



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

Fundamentos de la Programación
Examen 2ª Convocatoria Ordinaria

07/09/18

Apellidos, Nombre:

Titulación:

Grupo:

Código PC usado:

NOTAS PARA LA REALIZACIÓN DEL EXAMEN:

- La solución se almacenará en la carpeta **EXAMENSEPTFP**, 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 1, 2, 3 y 4 serán **ejercicio1.cpp**, **ejercicio2.cpp**, **ejercicio3.cpp** y **ejercicio4.cpp**, respectivamente.
- 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.
 - Utilizar dispositivos electrónicos (móviles, tablets, ...)

(1 pto) 1.- **Diseña** una **función** *ordenado* que recibe como parámetro un array completo de tamaño TAM (una constante establecida) de valores enteros y devuelve un valor booleano indicando si el array está o no ordenado correctamente de menor a mayor valor.

Importante: sólo se completará el código de la función *ordenado* en el fichero *ejercicio1.cpp* proporcionado en el campus virtual. Puedes añadir más procedimientos o funciones si lo estimas necesario. No se debe modificar el resto del código proporcionado. 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 (tras diseñar la función solicitada) será:

El primer array SI esta ordenado El segundo array NO esta ordenado

(2 ptos) 2.- Diseña un algoritmo que lea de teclado 3 listas (*lista1*, *lista2*, *lista3*) de números enteros. Para cada lista, se leerán valores hasta encontrar un 0 (que actúa como terminador y no forma parte de la lista) o hasta que se hayan leído TAM (una constante establecida, por ejemplo 10) valores. En cada lista, los elementos repetidos se desechan (no cuentan y no se almacenan). Una vez leídas las tres listas, el algoritmo mostrará por pantalla en primer lugar los valores de dichas listas y después los posibles tríos de números (*num1* de la *lista1*, *num2* de la *lista2*, *num3* de la *lista3*) que satisfagan la relación $num1+num2=num3$.

Ejemplo (para TAM = 10):

```
Entrada:
    Introduzca Lista 1: 3 -2 3 5 -2 0
    Introduzca Lista 2: 6 3 4 18 0
    Introduzca Lista 3: 2 4 3 5 8 6 13 9 1 38 7 14 0

Salida:
    Lista 1: 3 -2 5
    Lista 2: 6 3 4 18
    Lista 3: 2 4 3 5 8 6 13 9 1 38
    Los trios de numeros son:
    3 6 9
    3 3 6
    -2 6 4
    -2 3 1
    -2 4 2
    5 3 8
    5 4 9
```

(3.5 ptos) 3.- El denominado *juego de la vida* fue ideado por J.H. Conway en 1969. Aunque parezca un juego, realmente es un modelo de una población de seres vivos idealizados que interactúan con su entorno. Los seres vivos se encuentran distribuidos en una matriz cuadrada de TAM (una constante establecida) filas y columnas que representa su mundo. En un instante dado, cada casilla de la matriz puede estar ocupada por un único ser vivo o vacía. La simulación comienza con una generación inicial. Cada nueva generación se obtiene a partir de la generación anterior atendiendo a las siguientes reglas:

- Si en una generación una determinada casilla está vacía, en la siguiente generación nacerá un ser vivo en la casilla correspondiente si el número de seres vivos vecinos (arriba, abajo, derecha, izquierda y diagonales) es igual a 3.
- Si en una generación una determinada casilla está ocupada por un ser vivo, en la siguiente generación ese ser vivo permanecerá en la casilla correspondiente si el número de seres vivos vecinos es igual a 2 o 3. En otro caso, ese ser vivo morirá, por lo que la casilla correspondiente en la siguiente generación permanecerá vacía.

Diseña un algoritmo que lea de teclado:

- el número de generaciones que se desean mostrar (incluida la inicial)
- y los datos para completar una matriz de tamaño TAM x TAM que constituye la generación inicial. El carácter 'o' representará una casilla vacía y el carácter 'x' representará un ser vivo.

Posteriormente mostrará por pantalla la generación inicial introducida e irá obteniendo y mostrando las sucesivas generaciones hasta completar el número de generaciones solicitadas.

Ayuda: utiliza dos matrices para almacenar una generación y la siguiente.

Nota: Se recomienda no utilizar el tipo `unsigned`, sino el tipo `int` para trabajar con valores de tipo entero y natural.

Ejemplo (para TAM = 5):

```
Entrada:
    Introduzca numero de generaciones: 3
    Introduzca generacion inicial:
    ooxoo
    oxoox
    xooxx
    ooxoo
    xoooo

Salida:
    Generacion 1 (inicial) :
    ooxoo
    oxoox
    xooxx
    ooxoo
    xoooo
    Generacion 2:
    ooooo
    oxxox
    oxxxx
    oxoxo
    ooooo
    Generacion 3:
    ooooo
    oxoox
    xooox
    oxoxx
    ooooo
```

(3.5 ptos) 4.- Se dice que dos palabras son locogramas si tienen la misma longitud y contienen las mismas letras, independientemente de que se repitan el mismo número de veces en ambas palabras. Así, “cursar” y “surcas” son locogramas, porque tienen las mismas letras, aunque la letra ‘r’ aparece 2 veces en la primera y 1 vez en la segunda, mientras que la letra ‘s’ aparece 1 vez en la primera y 2 veces en la segunda. En cambio, “sacas” y “cosas” no son locogramas, porque no tienen las mismas letras.

Diseña un algoritmo que lea un texto de teclado y muestre por pantalla aquellas palabras que son locograma de la primera palabra que aparece dentro de dicho texto.
En la salida no habrá palabras repetidas.

Ejemplo:

Entrada:

Introduzca un texto (FIN para terminar): SACAS LAS COSAS DE TUS CASAS
Y LUEGO ME DICES CASAS Y CCSAS SSSAS SIN SENTIDO FIN

Salida:

Las palabras que son locogramas de la primera son:
CASAS
CCSAS

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 habrá un número máximo MAX_PAL_DIST (una constante) de palabras distintas.
- El carácter separador de palabras es el espacio en blanco.