



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

Fundamentos de la Programación
Examen 2ª Convocatoria Ordinaria
Problemas

15/09/21

Apellidos, Nombre:
Titulación:

Grupo:

NOTAS PARA LA REALIZACIÓN DEL EXAMEN:

- El alumnado debe haber **comprobado** si cumple los requisitos para acogerse a la modalidad de **evaluación continua**. De no ser así, además de la prueba práctica, deberá realizar un examen teórico adicional y, por lo tanto, ponerse en contacto con el profesorado.
- La solución se almacenará en la carpeta **FPEXAMENSEPT**, 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 y 3 serán **ejercicio1.cpp**, **ejercicio2.cpp** y **ejercicio3.cpp**, respectivamente.
- Al inicio del contenido de cada fichero deberá aparecer un comentario con **el nombre del alumno o de la alumna, 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 compañeros/as.
 - Utilizar soportes de almacenamiento.
 - Utilizar dispositivos electrónicos (móviles, tablets, ...)

(2 ptos) 1.- Vamos a trabajar con matrices cuadradas de tamaño $MAX * MAX$, siendo MAX una constante cualquiera.

Define (en el código suministrado en el Campus Virtual “ej1.cpp”) el tipo **TMatriz** y **diseña** un **procedimiento** denominado **rotar** que recibe como parámetros una matriz de tipo **TMatriz** y un número entero n . El procedimiento modificará la matriz de forma que contenga al final el resultado de rotar la misma hacia la derecha un número n de veces. Se pueden utilizar estructuras de datos auxiliares y diseñar más procedimientos/funciones auxiliares.

Ejemplo:

Dada la siguiente matriz y $n = 2$:

1	6	7	9
2	5	15	16
8	7	1	4
3	12	11	10

Después de una rotación quedaría:

3	8	2	1
12	7	5	6
11	1	15	7
10	4	16	9

Después de dos rotaciones quedaría:

10	11	12	3
4	1	7	8
16	15	5	2
9	7	6	1

(4.5 pts) 2.- Un número natural n se dice que es triangular cuando su valor se puede obtener mediante la suma de unos cuantos números naturales consecutivos, comenzando con el número 1. Por ejemplo, el número 15 es triangular porque $15 = 1+2+3+4+5$.

El juego de cartas llamado “El solitario Búlgaro” consiste en lo siguiente. Consideremos un mazo de n cartas, siendo n un número triangular. Se reparte la totalidad de las cartas en un número arbitrario de montones, cada uno de ellos con una cantidad arbitraria de cartas. El lote de montones se puede reorganizar así:

Se toma una carta de cada montón y con todas ellas se forma un nuevo montón, que se agrega al final de la lista de montones. Los montones que tenían una única carta desaparecen.

Si repetimos esta operación con los montones que van quedando, se asegura que finalmente obtendremos una disposición tal que el primer montón tendrá 1 carta, el segundo 2 cartas, el tercero 3 cartas y así sucesivamente hasta el último.

Por ejemplo, partiendo de un mazo de 15 cartas repartidas en tres montones con 5, 7 y 3 cartas cada uno, las reorganizaciones sucesivas evolucionan como sigue:

5 7 3
4 6 2 3
3 5 1 2 4
2 4 1 3 5
1 3 2 4 5
2 1 3 4 5
1 2 3 4 5

Diseña un **algoritmo** (la función **main** y los procedimientos y las funciones que sean necesarios) que lea de teclado el número de montones inicial y el número de cartas de cada montón (se puede suponer que los valores serán correctos, tal como se indica en las Notas más adelante). El algoritmo llevará a cabo el proceso descrito anteriormente e irá mostrando por pantalla el contenido de los montones en cada paso hasta finalizar el solitario (ver ejemplo anterior).

Notas:

- Podemos suponer que el número de cartas del que se parte (repartidas entre los montones iniciales) es un número triangular correcto.
- Podemos suponer que en ningún momento el número de montones será mayor de una constante $MAX = 20$.

(3.5 ptos) 3.- Diseña un algoritmo (la función **main** y los procedimientos y las funciones que sean necesarios) que lea de teclado una cadena de caracteres (**patron**), un número entero (**x**) y un texto, y muestre por pantalla un listado de todas las palabras del texto que contengan al menos **x** caracteres en común con la cadena **patron**. **En la salida no habrá palabras repetidas.**

Ejemplo:

Entrada:

```
Introduzca el patron: ALICIA
Introduzca el valor de x: 2
Introduzca el texto (FIN para terminar):

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

Salida:

```
LA DIRECCION TAMBIEN CONOCIA
```

Ya que LA contiene la L y la A, DIRECCION contiene la C y la I, TAMBIEN contiene la A y la I y CONOCIA la A y la I

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 (con valor igual a 10) de palabras distintas que cumplan la condición especificada.
- El carácter separador de palabras es el espacio en blanco y/o nueva-línea (salto-de-línea).