

## lesson 6 CPP Sort

فى بعض الأحيان نحتاج إلى ترتيب عناصر **Array** أو **Vector**  
مثال :

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

int main() {
    int x[5] = { 9,8,7,6,5 };
    vector <int> y;
    y.push_back(9);
    y.push_back(8);
    y.push_back(7);
    y.push_back(6);
    y.push_back(5);

    cout << "Array: \n";
    for (int i = 0; i < 5; i++)
        cout << x[i] << " ";
    // هنا قمنا بطباعة array قبل الترتيب
    cout << "\nVector: \n";
    for (int i = 0; i < y.size(); i++)
        cout << y[i] << " ";
    // هنا قمنا بطباعة vector قبل الترتيب

}
```

**output:**

Array:

9 8 7 6 5

Vector:

9 8 7 6 5

هنا قمنا بطباعة عناصر **Array** و **Vector** بدون ترتيب  
لكن لترتيب عناصر **Array**، هناك مكتبة تسمى **algorithm** تحتوى على بعض  
الأوامر التى تساعدنا كثيراً فى لغة **C++** مثل أمر ترتيب العناصر و هو **sort** وطريقة  
استخدامه فى **array** كالتالى :

```
sort(x, x+n);
```

و **x+n** هو عدد الأرقام التى اريد ترتيبها

**مثال :**

إذا كان هناك **array** تحتوى على 10 أرقام ونريد ترتيب أول 5 عناصر فقط سيكون  
كالتالى :

```
sort(x, x+5);
```

هذا بالنسبة لترتيب **Array**

لكن بالنسبة لترتيب **vector** فنقوم باستخدام أمر **sort** لكن كالتالى :

```
sort(y.begin(), y.end());
```

كما يمكنك تحديد أيضاً من أين تبدأ

```
sort(y.begin()+n, y.end());
```

أو من أين ينتهى الترتيب

```
sort(y.begin(), y.end()-n);
```

و **n** هنا الرقم الذى تريد وضعه سواء أن متى تبدأ الترتيب أو تنتهى عنده

```

#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int main() {
    int x[5] = { 9,8,7,6,5 };
    vector <int> y;
    y.push_back(9);
    y.push_back(8);
    y.push_back(7);
    y.push_back(6);
    y.push_back(5);
    sort(x, x + 5);
    cout << "Array: \n";
    for (int i = 0; i < 5; i++)
        cout << x[i] << " ";
    // هنا قمنا بطباعة array بعد الترتيب
    sort(y.begin(), y.end());
    cout << "\nVector: \n";
    for (int i = 0; i < y.size(); i++)
        cout << y[i] << " ";
    // هنا قمنا بطباعة vector بعد الترتيب
}

```

output:

Array:

5 6 7 8 9

Vector:

5 6 7 8 9

كما يمكننى أيضاً عكس العناصر **array** باستخدام **reverse** لكن ينبغى أن نقوم بترتيبها أولاً إذا لم نقم بترتيبها سيكون أول عنصر آخر عنصر تم تبديلهم فقط

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

```
int main() {
    int x[5] = { 2,4,5,1,3 };

    reverse(x, x + 5);
    cout << "Array: \n";
    for (int i = 0; i < 5; i++)
        cout << x[i] << " ";
}
```

output:

Array:

3 1 5 4 2

لذلك نقوم بترتيبها أولاً :

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

int main() {
    int x[5] = { 2,4,5,1,3 };
    sort(x, x + 5);
    reverse(x, x + 5);
    cout << "Array: \n";
    for (int i = 0; i < 5; i++)
        cout << x[i] << " ";

}
```

**output:**

Array:

5 4 3 2 1

أما بالنسبة vector فيمكنك عكس عناصره، باستخدام reverse كالتالي :

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int main() {
    vector <int> y;
    y.push_back(2);
    y.push_back(4);
    y.push_back(5);
    y.push_back(1);
    y.push_back(3);
    sort(y.begin(), y.end());
    reverse(y.begin(), y.end());
    cout << "Vector: \n";;
    for (int i = 0; i < y.size(); i++)
        cout << y[i] << " ";
}
```

**output:**

Vector:

5 4 3 2 1