1- Create two vectors and concatenate them into a third vector

x = {1, 2, 3}; y = {4, 5, 6};

أنشئ two vectors وقم بربطهما في vector ثالث

x = {1, 2, 3}

y= {4، 5، 6}

Output

```
1
2
3
4
5
6
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    vector<int> x = {1, 2, 3};
    vector<int> y = {4, 5, 6};

    vector<int> z;

    for (int i = 0; i < x.size(); i++) {
        z.push_back(x[i]);
    }

    for (int i = 0; i < y.size(); i++) {
        z.push_back(y[i]);
    }

    for (int i = 0; i < z.size(); i++) {
        cout << z[i] << endl;
    }

    return 0;
}
```

2- Create a vector of floating-point numbers and find the average numbers = {2.5, 3.0, 5.5, 1.8, 4.2};

أنشئ متجهًا لأرقام floating-point وأوجد المتوسط numbers= {2.5، 3.0، 5.5، 1.8، 4.2}؛

Output

```
Average: 3.4
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    vector<double> numbers = {2.5, 3.0, 5.5, 1.8, 4.2};

    double sum = 0;
    for (int i = 0; i < numbers.size(); i++) {
        sum += numbers[i];
    }

    double average = sum / numbers.size();

    cout << "Average: " << average << endl;
    return 0;
}
```

3- Create a vector and find the minimum element

x = {8, 3, 12, 5, 7}

قم بإنشاء vector وابحث عن الحد الأدنى للعنصر

x = {8, 3, 12, 5, 7}

Output

```
Minimum element: 3
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    vector<int> x = {8, 3, 12, 5, 7};

    int minElement = x[0];
    for (int i = 1; i < x.size(); i++) {
        if (x[i] < minElement) {
            minElement = x[i];
        }
    }

    cout << "Minimum element: " << minElement << endl;
    return 0;
}
```

4- Write a program to sort the first 4 elements of an array in descending order. The array x is initially {9, 8, 7, 6, 5}.

اكتب برنامجًا لفرز العناصر الأربعة الأولى في array ترتيبًا تنازليًا. array x هي في البداية {9، 8، 7، 6، 5}.

Output

```
Sorted Array in Descending Order:
9 8 7 6 5
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    int x[5] = {9, 8, 7, 6, 5};

    // Sorting the first 4 elements of the array in descending order
    sort(x, x + 4, greater<int>());

    // Printing the sorted array
    cout << "Sorted Array in Descending Order:\n";
    for(int i = 0; i < 5; i++)
        cout << x[i] << " ";

    return 0;
}
```

5- Create a program that sorts a vector of floating-point numbers in descending order. The vector floatVector is initially {3.14, 2.71, 1.618, 2.718}.

قم بإنشاء برنامج يقوم بفرز متجه أرقام floating-point بترتيب تنازلي.
vector floatVector هو في البداية {3.14، 2.71، 1.618، 2.718}.

Output

```
Sorted Vector of Floating-Point Numbers in Descending Order:
3.14 2.718 2.71 1.618
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    // Vector of floating-point numbers
    vector<float> floatVector = {3.14, 2.71, 1.618, 2.718};

    // Sorting the vector of floating-point numbers in descending order
    sort(floatVector.begin(), floatVector.end(), greater<float>());

    // Printing the sorted vector of floating-point numbers
    cout << "Sorted Vector of Floating-Point Numbers in Descending Order:\n";
    for(int i = 0; i < floatVector.size(); i++)
        cout << floatVector[i] << " ";

    return 0;
}
```

6- Develop a program that asks the user to input 7 integers, stores them in a vector and sorts the vector

تطوير برنامج يطلب من المستخدم إدخال 7 أعداد صحيحة، وتخزينها and sorts the vector

Input

```
Enter 7 integers:
5 6 3 1 5 8 9
```

Output

```
Sorted Vector:
1 3 5 5 6 8 9
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    vector<int> userVector;

    // Taking 7 integers from the user and storing in vector
    cout << "Enter 7 integers:\n";
    for(int i = 0; i < 7; i++) {
        int userInput;
        cin >> userInput;
        userVector.push_back(userInput);
    }

    // Sorting the vector in ascending order
    sort(userVector.begin(), userVector.end());

    // Printing the sorted vector
    cout << "Sorted Vector:\n";
    for(int i = 0; i < 7; i++)
        cout << userVector[i] << " ";


    return 0;
}
```

7- Write a program that uses a set to store and automatically sort a sequence of floating-point numbers. Insert the values 3.5, 1.2, 2.8, 4.0, and 1.2 into the set.

اكتب برنامجًا يستخدم set لتخزين سلسلة من أرقام floating-point وفرزها تلقائيًا. أدخل القيم 3.5 و1.2 و2.8 و4.0 و1.2 في set.

Output

```
Sorted Set Elements:
1.2 2.8 3.5 4
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    set<double> floatSet;

    floatSet.insert(3.5);
    floatSet.insert(1.2);
    floatSet.insert(2.8);
    floatSet.insert(4.0);
    floatSet.insert(1.2);

    // Printing the elements of the set
    set<double>::iterator it;
    cout << "Sorted Set Elements:\n";
    for (it = floatSet.begin(); it != floatSet.end(); it++)
        cout << (*it) << " ";

    return 0;
}
```

8- Create a program that employs a set to store and automatically sort a sequence of integers entered by the user. Prompt the user to input 6 integers and insert them into the set.

قم بإنشاء برنامج يستخدم set لتخزين وفرز سلسلة من الأعداد الصحيحة التي أدخلها المستخدم تلقائيًا. اطلب من المستخدم إدخال 6 أعداد صحيحة وإدراجها في set .

Input

```
Enter 6 integers:
1 5 6 2 8 7
```

Output

```
Sorted Set Elements:
1 2 5 6 7 8
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    set<int> userIntSet;

    // Taking 6 integers from the user and inserting into the set
    cout << "Enter 6 integers:\n";
    for (int i = 0; i < 6; i++) {
        int userInput;
        cin >> userInput;
        userIntSet.insert(userInput);
    }

    // Printing the elements of the set
    set<int>::iterator it;
    cout << "Sorted Set Elements:\n";
    for (it = userIntSet.begin(); it != userIntSet.end(); it++)
        cout << (*it) << " ";

    return 0;
}
```

9- Develop a program that uses a set to store and automatically sort a sequence of characters entered by the user. Prompt the user to input 8 characters and insert them into the set.

قم بتطوير برنامج يستخدم set لتخزين وفرز سلسلة من الأحرف التي أدخلها المستخدم تلقائيًا. اطلب من المستخدم إدخال 8 أحرف وإدراجها في المجموعة.

Input

```
Enter 8 characters:
n k p d H j p o
```

Output

```
Sorted Set Elements:
H d j k n o p
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    set<char> userCharSet;

    // Taking 8 characters from the user and inserting into the set
    cout << "Enter 8 characters:\n";
    for (int i = 0; i < 8; i++) {
        char userInput;
        cin >> userInput;
        userCharSet.insert(userInput);
    }

    // Printing the elements of the set
    set<char>::iterator it;
    cout << "Sorted Set Elements:\n";
    for (it = userCharSet.begin(); it != userCharSet.end(); it++)
        cout << (*it) << " ";

    return 0;
}
```

10- Create a program that utilizes a set to store and automatically sort a sequence of integers entered by the user. Prompt the user to input integers until they enter a negative number, and then insert them into the set.

قم بإنشاء برنامج يستخدم set لتخزين وفرز سلسلة من الأعداد الصحيحة التي أدخلها المستخدم تلقائيًا. اطلب من المستخدم إدخال أعداد صحيحة حتى يقوم بإدخال رقم سالب، ثم أدخلها في set.

Input

```
Enter integers (enter a negative number to stop):
5 6 7 9 -1
```

Output

```
Sorted Set Elements:
5 6 7 9
```

Solution

```cpp
// www.gammal.tech

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

int main() {
    set<int> userIntSet;

    // Taking integers from the user until a negative number is entered
    cout << "Enter integers (enter a negative number to stop):\n";
    int userInput;
    cin >> userInput;
    while (userInput >= 0) {
        userIntSet.insert(userInput);
        cin >> userInput;
    }

    // Printing the elements of the set
    set<int>::iterator it;
    cout << "Sorted Set Elements:\n";
    for (it = userIntSet.begin(); it != userIntSet.end(); it++)
        cout << (*it) << " ";

    return 0;
}
```
