

ProblemasControl22021.pdf



pablofa02



Fundamentos de la Programación



1º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Máster

Online en Ciberseguridad

Nº1 en España según El Mundo



**Hasta el 46%
de beca**



Mejor Máster
según el
Ranking de
ELMUNDO

Para ser el mejor hay que aprender
de los mejores.

IMEF

Smart Education

Deloitte.

Infórmate

Que no te escriban poemas de amor
cuando terminen la carrera ▶▶▶▶▶▶▶▶▶▶



WUOLAH

la nosotros por suerte nos pasa)

No si antes decirte
Mañana mi diploma y título he de
pagar

Pero me voy a graduar.
Mañana mi diploma y título he de
pagar

Llegó mi momento de despedirte
Tras años en los que has estado mi
lado.

Siempre me has ayudado
Cuando por exámenes me he
agobiado

Oh Wuolah wuolah
Tu que eres tan bonita



UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y CC. Computación
E.T.S.I. Informática

Fundamentos de la Programación
Control de Evaluación Nº 2

Apellidos, Nombre:

Código PC usado:

11/01/2021

Grupo:

NOTAS PARA LA REALIZACIÓN DEL EXAMEN:

- La solución se almacenará en la carpeta **CONTROL2FP** dentro de **Documentos**. Si la carpeta ya existe, debe borrarse todo su contenido. En otro caso, debe crearse.
- Los nombres de los ficheros con la solución para los ejercicios serán **Ej1.cpp** y **Ej2.cpp**.
- Al inicio del contenido de cada fichero deberá aparecer un comentario con **el nombre del alumno, titulación, grupo y código del equipo** que se está utilizando (cada dato en una línea diferente).
- **Debe consultarse** el documento **"Obligaciones y Recomendaciones Estilo de Programación"**, disponible en el Campus Virtual de la asignatura, con objeto de tener en cuenta los puntos allí señalados en las soluciones a los ejercicios.
- Una vez terminado el examen, se subirán los ficheros ***.cpp** a la tarea creada en el **campus virtual** para ello.
- **No está permitido:**
 - Utilizar documentación electrónica o impresa.
 - Intercambiar documentación con otros compañeros.
 - Utilizar soportes de almacenamiento.
 - Utilizar dispositivos electrónicos (móviles, tablets, ...)

1. Dada una colección de números enteros, se dice que un número de la misma es un *número destacado* si es mayor al primer valor anterior a él en la colección que sea diferente a él (si lo hay) y también es mayor al primer valor posterior a él en la colección que sea diferente a él (si lo hay). Si, para un número, no existe ese valor anterior o posterior diferente, el número no es destacado. Es obvio que tanto el primero como el último de la colección no pueden ser números destacados. Ejemplos:

2, 3, 4, 4, 4, 4	Tiene 0 destacados
5, 3, 2, 1, 2, 3	Tiene 0 destacados
1, 2, 5, 2, 1, 0	Tiene 1 destacado (el 5)
1, 2, 3, 3, 3, 1	Tiene 3 destacados (los 3 elementos de valor 3)
1, 4, 2, 9, 9, 0	Tiene 3 destacados (el 4 y los dos 9)
3, 7, 5, 6, 4, -1	Tiene 2 destacados (el 7 y el 6)

Diseña un programa que lea de teclado una colección de MAX (una constante definida, por ejemplo con valor 6) números enteros separados por espacios en blanco y muestre por pantalla cuántos *números destacados* tiene (no es necesario decir cuáles son, sólo cuántos hay). Prueba el funcionamiento del mismo con los ejemplos anteriores. Por ejemplo, una ejecución del programa sería:

Introduzca una coleccion de 6 numeros enteros: 1 2 3 3 3 1

En esa coleccion hay 3 numeros destacados

WUOLAH

2. Dos listas de números enteros se dice que son *encajables* si, independientemente de sus elementos negativos, todos los números positivos de ambas se encuentran en el mismo orden relativo (en las listas no hay valores 0). Si no hay ningún número positivo en ambas listas, también son *encajables*.

Por ejemplo, las listas 1, -3, 14, 1, -24 y -7, 1, 14, -4, -8, -6, 1 son *encajables*, mientras que las listas 1, -3, 1, 14, -24 y -7, 1, 14, -4, 1, -6 no lo son. Tampoco lo son las listas 1, -3, 1, 14, -24, -7 y 1, 1, -4, -8, -6, 14, 7.

Define un tipo de datos (TLista) para poder almacenar una lista de hasta un máximo de MAX (una constante definida) números naturales. El tipo debe definirse de forma que se pueda controlar en todo momento la cantidad de números almacenados.

Diseña un programa que:

Lea por teclado dos listas de números enteros y las almacene en dos variables (lista1 y lista2) del tipo previamente definido TLista. Para introducir cada lista, el usuario dará una secuencia de números enteros separados por espacios en blanco y terminará la misma introduciendo un 0. Si el usuario introduce más de MAX números enteros antes de introducir el 0, sólo se almacenarán los MAX primeros (los números restantes de la entrada (incluido el 0) se descartarán (leyéndolos y no haciendo nada con ellos)).

Tras leer las dos listas, el programa mostrará por pantalla su contenido e indicará si las dos listas son *encajables* o no. Para la comprobación de si las dos listas son o no *encajables* no se podrán utilizar más variables de tipo estructurado además de las dos listas.

Ejemplos de ejecución del programa para una constante definida MAX = 5:

```
Introduzca la primera lista: 1 -3 1 14 0
Introduzca la segunda lista: -7 1 14 -4 1 -6 0
lista1 = 1 -3 1 14
lista2 = -7 1 14 -4 1
Las listas NO son encajables
```

```
Introduzca la primera lista: 1 -3 1 14 0
Introduzca la segunda lista: -7 1 1 -4 14 -6 0
lista1 = 1 -3 1 14
lista2 = -7 1 1 -4 14
Las listas SI son encajables
```

```
Introduzca la primera lista: 1 -3 1 14 3 0
Introduzca la segunda lista: -7 1 1 -4 14 -6 0
lista1 = 1 -3 1 14 3
lista2 = -7 1 1 -4 14
Las listas NO son encajables
```