

Normas importantes del examen:

- Los profesores no contestarán ninguna cuestión durante la realización del examen, exceptuando aquellas que estén relacionadas con algún posible error en el enunciado de alguna pregunta.
- Está prohibido llevar encima cualquier dispositivo con conexión a Internet. Los teléfonos móviles deben dejarse apagados en las mochilas, y éstas en los pasillos.
- No se puede desgrapar el examen.
- Puedes utilizar bolígrafo o lápiz (que se vea bien).
- No está permitido: instrucciones break fuera de sentencias switch, el uso de variables globales ni más de una sentencia return en una misma función.
- La duración del examen es de 2h.

Ejercicio 1 (2 puntos)

Escribe la función recursiva `sonIgualesEnImpares(char cad1[], char cad2[], int i)`, que recibe dos cadenas y un índice `i`. La función debe devolver `true` si los caracteres en las posiciones impares de ambas cadenas son iguales, y `false` en caso contrario. La verificación debe comenzar desde el índice `i`.

Ejemplo:

```
char cad1[] = "a1c2eA";
```

```
char cad2[] = "b1d2fA";
```

La llamada `sonIgualesEnImpares(cad1, cad2, 0)` devolverá `true` porque 1, 2 y A son iguales en ambas cadenas.

Ejercicio 2 (2 puntos)

Escribe la función `unirCadenasDinamica` cuyo prototipo es:

```
char* unirCadenasDinamica(char* cad1, char* cad2);
```

La función debe devolver una nueva cadena resultante de unir `cad1` y `cad2`. La concatenación debe realizarse utilizando reserva dinámica de memoria, ajustando el tamaño de la memoria asignada a la memoria necesaria. La única función de la librería `<string.h>` que puedes utilizar es `strlen()`.

Ejemplo:

```
char* resultado = unirCadenasDinamica("Examen2 ", "de P1");  
// resultado contendrá "Examen2 de P1"
```

Ejercicio 3 (2,5 puntos)

Dadas las siguientes definiciones de la práctica 7:

```
typedef enum {NORTE, SUR, ESTE, OESTE} TDireccion;
```

<pre>typedef struct { int x; int y; } TPosicion; typedef struct { TPosicion posicion; TDireccion direccion; int distanciaMovimiento; char id; } TRobot;</pre>	<pre>typedef struct { TRobot* robots; int numRobots; } TGrupoRobots; typedef struct { char** matriz; int tamano; } TMapa;</pre>
--	--

- a) Implementa la función `trasladaRobot` que reciba como parámetros un robot, un incremento en X y un incremento en Y. La función debe modificar la posición del robot incrementándola con los valores pasados como parámetro, tal y como indica el ejemplo:

Datos entrada:

robot en posición (2,3)

incremento X: 10

incremento Y: 2

Después de llamar a la función `trasladaRobot`, la posición del robot será:
(12,5)

- b) Escribe ahora la función `trasladaRobots` que recorra todos los robots de un grupo de robots pasado como parámetro y, utilizando la función del apartado anterior, traslade con incremento en X: 10 y con incremento en Y: 15, aquellos robots que no se salgan de los límites del mapa (también pasado como parámetro). Si el traslado hace que un robot se salga de los límites del mapa, ese robot se quedará como estaba (sin trasladar su posición).

Ejercicio 4 (3,5 puntos)

- a) Diseña los tipos de datos necesarios para gestionar el inventario de productos de una tienda. De cada producto se necesita almacenar su código (entero), su nombre (máximo 30 caracteres), su precio, el nombre de los proveedores que lo sirven (máximo 30 caracteres cada nombre) y la cantidad de los mismos (un mismo producto puede servirlo más de un proveedor). Además, trabajaremos con una lista de productos, que contendrá un array dinámico de productos y la cantidad de productos.
- b) Implementa la función `leerProductos` que reciba como parámetros una lista de productos y un nombre de fichero. La función debe leer del fichero toda la información y almacenarla correctamente en la lista de productos. La memoria para el array de productos se reservará de forma dinámica y solo debe reservarse la cantidad que indica el primer dato del fichero (número de productos).

Cada línea del fichero de lectura, excepto la primera, tendrá el siguiente formato:

`CodigoProducto NombreProducto Precio NumProveedores Prov1 Prov2 ...`

Ejemplo del fichero de lectura:

número de productos

```
3
101 Camiseta 19.99 2 Proveedor1 Proveedor2
201 Pantalón 29.99 3 Proveedor3 Proveedor4 Proveedor5
301 Gorra 9.99 1 Proveedor1
```