



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

Fundamentos de la Programación
Examen 2ª Convocatoria Ordinaria

02/09/16

Apellidos, Nombre:
Titulación:
Código PC usado:

Grupo:

NOTAS PARA LA REALIZACIÓN DEL EXAMEN:

- La solución se almacenará en el directorio **C:\FP\EXAMENSEPT**. 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 una función que determine si una matriz cuadrada de números enteros recibida como parámetro es una matriz identidad. La función devolverá *true* si la matriz es identidad y *false* en otro caso. Una matriz cuadrada es identidad si todos los elementos de la diagonal principal son 1 y el resto de elementos son 0.

Importante: sólo se completará el código de la función *identidad* en el fichero *ejercicio1.cpp* proporcionado en el campus virtual. No se debe modificar el resto del código proporcionado para probar la función. La puntuación de este problema será de 1 punto sólo en el caso de que el algoritmo 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 *identidad* será:

```
Mostramos el contenido de una matriz identidad:
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
La matriz mostrada SI es una matriz identidad

Mostramos el contenido de una matriz NO identidad:
1 0 5 0
0 1 0 0
0 0 1 0
0 0 0 1
La matriz mostrada NO es una matriz identidad
```

(2 ptos) 2.- Un conjunto es una colección de elementos no repetidos. **Se ha definido el tipo *Conjunto*** que permitirá almacenar hasta un máximo de MAX números naturales, siendo MAX una constante dada. **Se ha diseñado un programa** (*ejercicio2.cpp*, ofrecido en el campus virtual) para trabajar con conjuntos de números naturales. Además de la definición de dicho tipo, aparece el algoritmo principal (*main*) y dos subalgoritmos de lectura/escritura (*leer* y *escribir*). Para **completar el programa**, debes realizar los **dos** siguientes **procedimientos** (se podrán añadir más procedimientos y funciones si son necesarios para la solución):

- *calcularInterseccion* que recibe como parámetros de entrada dos conjuntos y devuelve como parámetro de salida el conjunto resultante de efectuar la intersección entre ambos. La intersección de dos conjuntos es un nuevo conjunto formado por aquellos elementos que pertenecen a ambos conjuntos.
- *calcularUnion* que recibe como parámetros de entrada dos conjuntos y devuelve como parámetro de salida el conjunto resultante de efectuar la unión entre ambos. La unión de dos conjuntos es un nuevo conjunto formado por aquellos elementos que pertenecen a alguno de los dos conjuntos (o a ambos). La operación tendrá otro parámetro de salida de tipo booleano, por el que devolverá true si la operación se ha podido realizar con éxito y false si no se ha podido realizar porque no hay cabida para almacenar todos los elementos que formarían la unión.

Ejemplo 1 de ejecución:

```
Primer conjunto:
Introduzca el numero de elementos a leer (<= 10): 5
Introduzca 5 numeros naturales diferentes: 4 2 6 15 9

Segundo conjunto:
Introduzca el numero de elementos a leer (<= 10): 6
Introduzca 6 numeros naturales diferentes: 6 3 4 26 1 2

Interseccion:
Los elementos del conjunto son: 4 2 6

Union:
Los elementos del conjunto son: 4 2 6 15 9 3 26 1
```

Ejemplo 2 de ejecución:

```
Primer conjunto:
Introduzca el numero de elementos a leer (<= 10): 5
Introduzca 5 numeros naturales diferentes: 4 2 6 15 9

Segundo conjunto:
Introduzca el numero de elementos a leer (<= 10): 8
Introduzca 8 numeros naturales diferentes: 1 10 2 7 3 8 11 17

Interseccion:
Los elementos del conjunto son: 2
Union:
La Union no se ha podido realizar
```

(3.5 ptos) 3.- Diseña un **procedimiento** *desplazar* que reciba como parámetros de entrada dos cadenas de caracteres *cadena1* y *cadena2*, que contienen exclusivamente letras minúsculas. El procedimiento debe devolver, como parámetros de salida, dos valores: el menor desplazamiento de *cadena2* sobre *cadena1* para conseguir que coincidan el mayor número de caracteres y cuantos caracteres coinciden para ese desplazamiento. Por ejemplo, para las cadenas "acbaabch" y "aabghc" el

procedimiento devolvería un desplazamiento de 3 y un total de 4 caracteres que coinciden, ya que los posibles desplazamientos serían:

- Desplazando 0 caracteres, coinciden 2 caracteres (en negrita):

a	c	b	a	a	b	c	h
a	a	b	g	h	c		

- Desplazando 1 carácter, coinciden 1:

a	c	b	a	a	b	c	h
	a	a	b	g	h	c	

- Desplazando 2 caracteres, coinciden 1:

a	c	b	a	a	b	c	h
		a	a	b	g	h	c

- Desplazando 3 caracteres, coinciden 4 caracteres (mayor coincidencia posible):

a	c	b	a	a	b	c	h
			a	a	b	g	h
						c	

- Y así sucesivamente hasta desplazar 7 caracteres (la longitud de la primera cadena) que coinciden 0 caracteres:

a	c	b	a	a	b	c	h								
								a	a	b	g	h	c		

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

(3.5 pts) 4.- Diseña un **algoritmo** que lea de teclado un texto, y muestre por pantalla un listado de todas las palabras del texto codificadas. La codificación de cada palabra consiste en introducir sílabas extra tras cada determinada letra de la palabra original. Concretamente se ha de insertar la sílaba PA, PE, PI, PO, PU según la letra anterior sea A, E, I, O, U respectivamente. **En la salida no habrá palabras repetidas.**

Ejemplo:

Entrada:

Introduzca el texto (FIN para terminar):

TU EXAMEN ESTA CHUPADO PERO MI EXAMEN NO FIN

Salida:

TUPU EPEXAPAMEPEN EPESTAPA CHUPUPAPADOPO PEPEROPO
MIPI NOPO

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.