

INSO - Introducción a la programación I – Curso 2024 - 2025

Ejercicio Evaluable 3

La práctica 3 estará formada por los siguientes 3 ejercicios:

- A. Lista de Nombres
- B. Colas
- C. Tic Tac Toe

A continuación, encontrarás los detalles de cada enunciado para escribir el código correspondiente. En todos ellos debes realizar el análisis del problema que se plantea, estudiando los datos de entrada que necesitas, plantea a continuación las sentencias de ejecución en base a las operaciones que debes realizar y finalmente muestra el resultado.

Se valorará si se ejecuta correctamente, así como la estructura general, la claridad y la legibilidad. Recuerda siempre utilizar identificadores que sean adecuados y el uso de comentarios.

Se valorará la simplicidad de las estructuras de control utilizada para implementar los algoritmos que resuelvan las necesidades de los programas. También debáis utilizar las funciones que se os sugiere en los enunciados

Ejercicio A. Lista de Nombres

Realiza un programa que almacene una lista de nombres en un array de cadenas (matriz de caracteres). El programa deberá comprobar que el nombre introducido no está repetido. Se podrán guardar un máximo de 15 nombres y cada nombre podrá tener como máximo 10 caracteres. El programa deberá presentar un menú con las siguientes opciones:

- 1- Introducir nombre
- 2- Mostrar nombres
- 3- Salir

A la hora de introducir nombre deberá comprobar si está repetido, en ese caso simplemente no lo guardará y mostrará un mensaje por pantalla. Si el nombre contiene más de 10 caracteres guardará los 10 primeros y desechará los demás.

El programa acaba cuando se pulsa salir o cuando se han introducido los 15 nombres, en cuyo caso antes de salir mostrará todos los nombres por pantalla.

Deberá utilizar las siguientes funciones:

```
//compara dos nombres y devuelve 1 si son diferentes y 0 si son
// iguales. Los dos prototipos son equivalentes
int sonDiferentes(char *nombre1, char * nombre2);
int sonDiferentes (char nombre1[], char nombre2[]);

//muestra todos los nombres guardados
void muestraNombres(char nombres[][11], int numeroNombres);

//devuelve 1 si encuentra el nombre en el array
//recibe como parametros la matriz, el número de nombres que contiene
//la matriz y la cadena a comprobar. Los dos prototipos son
equivalentes
int estaRepetido(char nombres[][11], int numeroNombres, char*
nuevoNombre );
int  estaRepetido(char  nombres[][11],  int  numeroNombres,  char
nuevoNombre []);

//presenta el menu y devuelve una opción válida
int menu()
```

Un ejemplo de ejecución puede ser:

```
Introduce una opcion
1 - Introducir jugador
2 - Mostrar jugadores
3 - Salir
    1
Introduce un nombre
    Carmen

Introduce una opcion
1 - Introducir jugador
2 - Mostrar jugadores
3 - Salir
    1
Introduce un nombre
    Juan Antonio

Introduce una opcion
1 - Introducir jugador
2 - Mostrar jugadores
3 - Salir
    1
Introduce un nombre
    Carmen
El nombre estaba repetido

Introduce una opcion
1 - Introducir jugador
2 - Mostrar jugadores
3 - Salir
    2
Carmen
Juan Anton

Introduce una opcion
1 - Introducir jugador
2 - Mostrar jugadores
3 - Salir
    3
Saliendo del programa
```

Ejercicio B. Colas

El alumno ha de implementar un sistema de gestión de colas mediante un array. Una cola es una estructura de datos que almacena múltiples elementos y sigue el patrón de First Input First Output (FIFO) que significa que el primer elemento encolado es el primer elemento en salir. Implementaremos una cola de números enteros sobre un array finito (cuya longitud es de 10 elementos) y dos funciones, una para encolar y otra para desencolar. La función de encolar se encargará de añadir un elemento nuevo al final de la cola (al final de los valores del array). La función para desencolar se encargará de extraer el primer elemento de la cola (el primer valor del array) y mover todos los valores de la cola una posición hacia delante, finalmente la función devuelve el elemento extraído inicialmente de la cola. Este ejercicio reforzará el aprendizaje del alumno con el movimiento de elementos en un array.

Hay que implementar una interfaz de usuario básica para probar el correcto funcionamiento de la cola. La interfaz tiene la siguiente forma:

Prueba de funcionamiento de la cola.

1. Encolar
2. Desencolar
3. Salir

La primera opción pide un número entero al usuario para encolarlo, la segunda opción desencola el primer elemento de la cola y la tercera opción termina el programa.

Los prototipos de funciones a implementar:

```
// Esta función recibe como parámetro el array de la cola, la longitud de la cola como puntero que variará tras encolar, la longitud máxima del array y el valor a encolar.
```

```
void encolar(int *cola, int *longitud, int longitudArray, int valor);
```

```
//Esta función recibe como parámetro el array de la cola y el puntero a la longitud de la cola que variará tras desencolar.
```

```
int desencolar(int *cola, int *longitud);
```

```
// Esta función imprime el menú y devuelve la opción elegida como resultado.
```

```
int mostrarMenuConResultado();
```

Prototipos de funciones auxiliares a implementar (opcionales):

```
// Esta función es para inicializar el array con ceros, que ayuda a inspeccionar mejor el estado del array con la función de mostrarArray.
```

```
void inicialziarArray(int *array, int longitudArray);
```

//Esta función sirve para inspeccionar el estado del array, lo que ayuda a revisar el correcto funcionamiento de la cola.

void mostrarArray(int *array, int longitudArray);

Los prototipos de funciones a implementar:

Prueba de funcionamiento de la cola.

1. Encolar
2. Desencolar
3. Salir

Introducir opcion: 1

Introduzca un valor entero a encolar: 3

Encolado el valor 3

Longitud de la cola: 1

Prueba de funcionamiento de la cola.

1. Encolar
2. Desencolar
3. Salir

Introducir opcion: 2

Valor desencolado: 3

Longitud de la cola: 0

Prueba de funcionamiento de la cola.

1. Encolar
2. Desencolar
3. Salir

Introducir opcion: 3

Saliendo del programa.

Ejercicio C. Tic Tac Toe

Implementa un juego de Tic Tac Toe para jugar en la terminal. Para implementar el tablero de 3 por 3 celdas (9 en total) hay que emplear un array bidimensional. El juego debe permitir que dos jugadores se turnen para ingresar sus marcas ("X" o "O") y determinar el ganador o si hay un empate. Se considera empate cuando un jugador no puede introducir nuevas marcas y se considera victoria de un jugador cuando este cumple 3 marcas seguidas formando una línea vertical, horizontal o diagonal. Este ejercicio permite desarrollar la capacidad de resolver problemas con arrays con más de una dimensión.

Los prototipos de funciones a implementar:

```
void inicializarTablero(char tablero[3][3]);  
void imprimirTablero(char tablero [3][3]);  
int comprobarVictoria(char tablero [3][3]);  
void hacerMovimiento(char tablero [3][3], int jugador);
```

Ejemplo de ejecución:

```
  |  |  
---|---|---  
  |  |  
---|---|---  
  |  |  
Jugador1, introduce tu movimiento (fila y columna): 0 0  
  
X |  |  
---|---|---  
  |  |  
---|---|---  
  |  |  
Jugador2, introduce tu movimiento (fila y columna): 1 1  
  
X |  |  
---|---|---  
  | O |  
---|---|---  
  |  |  
Jugador1, introduce tu movimiento (fila y columna): 0 1  
  
X | X |  
---|---|---  
  | O |  
---|---|---  
  |  |
```

Jugador2, introduce tu movimiento (fila y columna): 2 2

```
X | X |  
---|---|---  
  | O |  
---|---|---  
  |   | O
```

Jugador1, introduce tu movimiento (fila y columna): 0 2

```
X | X | X  
---|---|---  
  | O |  
---|---|---  
  |   | O
```

Jugador1 gana!