

## Coding C++

RUN

MENU

Auto saved at 04:16:48

```
5
6 class Pila {
7 private:
8     class Nodo {
9     public:
10         char simbolo;
11         Nodo* sig;
12     };
13
14     Nodo* raiz;
15
16 public:
17     Pila() {
18         raiz = NULL;
19     }
20
21     ~Pila() {
22         while (!vacia()) {
23             extraer();
24         }
25     }
26
27     void insertar(char x) {
28         Nodo* nuevo = new Nodo();
29         nuevo->simbolo = x;
30         nuevo->sig = raiz;
31         raiz = nuevo;
32     }
33
34     char extraer() {
35         if (raiz == NULL) return '\0';
36         char informacion = raiz->simbolo;
37         Nodo* temp = raiz;
38         raiz = raiz->sig;
39         delete temp;
40         return informacion;
41     }
42
43     bool vacia() {
44         return raiz == NULL;
45     }
46 };
47
48 bool balanceada(char cadena[]) {
49     Pila* pila1 = new Pila();
50
51     for (int f = 0; f < strlen(cadena); f++) {
52         char c = cadena[f];
53         if (c == '(' || c == '[' || c == '{') {
```

Tab

{ }

“ ”

;

↶

↷

↺

=

\

&amp;

,

↶

↷

↺

## Coding C++

RUN

MENU

Auto saved at 04:16:48

```
44     return raiz == NULL;
45 }
46 };
47
48 bool balanceada(char cadena[]) {
49     Pila* pila1 = new Pila();
50
51     for (int f = 0; f < strlen(cadena); f++) {
52         char c = cadena[f];
53         if (c == '(' || c == '[' || c == '{') {
54             pila1->insertar(c);
55         } else if (c == ')' || c == ']' || c == '}') {
56             if (pila1->vacía()) {
57                 delete pila1;
58                 return false;
59             }
60             char abierto = pila1->extraer();
61             if ((c == ')' && abierto != '(') ||
62                 (c == ']' && abierto != '[') ||
63                 (c == '}' && abierto != '{')) {
64                 delete pila1;
65                 return false;
66             }
67         }
68     }
69
70     bool resultado = pila1->vacía();
71     delete pila1;
72     return resultado;
73 }
74
75 int main() {
76     char cadena[100];
77
78     cout << "Ingrese una formula con parentesis, corchetes y llaves";
79     cin.getline(cadena, 100);
80
81     if (balanceada(cadena)) {
82         cout << "La formula esta correctamente balanceada." << endl;
83     } else {
84         cout << "La formula no esta correctamente balanceada." << endl;
85     }
86
87     return 0;
88 }
```

Tab

{ }

“ ”

;

↶

↑

↷

=

\

&amp;

,

↵

↓

⇨

## Compile Result

Ingrese una formula con parentesis, corch  
etes y llaves:

## Coding C++

RUN

MENU

Auto saved at 04:19:53

```
1 #include <iostream>
2 using namespace std;
3
4 struct Nodo {
5     int info;
6     Nodo *sig;
7 };
8
9 class Cola {
10 private:
11     Nodo *raiz, *fondo;
12
13 public:
14     Cola() {
15         raiz = NULL;
16         fondo = NULL;
17     }
18
19     void insertar(int x) {
20         Nodo *nuevo = new Nodo();
21         nuevo->info = x;
22         nuevo->sig = NULL;
23         if (vacía()) {
24             raiz = nuevo;
25             fondo = nuevo;
26         } else {
27             fondo->sig = nuevo;
28             fondo = nuevo;
29         }
30     }
31
32     int extraer() {
33         if (!vacía()) {
34             int informacion = raiz->info;
35             Nodo *bor = raiz;
36             raiz = raiz->sig;
37             delete bor;
38             return informacion;
39         } else {
40             return -1;
41         }
42     }
43
44     void imprimir() {
45         Nodo *reco = raiz;
46         cout << "Listado de todos los elementos de la cola:\n";
47         while (reco != NULL) {
48             cout << reco->info << " - ";
49             reco = reco->sig;
50         }
51         cout << "\n";
52     }
```

Tab

{ }

“ ”

;

↶

↷

↺

=

\

&amp;

,

↵

↴

↲



# Coding C++

RUN

MENU

Auto saved at 04:19:35

```
56     }
57
58     ~Cola() {
59         while (!vacía()) {
60             extraer();
61         }
62     }
63 };
64
65 int main() {
66     Cola *cola1 = new Cola();
67
68     cola1->insertar(5);
69     cola1->insertar(10);
70     cola1->insertar(50);
71
72     cola1->imprimir();
73
74     cout << "Extraemos uno de la cola: " << cola1->extraer() << "
75
76     cola1->imprimir();
77
78     delete cola1;
79
80     return 0;
81 }
```

Tab

{ }

“ ”

;

↶

↷

↺

=

\

&amp;

,

↵

↴

↵

## Compile Result

```
Listado de todos los elementos de la cola
:
5 - 10 - 50 -
Extraemos uno de la cola: 5
Listado de todos los elementos de la cola
:
10 - 50 -

[Process completed - press Enter]
```