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 3.3

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 dos arrays `v1` y `v2` de números enteros de tamaños $n \geq 1$ y $m \geq 1$ respectivamente, con asignación dinámica de memoria. Los tamaños n y m de los arrays deben leerse desde teclado.
2. Asignar valores leídos desde teclado a los elementos de los arrays.
3. Implementar las siguientes funciones:

```
//Imprime en pantalla los elementos del array v de tamaño n
void imprimir(int * v, int n)

// Devuelve un array con la concatenación de los arrays v1 y v2
// de tamaños n y m respectivamente
int * concatenar(int * v1, int * v2, int n, int m)
```

4. Crear un array `v3` como concatenación de `v1` y `v2` usando la función `concatenar`.
5. Imprimir `v1`, `v2` y `v3` usando la función `imprimir`.
6. Liberar todos los arrays.

BL0QUE 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 n elementos de la
// sucesión {a_k}_{k=1}^n, con a_1 = q, a_k = r · a_{k-1}, k = 2, ..., n
Estructura progresion_geometrica(int n, int q, int r)
```

9. Crear una estructura de tamaño `n` mediante la función `progresion_geometrica`, con `q=5` y `r=3` (véase ejemplo de la Figura 1 para `n=5`).

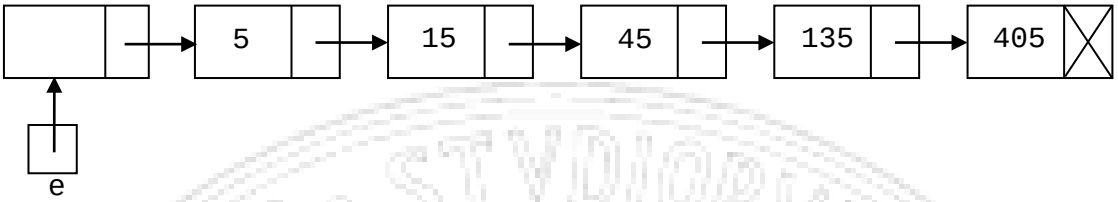

Fig. 1: Ejemplo de estructura enlazada obtenida con `e = progresion_geometrica(5,5,3)`.

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

BL0QUE 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.

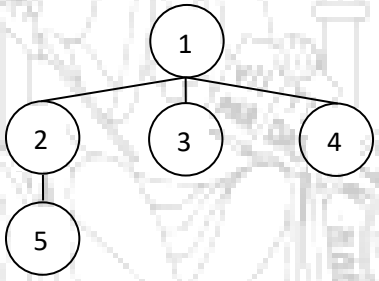

Fig. 2: Ejemplo de árbol general.

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 postorden
void postorden(Arbol a)

// Devuelve el número de nodos internos del árbol a
```

```
int nodos_internos(Arbol a)

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

15. Realizar las siguientes tareas:
    15.1. Imprimir en pantalla en postorden los elementos del árbol con la función `postorden`.
    15.2. Imprimir en pantalla el número de nodos internos del árbol usando la función `nodos_internos`.
    15.3. Liberar el árbol `a` con la función `liberar`.

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