



Tecnología de la Programación

Examen de prácticas 2023-2024 – Evaluación continua – Grupos 2, 3 y 4

Realizar un Proyecto de Programación en C mediante *Code::Blocks* (windows), llamado NombreApellidosDNI.cbp, el cual contenga un fichero de código fuente llamado NombreApellidosDNI.c que realice las siguientes tareas:

Cuestión 1. Arrays (3 puntos)

- Implementar la siguiente función:

```
// Crea y devuelve un array con las cantidades anuales de un
// depósito bancario que comienza con una cantidad de c euros y
// se incrementa anualmente con interés compuesto del r%, en un
// periodo de n años, es decir:
//  $v_0 = c$ 
//  $v_i = v_{i-1} + v_{i-1} \cdot r / 100, 0 < i \leq n$ 
double * deposito(int n, double c, double r)
```

- Construir un array invocando a la función anterior para un periodo de $n=5$ años, cantidad inicial de 1000 euros y tasa de interés del 3% ($c=1000, r=3$).
- Imprimir en pantalla los elementos del array. En el ejemplo se imprimiría lo siguiente:

```
1000.00 1030.00 1060.90 1092.73 1125.51 1159.27
```

- Liberar el array.

Cuestión 2. Estructuras de datos enlazadas (3 puntos)

- Declarar un tipo de datos *Estructura* para representar estructuras de datos enlazadas (lineales) de números enteros.
- Leer un número m desde teclado ($m \geq 0$) y construir una estructura de datos enlazada e , con celda de encabezamiento, de m números enteros aleatorios entre $-m$ y m .
- Imprimir en pantalla los elementos de la estructura de datos enlazada e .
- Implementar la siguiente función:

```
// Suprime los números negativos de la estructura de datos
// enlazada e.
void suprimir_negativos(Estructura e)
```

- Suprimir los elementos negativos de la estructura de datos enlazada e invocando la función *suprimir_negativos*.
- Imprimir en pantalla de nuevo los elementos de la estructura de datos enlazada e .
- Liberar la estructura de datos enlazada e .

Cuestión 3. Tipos de datos abstractos y programación modular (3 puntos)

- Dado el siguiente fichero de cabecera del TDA Línea (*Linea.h*):



```
#ifndef __LINEA_H
#define __LINEA_H

typedef struct LineaRep * Linea;

// Crea y devuelve una nueva línea desde el punto (x1,y1) hasta
// (x2,y2)
Linea LineaCrea(double x1, double y1, double x2, double y2);

// Libera la memoria asignada a la línea l
void LineaLibera(Linea l);

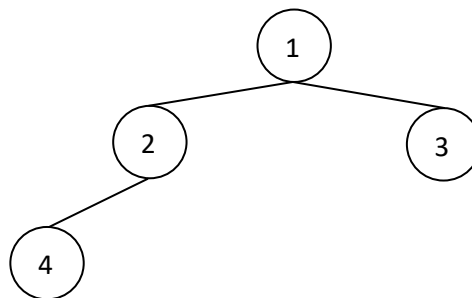
// Mueve la línea l con desplazamientos dx y dy en las
// coordenadas x e y respectivamente. Es decir, incrementa los
// valores x1 y x2 en una cantidad dx y los valores y1 e y2
// en una cantidad dy.
void LineaMueve(Linea l, double dx, double dy);

#endif // __LINEA_H
```

- Implementar el fichero `Linea.c` y compilarlo, generando el fichero `Linea.o`.

Cuestión 4. Árboles (3 puntos)

- Declarar una estructura de datos enlazada para representar árboles binarios de números enteros, de nombre `ArbolBinario`.
- Construir el siguiente árbol:



- Implementar la siguiente función:

```
// Devuelve el número de elementos mayores que b en el árbol
// binario a.
int mayores(ArbolBinario a, int b)
```

- Imprimir en pantalla el número de elementos mayores que un número `b` (leído desde teclado) del árbol `a`. En el ejemplo, `mayores(a,b)` con `b=2` devuelve 2.
- Liberar la memoria del árbol binario.

NOTA: No es necesario realizar gestión de errores de entrada de datos desde teclado.

ENTREGA: Se entregará un fichero `.zip` o `.rar` con el directorio del proyecto a través de la tarea correspondiente del Aula Virtual.