ejemplo

```
2
2
1 2
3 3
2 1
1 6
4
1 2 3 4
4 5 6 7
7 8 9 10
10 11 12 13
3 3 3 3
3 3 3 3
3 3 3 3
3 3 3 3
```

Salida de ejemplo

```
22
84
```