

EJERCICIOS LISTAS

Ejercicio de listas 1.

Código

```
ListasEjer1.cpp > main()
1  #include <iostream>
2  #include <stdlib.h>
3  using namespace std;
4
5  struct Nodo {
6      int dato;
7      Nodo *siguiente;
8  };
9
10 void insertarLista(Nodo *&, int);
11 void mostrarLista(Nodo *);
12 void calcularMayorMenor(Nodo *);
13
14 int main() {
15     Nodo *lista = NULL;
16     int dato;
17     char opcion;
18
19     do {
20         cout << "Digite un numero para agregarlo a lista: ";
21         cin >> dato;
22         insertarLista(lista, dato);
23
24         cout << "\nDesea agregar un nuevo numero(s/n): ";
25         cin >> opcion;
26     } while (opcion == 's' || opcion == 'S');
27
28     cout << "\nElementos de la lista: \n";
29     mostrarLista(lista); // mostramos la lista
30
31     calcularMayorMenor(lista);
32
33     cout << "\nPresiona Enter para continuar...";
34     cin.ignore(); // Limpiar el buffer
35     cin.get(); // Esperar a que el usuario presione Enter
36
37     return 0;
38 }
```

```

ListasEjer1.cpp > main()
14  int main() {
15      //
16      //
17      //
18      //
19
20      // Insertar elemento en la lista
21      void insertarLista(Nodo *&lista, int n) {
22          Nodo *nuevo_nodo = new Nodo();
23          Nodo *aux;
24
25          nuevo_nodo->dato = n;
26          nuevo_nodo->siguiente = NULL;
27
28          if (lista == NULL) { // Si la lista está vacía
29              lista = nuevo_nodo; // agregamos el primer nodo
30          } else {
31              aux = lista;
32              while (aux->siguiente != NULL) { // recorremos la lista hasta llegar al final
33                  aux = aux->siguiente;
34              }
35              aux->siguiente = nuevo_nodo; // agregamos el nuevo nodo al final de la lista
36          }
37          cout << "\tElemento " << n << " agregado a lista correctamente\n";
38      }
39
40      // Mostrar todos los elementos de la lista
41      void mostrarLista(Nodo *lista) {
42          Nodo *actual = lista; // No es necesario crear un nuevo nodo aquí
43
44          while (actual != NULL) { // mientras no sea el final de la lista
45              cout << actual->dato << " -> "; // mostramos el dato
46              actual = actual->siguiente; // avanzamos a la siguiente posición de la lista
47          }
48          cout << "NULL" << endl; // Añadir "NULL" al final para indicar el final de la lista
49      }
50
51      // Calcular el menor y mayor elemento de la lista
52      void calcularMayorMenor(Nodo *lista) {
53          if (lista == NULL) { // Comprobamos si la lista está vacía
54              cout << "La lista está vacía." << endl;
55              return;

```

```

56          return;
57      }
58
59      int mayor = lista->dato;
60      int menor = lista->dato;
61
62      while (lista != NULL) { // mientras no sea el final de la lista
63          if (lista->dato > mayor) { // Comprobamos el número mayor
64              mayor = lista->dato;
65          }
66          if (lista->dato < menor) { // Comprobamos el número menor
67              menor = lista->dato;
68          }
69          lista = lista->siguiente; // Avanzamos en la lista
70      }
71
72      cout << "\n\nEl mayor numero es: " << mayor << endl;
73      cout << "El menor numero es: " << menor << endl;
74  }
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

```

Ejecución.

```
→ PracticasU3 ./ListasEjer1
Digite un numero para agregarlo a lista: 6
    Elemento 6 agregado a lista correctamente

Desea agregar un nuevo numero(s/n): S
Digite un numero para agregarlo a lista: 5
    Elemento 5 agregado a lista correctamente

Desea agregar un nuevo numero(s/n): s
Digite un numero para agregarlo a lista: 8
    Elemento 8 agregado a lista correctamente

Desea agregar un nuevo numero(s/n): s
Digite un numero para agregarlo a lista: 2
    Elemento 2 agregado a lista correctamente

Desea agregar un nuevo numero(s/n): n

Elementos de la lista:
6 -> 5 -> 8 -> 2 -> NULL

El mayor numero es: 8
El menor numero es: 2

Presiona Enter para continuar... 
```

Explicación.

El programa gestiona una lista enlazada simple, permitiendo al usuario agregar números, mostrar los elementos de la lista y calcular el mayor y menor número de la lista.

Ejercicio listas 2.

Código:

```
ListaEjer2.cpp > calcularMayorMenor(Nodo *)
1  #include <iostream>
2  #include <stdlib.h>
3  using namespace std;
4
5  struct Nodo {
6      int dato;
7      Nodo *siguiente;
8  };
9
10 void insertarLista(Nodo *&, int);
11 void mostrarLista(Nodo *);
12 void calcularMayorMenor(Nodo *);
13
14 int main() {
15     Nodo *lista = NULL;
16     int dato;
17     char opcion;
18
19     do {
20         cout << "Digite un numero para agregarlo a lista: ";
21         cin >> dato;
22         insertarLista(lista, dato);
23
24         cout << "\nDesea agregar un nuevo numero(s/n): ";
25         cin >> opcion;
26     } while (opcion == 's' || opcion == 'S');
27
28     cout << "\nElementos de la lista: \n";
29     mostrarLista(lista); // mostramos la lista
30
31     calcularMayorMenor(lista);
32
33     cout << "\nPresiona Enter para continuar...";
34     cin.ignore(); // Limpiar el buffer
35     cin.get(); // Esperar a que el usuario presione Enter
36
37     return 0;
38 }
```

```
39
40 // Insertar elemento en la lista
41 void insertarLista(Nodo *&lista, int n) {
42     Nodo *nuevo_nodo = new Nodo();
43     Nodo *aux;
44
45     nuevo_nodo->dato = n;
46     nuevo_nodo->siguiente = NULL;
47
48     if (lista == NULL) { // Si la lista está vacía
49         lista = nuevo_nodo; // agregamos el primer nodo
50     } else {
51         aux = lista;
52         while (aux->siguiente != NULL) { // recorremos la lista hasta llegar al final
53             aux = aux->siguiente;
54         }
55         aux->siguiente = nuevo_nodo; // agregamos el nuevo nodo al final de la lista
56     }
57     cout << "\tElemento " << n << " agregado a lista correctamente\n";
58 }
59
60 // Mostrar todos los elementos de la lista
61 void mostrarLista(Nodo *lista) {
62     Nodo *actual = lista; // No es necesario crear un nuevo nodo aquí
63
64     while (actual != NULL) { // mientras no sea el final de la lista
65         cout << actual->dato << " -> "; // mostramos el dato
66         actual = actual->siguiente; // avanzamos a la siguiente posición de la lista
67     }
68     cout << "NULL" << endl; // Añadir "NULL" al final para indicar el final de la lista
69 }
70
71 // Calcular el menor y mayor elemento de la lista
72 void calcularMayorMenor(Nodo *lista) {
73     if (lista == NULL) { // Comprobamos si la lista está vacía
74         cout << "La lista está vacía." << endl;
75         return;
76     }
77 }
```

```

74         cout << "La lista está vacía." << endl;
75         return;
76     }
77
78     int mayor = lista->dato;
79     int menor = lista->dato;
80
81     while (lista != NULL) { // mientras no sea el final de la lista
82         if (lista->dato > mayor) { // Comprobamos el número mayor
83             mayor = lista->dato;
84         }
85         if (lista->dato < menor) { // Comprobamos el número menor
86             menor = lista->dato;
87         }
88         lista = lista->siguiente; // Avanzamos en la lista
89     }
90
91     cout << "\n\nEl mayor numero es: " << mayor << endl;
92     cout << "El menor numero es: " << menor << endl;
93 }
94

```

Ejecución:

```

Digite un numero para agregarlo a lista: 3
    Elemento 3 agregado a lista correctamente

Desea agregar un nuevo numero(s/n): s
Digite un numero para agregarlo a lista: 5
    Elemento 5 agregado a lista correctamente

Desea agregar un nuevo numero(s/n): s
Digite un numero para agregarlo a lista: 1
    Elemento 1 agregado a lista correctamente

Desea agregar un nuevo numero(s/n): n

Elementos de la lista:
3 -> 5 -> 1 -> NULL

El mayor numero es: 5
El menor numero es: 1

Presiona Enter para continuar...

```

Explicación.

Implementa una lista enlazada simple que permite al usuario agregar números y calcular el valor máximo y mínimo de esos números

Ejercicio 3.

Código.

```
2
3  #include <iostream>
4
5  using namespace std;
6
7  int main()
8  {
9      int calificacion = 0;
10     int contador = 0;
11     int suma = 0;
12
13     do {
14         cout << "\nIntroduzca calificación: ";
15         cin >> calificacion;
16         cout << "\nPresiona 0 para finalizar oh otra tecla para seguir\n";
17
18         if (calificacion != 0) {
19             if (calificacion >= 0 && calificacion <= 10) {
20                 contador++;
21                 suma = suma + calificacion;
22             }
23             else
24                 cout << "Calificación no es válida, solo valores entre 5 y 10\n";
25         }
26         else
27             cout << "\nFin de lectura de calificaciones\n";
28     }
29
30     while (calificacion != 0);
31
32     cout << "\nCalificaciones introducidas: " << contador;
33     cout << "\nEl promedio es: " << (suma/(float)contador);
34 }
```

Ejecución.

```
Introduzca calificación: 8
Presiona 0 para finalizar oh otra tecla para seguir
Introduzca calificación: 2
Presiona 0 para finalizar oh otra tecla para seguir
Introduzca calificación: 1
Presiona 0 para finalizar oh otra tecla para seguir
Introduzca calificación: 3
Presiona 0 para finalizar oh otra tecla para seguir
Introduzca calificación: 0
Presiona 0 para finalizar oh otra tecla para seguir
Fin de lectura de calificaciones

Calificaciones introducidas: 4
El promedio es: 3.5%
```

Explicación.

Permite al usuario ingresar calificaciones entre 0 y 10 hasta que se introduce 0 para finalizar. Calcula y muestra el número de calificaciones válidas ingresadas y su promedio. Si no se introdujeron calificaciones, informa que no hay datos válidos