

1- Write a C++ program that creates a list and adds some elements to it using the push_back function. Then, iterate over the list and print each element.

اكتب برنامج ++C يقوم بإنشاء list وإضافة بعض العناصر إليها باستخدام وظيفة Push_back. ثم قم بالتكرار على القائمة وطباعة كل عنصر.

Output

```
5 10 15
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
using namespace std;

int main() {
    // Create a list
    list<int> myList;

    // Add elements to the list
    myList.push_back(5);
    myList.push_back(10);
    myList.push_back(15);

    // Iterate over the list and print each element
    for (int num : myList) {
        cout << num << " ";
    }

    return 0;
}
```

2- Write a C++ program that creates a list with elements 1, 2, 3, 4, 5, and then removes the last and first elements using the `pop_back` and `pop_front` functions, respectively. Finally, iterate over the modified list and print each element.

اكتب برنامج ++C يقوم بإنشاء list بالعناصر 1، 2، 3، 4، 5، ثم يقوم بإزالة العنصرين الأخير والأول باستخدام الدالتين `pop_back` و `pop_front`، على التوالي. وأخيرًا، قم بالتكرار على القائمة المعدلة وطباعة كل عنصر.

Output

```
2 3 4
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
using namespace std;

int main() {
    // Create a list with initial elements
    list<int> myList = {1, 2, 3, 4, 5};

    // Remove the last and first elements
    myList.pop_back();
    myList.pop_front();

    // Iterate over the modified list and print each element
    for (int num : myList) {
        cout << num << " ";
    }

    return 0;
}
```

3- Write a C++ program that creates a list with elements 1, 2, 3, 4, 5, and then prints the size of the list.

اكتب برنامج ++C يقوم بإنشاء list بالعناصر 1، 2، 3، 4، 5، ثم يطبع حجم القائمة.

Output

```
Size of the list: 5
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
using namespace std;

int main() {
    // Create a list with initial elements
    list<int> myList = {1, 2, 3, 4, 5};

    // Print the size of the list
    cout << "Size of the list: " << myList.size() << endl;

    return 0;
}
```

4- Write a C++ program that creates an empty list and then checks whether the list is empty or not. Print "Yes" if the list is empty, and "No" otherwise.

اكتب برنامج ++C يقوم بإنشاء list فارغة ثم يتحقق ما إذا كانت القائمة فارغة أم لا. اطبع "نعم" إذا كانت القائمة فارغة، و"لا" بخلاف ذلك.

Output

```
Is list empty? Yes
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
using namespace std;

int main() {
    // Create an empty list
    list<int> myList;

    // Check if the list is empty and print the result
    cout << "Is list empty? " << (myList.empty() ? "Yes" : "No") << endl;

    return 0;
}
```

5- Write a C++ program that creates a list with elements {1, 2, 3, 4, 5}, clears the list, and then prints the size of the list after clearing.

اكتب برنامج ++C يقوم بإنشاء list بالعناصر {1، 2، 3، 4، 5}، ويمسح list ثم يطبع حجم القائمة بعد المسح.

Output

```
Size of the list after clearing: 0
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
using namespace std;

int main() {
    // Create a list with elements {1, 2, 3, 4, 5}
    list<int> myList = {1, 2, 3, 4, 5};

    // Clear the list
    myList.clear();

    // Print the size of the list after clearing
    cout << "Size of the list after clearing: " << myList.size() << endl;

    return 0;
}
```

6- Write a C++ program that creates a list with elements {1, 2, 3, 1, 4, 2, 5}, sorts the list, removes consecutive duplicate elements, and then prints the unique elements of the list.

اكتب برنامج ++C يقوم بإنشاء list بالعناصر {1، 2، 3، 1، 4، 2، 5}، وفرز القائمة، وإزالة العناصر المكررة المتتالية، ثم طباعة العناصر الفريدة للقائمة.

Output

```
1 2 3 4 5
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
#include <algorithm>
using namespace std;

int main() {
    // Create a list with elements {1, 2, 3, 1, 4, 2, 5}
    list<int> myList = {1, 2, 3, 1, 4, 2, 5};

    // Sort the list
    myList.sort();

    // Remove consecutive duplicate elements
    myList.unique();

    // Print the unique elements of the list
    for (int num : myList) {
        cout << num << " ";
    }

    return 0;
}
```

7- Write a C++ program that creates a list with elements {1, 2, 3, 4, 5}, reverses the order of the elements, and then prints the reversed list.

اكتب برنامج ++C يقوم بإنشاء list بالعناصر {1، 2، 3، 4، 5}، ويعكس ترتيب العناصر، ثم يطبع القائمة المعكوسة.

Output

5 4 3 2 1

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
using namespace std;

int main() {
    // Create a list with elements {1, 2, 3, 4, 5}
    list<int> myList = {1, 2, 3, 4, 5};

    // Reverse the order of the elements
    myList.reverse();

    // Print the reversed list
    for (int num : myList) {
        cout << num << " ";
    }

    return 0;
}
```

8- Write a C++ program that creates a list with elements {1, 2, 3, 4, 5}, makes a copy of the list, and then prints the copied list.

اكتب برنامج ++C يقوم بإنشاء list بالعناصر {1، 2، 3، 4، 5}، وعمل نسخة من القائمة، ثم طباعة القائمة المنسوخة.

Output

```
Copied list: 1 2 3 4 5
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
using namespace std;

int main() {
    // Create a list with elements {1, 2, 3, 4, 5}
    list<int> originalList = {1, 2, 3, 4, 5};

    // Make a copy of the original list
    list<int> copiedList(originalList.begin(), originalList.end());

    // Print the copied list
    cout << "Copied list: ";
    for (int num : copiedList) {
        cout << num << " ";
    }

    return 0;
}
```


9- Write a C++ program that creates a list with elements {1, 2, 3, 4, 5} and then searches for a key element (in this case, 3) in the list. If the element is found, print its position; otherwise, print a message indicating that the element is not found.

اكتب برنامج ++C يقوم بإنشاء list بالعناصر {1، 2، 3، 4، 5} ثم يبحث عن عنصر رئيسي (في هذه الحالة، 3) في القائمة. إذا تم العثور على العنصر، قم بطباعة موضعه؛ وإلا، قم بطباعة رسالة تشير إلى عدم العثور على العنصر.

Output

```
Element 3 found at position: 2
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
#include <algorithm>
using namespace std;

int main() {
    // Create a list with elements {1, 2, 3, 4, 5}
    list<int> myList = {1, 2, 3, 4, 5};

    // Define the key to search
    int key = 3;

    // Find the key element in the list
    auto it = find(myList.begin(), myList.end(), key);

    // Check if the element is found
    if (it != myList.end()) {
        // Calculate the position of the found element
        int position = distance(myList.begin(), it);
        cout << "Element " << key << " found at position: " << position << endl;
    } else {
        cout << "Element not found in list." << endl;
    }

    return 0;
}
```

10- Write a C++ program that creates a list with elements {1, 2, 3, 4, 5} and removes a specified key element (in this case, 3) from the list. Print the list after removing the key element.

اكتب برنامج ++C يقوم بإنشاء list بالعناصر {1، 2، 3، 4، 5} ويزيل عنصرًا رئيسيًا محددًا (في هذه الحالة، 3) من القائمة. اطبع list بعد إزالة العنصر الأساسي.

Output

```
List after removing 3: 1 2 4 5
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <list>
using namespace std;

int main() {
    // Create a list with elements {1, 2, 3, 4, 5}
    list<int> myList = {1, 2, 3, 4, 5};

    // Define the key to remove
    int key = 3;

    // Remove the key element from the list
    myList.remove(key);

    // Print the list after removing the key element
    cout << "List after removing " << key << ": ";
    for (int num : myList) {
        cout << num << " ";
    }

    return 0;
}
```