



UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y Ciencias de la Computación
E.T.S.I. Informática

Fundamentos de la Programación
Examen 1ª Convocatoria Ordinaria

05/02/16

Apellidos, Nombre:
Titulación:
Código PC usado:

Grupo:

NOTAS PARA LA REALIZACIÓN DEL EXAMEN:

- La solución se almacenará en el directorio **C:\FP\EXAMENFEB**. Para ello, al arrancar Eclipse, debe establecerse este directorio como Espacio de Trabajo (Workspace).
- El nombre de los proyectos serán **prEj1, prEj2, prEj3, prEj4** para los ejercicios 1, 2, 3 y 4 respectivamente.
- Los nombres de los ficheros con la solución serán **ejercicio1.cpp, ejercicio2.cpp, ejercicio3.cpp y ejercicio4.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).
- 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.

(1 pto) 1.- **Diseña un procedimiento** que busque un elemento dentro de una matriz. Estos dos datos los recibe como parámetros. El procedimiento devolverá como parámetros de salida los índices de la fila y de la columna de la matriz donde está el elemento o los valores -1 en caso de no estar almacenado en la misma.

Importante: sólo se completará el código del procedimiento *buscar* en el fichero *ejercicio1.cpp* proporcionado en el campus virtual. No se debe modificar el resto del código proporcionado para probar la función buscar. La puntuación de este problema será de 1 punto sólo en el caso de que la búsqueda funcione correctamente y se haga de forma **eficiente**. En otro caso la puntuación será de 0 puntos.

La ejecución del código suministrado una vez diseñada la función *buscar* será:

```
Mostramos el contenido de la matriz proporcionada:
2 7 12 89
5 2 9 10
10 está en la fila 1, columna 3 de la matriz
2 está en la fila 0, columna 0 de la matriz
27 no se encuentra en la matriz
```

(2 ptos) 2.- **Diseña un procedimiento** *ordenar* que realice una ordenación de menor a mayor de una lista de números enteros almacenados en el array unidimensional que recibe como parámetro. Para llevar a cabo dicha ordenación se seguirá el siguiente proceso (método de Selección):

- Buscar el mínimo elemento de la lista e intercambiarlo con el primero.
- Buscar el siguiente mínimo elemento del resto de la lista e intercambiarlo con el segundo.

- Buscar el siguiente mínimo elemento del resto de la lista e intercambiarlo con el tercero.
- Así sucesivamente.

En general: Buscar el mínimo elemento entre una posición i y el final de la lista e intercambiarlo con el elemento de esa posición i . El valor de i comienza en la primera posición y se aumenta de uno en uno.

Importante: sólo se completará el código del procedimiento *ordenar* en el fichero *ejercicio2.cpp* proporcionado en el campus virtual. Se podrán añadir más procedimientos y funciones si son necesarios para la solución de *ordenar*.

La ejecución del código suministrado una vez diseñado el procedimiento *ordenar* será:

El array antes de ordenarlo: 2 7 12 89 5 2 9 10
El array después de ordenarlo: 2 2 5 7 9 10 12 89

(3.5 pts) 3.- Diseña un algoritmo que lea de teclado una cadena de caracteres (patrón) y un texto, y muestre por pantalla un listado de todas las palabras del texto que contengan la cadena patrón de “forma relativa”. Esto significa que la palabra contendrá todas las letras de la cadena patrón en el mismo orden, pero no tienen que estar dispuestas juntas. **En la salida no habrá palabras repetidas.**

Ejemplo:

Entrada:

Introduzca el patrón: EN

Introduzca el texto (FIN para terminar):

CREO QUE IREMOS A LA DIRECCION QUE NOS DIERON Y
TAMBIEN IREMOS A LA DIRECCION NUEVA QUE YO CONOCIA
FIN

Salida:

DIRECCION DIERON TAMBIEN

NOTAS:

- El texto contiene un número indefinido de palabras.
- El texto termina con la palabra FIN.
- Cada palabra tiene un número indefinido pero limitado de caracteres (todos alfabéticos mayúsculas).
- En el texto aparecerán un número máximo `MAX_PAL_DIST` (una constante) de palabras distintas.
- El carácter separador de palabras es el espacio en blanco.

(3.5 pts) 4.- Diseña un procedimiento coincidencias que reciba como parámetros de entrada dos cadenas de caracteres *cadena1* y *cadena2*, que contienen exclusivamente letras mayúsculas. El procedimiento debe devolver, como parámetros de salida, dos valores: cuantos caracteres de *cadena2* se encuentran en *cadena1* en la misma posición; y cuantos caracteres de *cadena2* aparecen en *cadena1* en una posición diferente.

Cada carácter se contabilizará una vez, esto es, si hay un carácter repetido en la primera cadena que sólo aparece una vez en la segunda se contabilizará una vez o, de forma simétrica, si hay un carácter en la primera cadena que aparece más de una vez en la segunda también se contabilizará una vez. *Siempre tendrá prioridad la coincidencia de posición* (ver ejemplo cuarto más abajo). Si se considera necesario,

pueden utilizarse estructuras de datos auxiliares para controlar este tema de las repeticiones.

Ejemplos para probar el funcionamiento correcto del procedimiento pedido:

- Para las cadenas "ARXMBA" y "RZXMKJH" que tienen 2 caracteres en la misma posición ('X' y 'M') y un carácter en posición diferente ('R'), el procedimiento devolvería los valores 2 y 1.
- Para las cadenas "JAKZEF" y "AOA" el procedimiento devolvería los valores 0 y 1 ya que no tienen ningún carácter en la misma posición y un carácter en posición diferente ('A').
- Para las cadenas "JAKZAF" y "AA" que tienen un carácter en la misma posición ('A') y un carácter en posición diferente ('A'), el procedimiento devolvería 1 y 1.
- Para las cadenas "EA" y "AA" que tienen un carácter en la misma posición ('A') y ningún carácter en posición diferente, devolvería 1 y 0.

Diseña también el **algoritmo principal** que lea dos cadenas por teclado, llame al procedimiento *coincidencias* y muestre por pantalla los valores devueltos por él.