Departamento de Ingeniería de la Información y las Comunicaciones

Facultad de Informática
UNIVERSIDAD DE MURCIA

Examen de

Tecnología de la Programación

Evaluación Continua 2023

Grupo 3.2

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

BL0QUE I (4 puntos)

1. Construir un array de números reales de tamaño $n \geq 1$, con asignación dinámica de memoria. El tamaño n del array debe leerse desde teclado.
2. Asignar valores aleatorios entre `-100` y `100` a los elementos del array.
3. Imprimir en pantalla los elementos del array.
4. Implementar las siguientes funciones:

```
// requerimientos: n ≥ 1
// Devuelve la media aritmética de los elementos de v
double media(double * v, int n)

// requerimientos: n ≥ 1
// Devuelve la desviación típica de los elementos de v
double dt(double * v, int n)
```

5. Imprimir en pantalla, con 2 decimales, la media aritmética y la desviación típica de los elementos del array.
6. Liberar el array.

BLOQUE II (4 puntos)

7. Declarar una estructura de datos enlazada (lineal) de números enteros, de nombre `Estructura`.
8. Implementar la siguiente función:

```
// requerimientos: n ≥ 1
// Devuelve una estructura enlazada con la sucesión {a_k}_{k=1}^n,
// con a_1 = 1,  a_k = a_{k-1} · (k − 1),  k = 2, … , n
```

```
// (es decir, a_k = (k − 1)!, k = 1, … , n)
Estructura serie_factorial(int n)
```

9. Crear una estructura de tamaño `n` mediante la función `serie_factorial` (véase ejemplo de la Figura 1 para `n=5`).

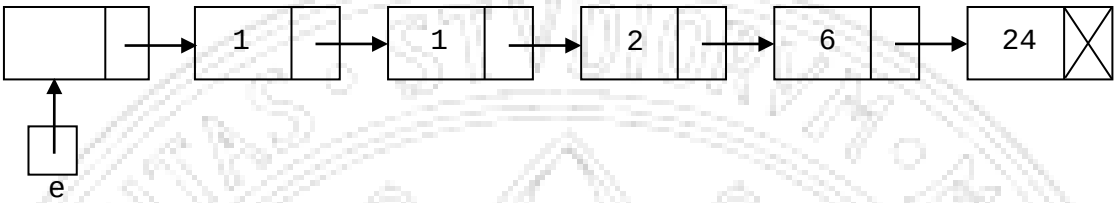

Fig. 1: Ejemplo de estructura enlazada obtenida con `e = serie_factorial(5)`.

10. Imprimir en pantalla los elementos de la estructura enlazada.
11. Liberar la estructura de datos enlazada.

BLOQUE III (4 puntos)

12. Declarar una estructura de datos enlazada para representar árboles generales de números enteros, de nombre `Arbol`.
13. Construir el árbol de la Figura 2.

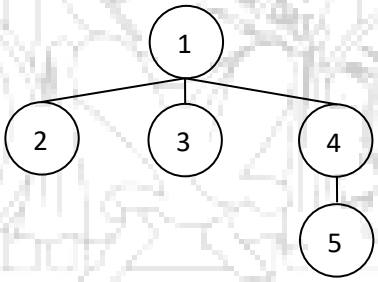

Fig. 2: Ejemplo de árbol general.

Usar la siguiente función para crear cada nodo del árbol:

```
// Devuelve un árbol de un sólo nodo raíz con valor elem e hijo
// izquierdo y hermano derecho a NULL
Arbol crea_arbol(int elem)
```

14. Implementar las siguientes funciones recursivas:

```
// Imprime en pantalla los elementos del árbol a en preorden
```

```
void preorden(Arbol a)

// Devuelve el producto de los elementos del árbol a, y 1 si el
// árbol a es vacío
int producto(Arbol a)

// Libera la memoria del árbol a
void liberar(Arbol a)
```

15. Realizar las siguientes tareas:
    15.1. Imprimir en pantalla en preorden los elementos del árbol con la función `preorden`.
    15.2. Imprimir en pantalla el producto de los elementos del árbol usando la función `producto`.
    15.3. Liberar el árbol `a` con la función `liberar`.

FÓRMULAS de la media y desviación típica de un vector $\boldsymbol{v} = (v_1, \dots, v_n)$:

Media:

$$\mu(\boldsymbol{v}) = \frac{1}{n}\sum_{i=1}^{n} v_i$$

Desviación típica:

$$\sigma(\boldsymbol{v}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (v_i - \mu(\boldsymbol{v}))^2}$$

NOTA: No se requiere implementar control de errores de entrada de datos.