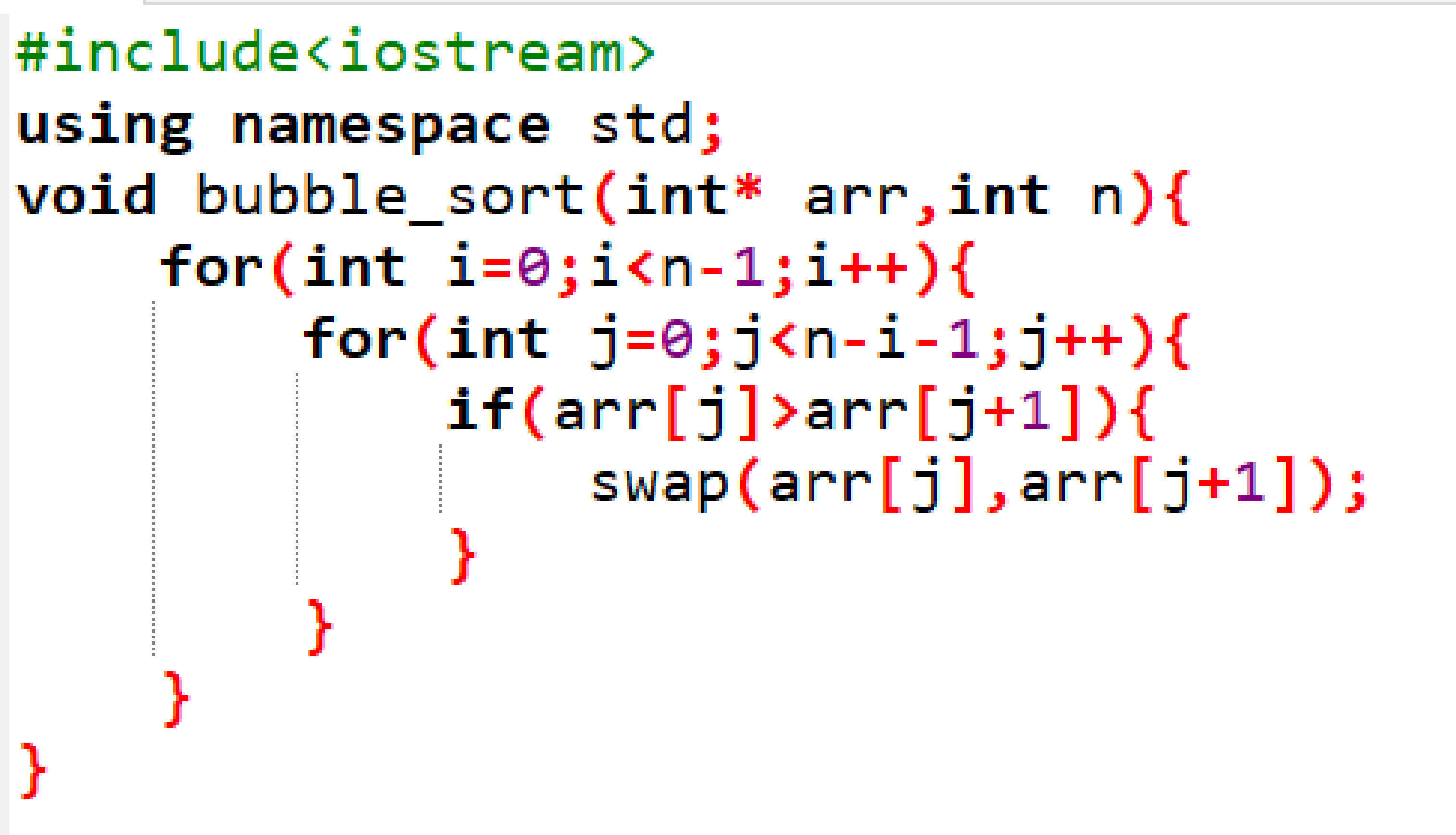
STL函數介紹

SPROUT 2019

bubble sort



```
#include<iostream>
using namespace std;
void bubble_sort(int* arr,int n){
    for(int i=0;i<n-1;i++){
        for(int j=0;j<n-i-1;j++){
            if(arr[j]>arr[j+1]){
                swap(arr[j],arr[j+1]);
            }
        }
    }
}
```

stl :: sort

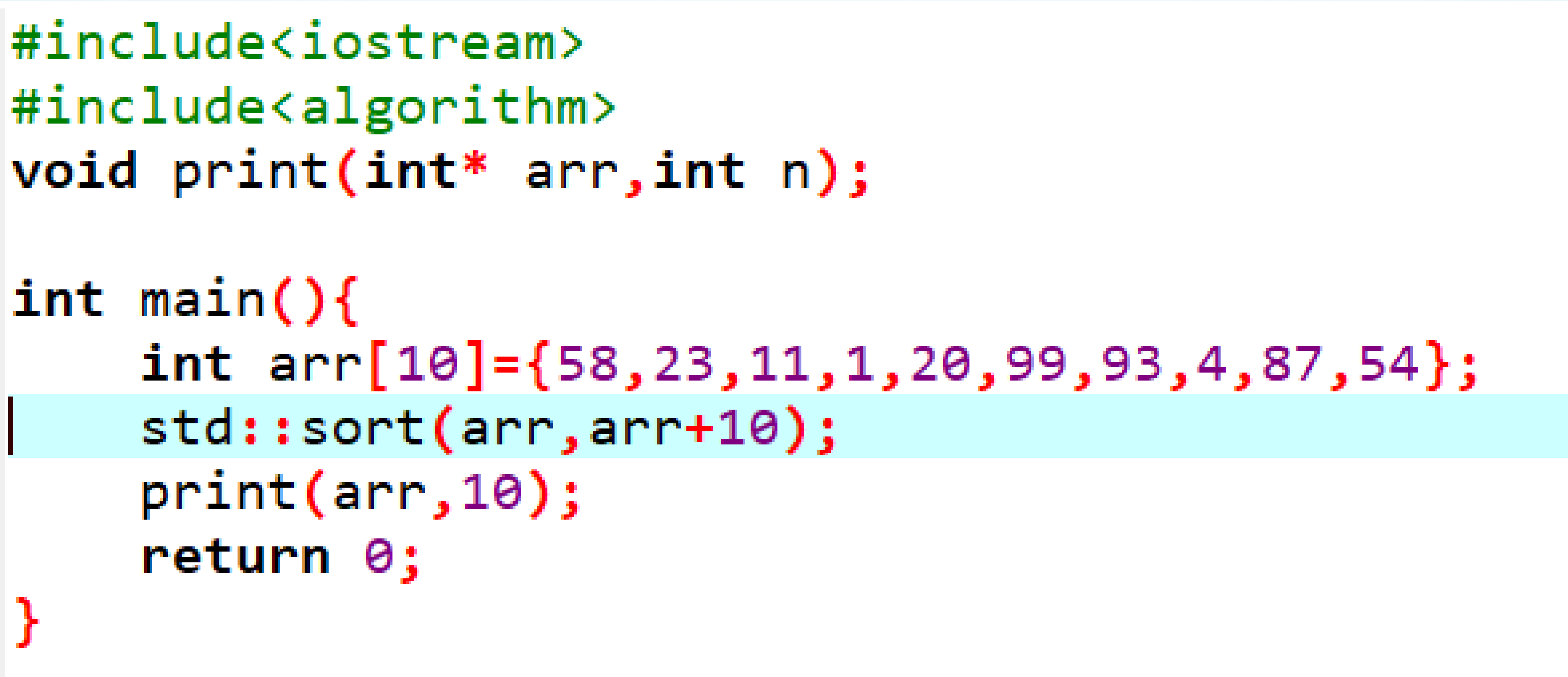
stl :: sort

- 定義於〈algorithm〉標頭檔中 → # include < algorithm >
- 格式：

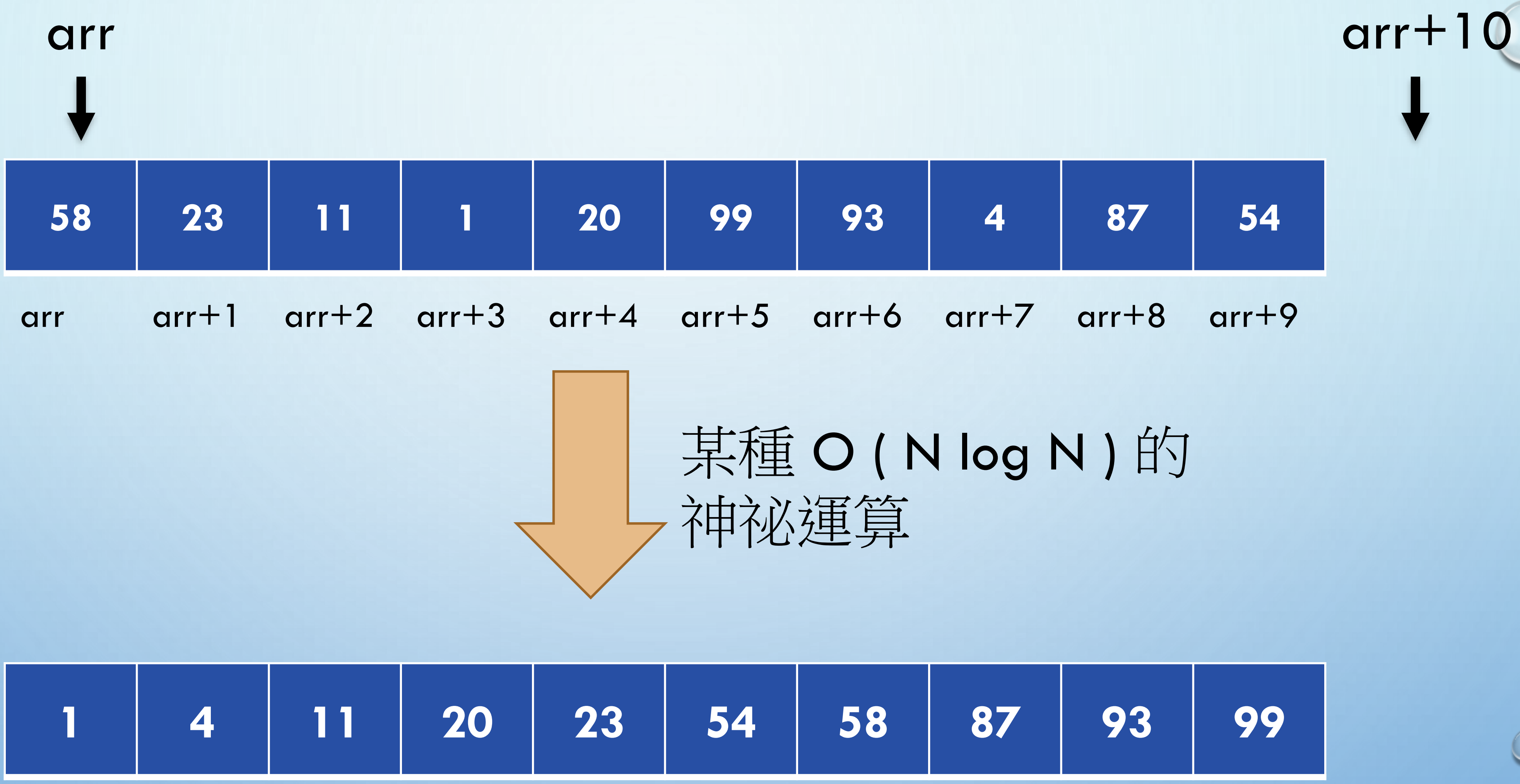
```
std :: sort ( begin , end ) ;
```

- 無回傳值
- 升冪排序

範例 (with array)



```cpp
#include<iostream>
#include<algorithm>
void print(int* arr,int n);

int main(){
    int arr[10]={58,23,11,1,20,99,93,4,87,54};
    std::sort(arr,arr+10);
    print(arr,10);
    return 0;
}
```



arr

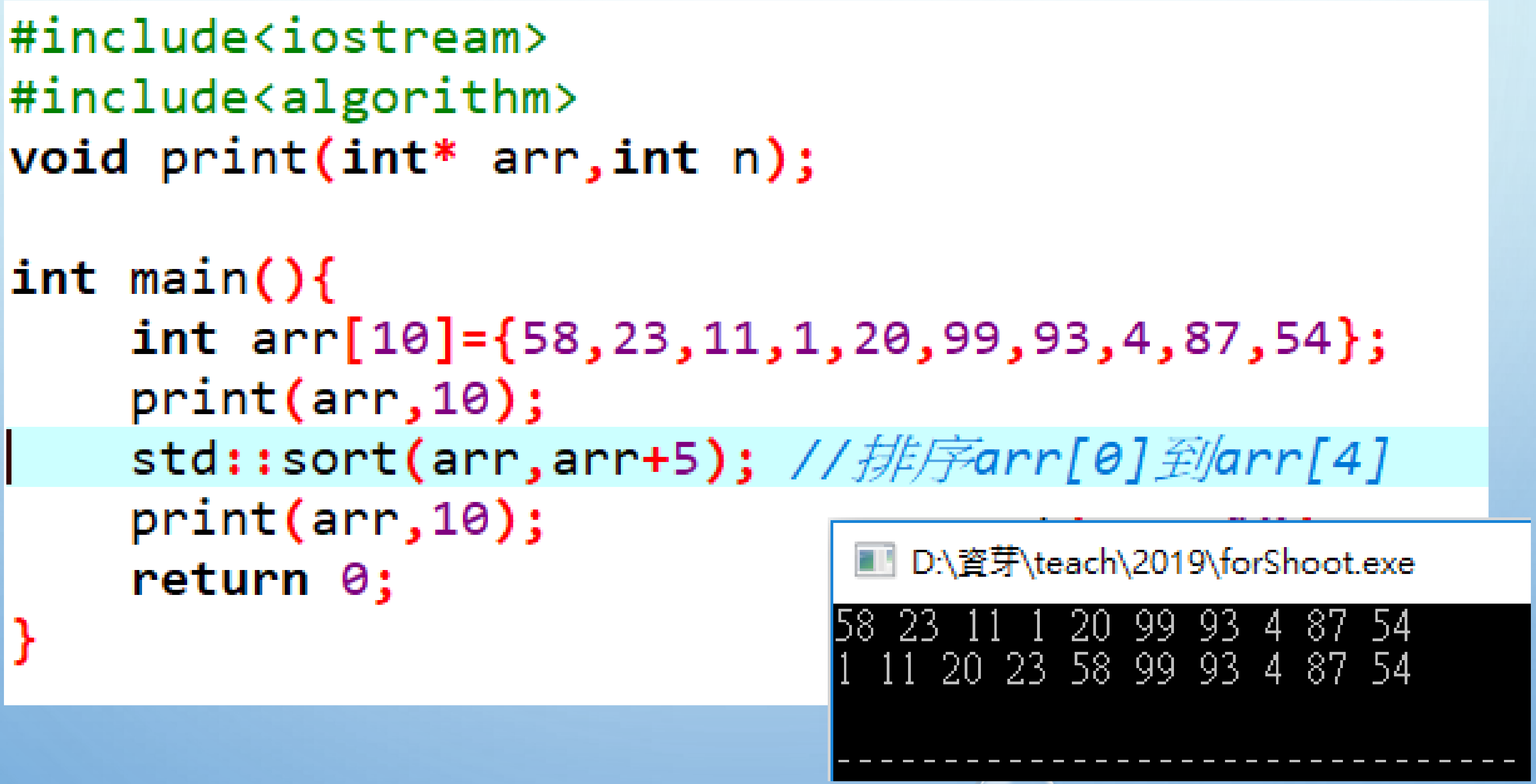
arr+10

58	23	11	1	20	99	93	4	87	54
arr	arr+1	arr+2	arr+3	arr+4	arr+5	arr+6	arr+7	arr+8	arr+9

某種 O (N log N) 的
神祕運算

1	4	11	20	23	54	58	87	93	99

如果只要排序一部分的話...

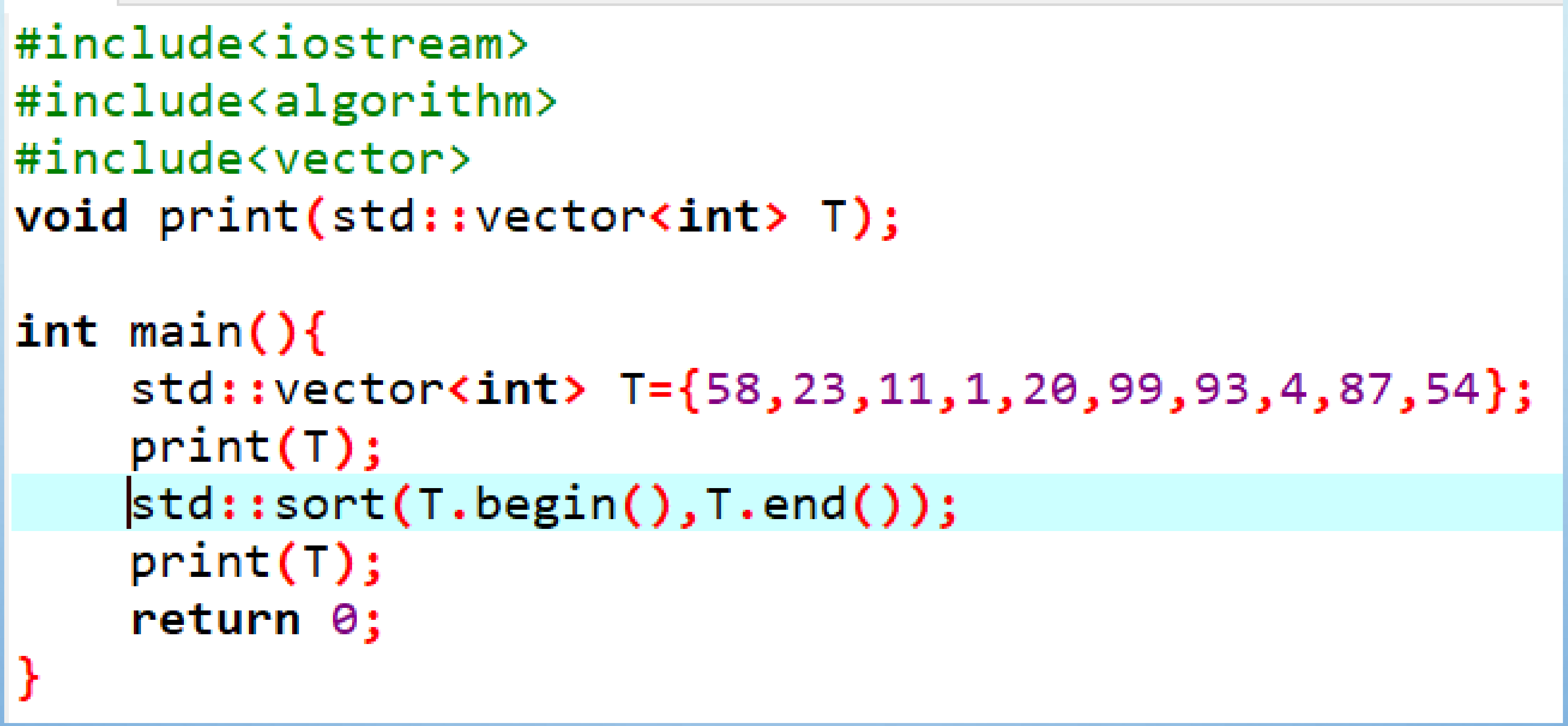


```cpp
#include<iostream>
#include<algorithm>
void print(int* arr,int n);

int main(){
    int arr[10]={58,23,11,1,20,99,93,4,87,54};
    print(arr,10);
    std::sort(arr,arr+5); //排序arr[0]到arr[4]
    print(arr,10);
    return 0;
}
```

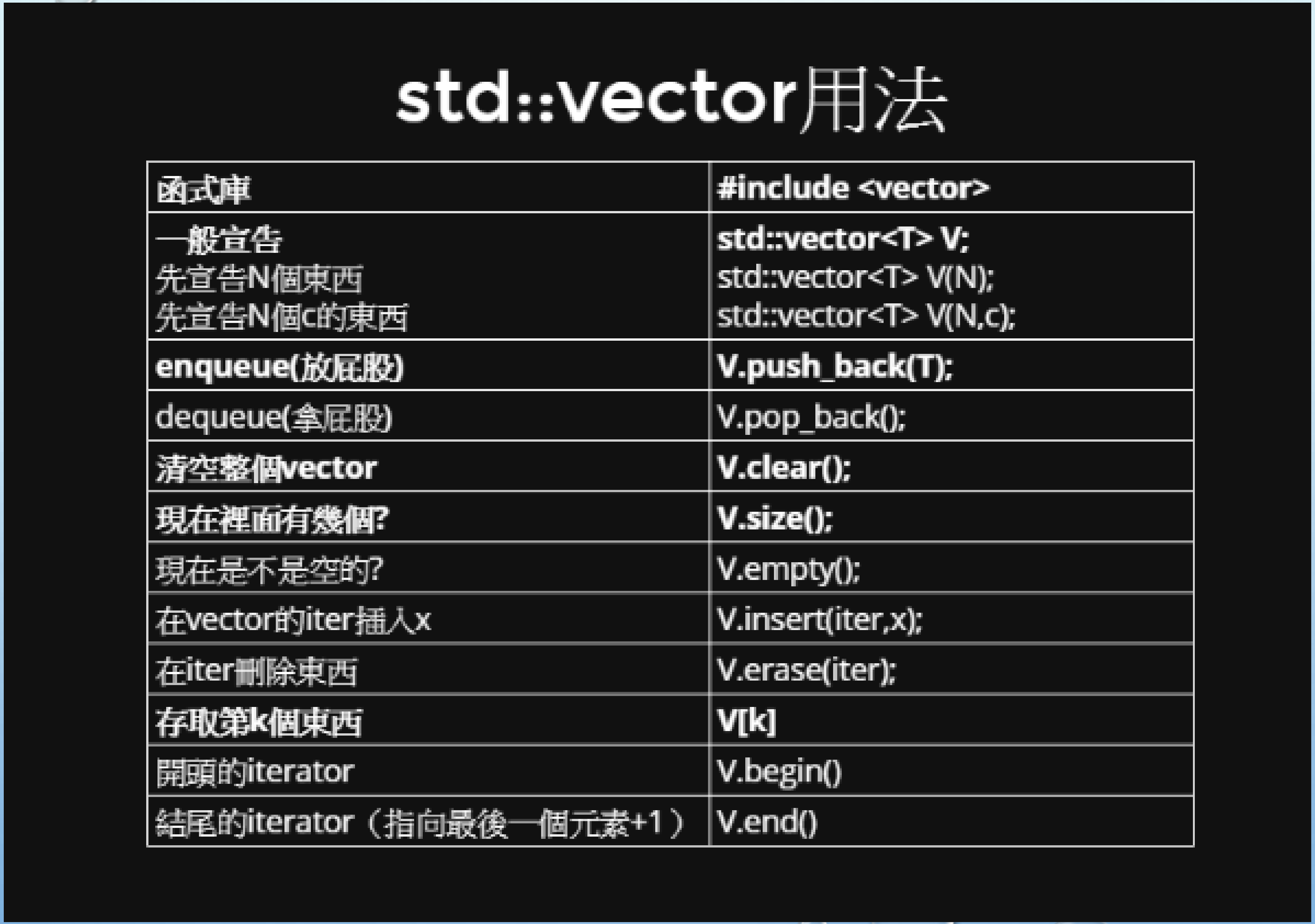
D:\資芽\teach\2019\forShoot.exe

```
58 23 11 1 20 99 93 4 87 54
1 11 20 23 58 99 93 4 87 54


--------------------------------
```

範例 (with vector)



```cpp
#include<iostream>
#include<algorithm>
#include<vector>
void print(std::vector<int> T);

int main(){
    std::vector<int> T={58,23,11,1,20,99,93,4,87,54};
    print(T);
    std::sort(T.begin(),T.end());
    print(T);
    return 0;
}
```



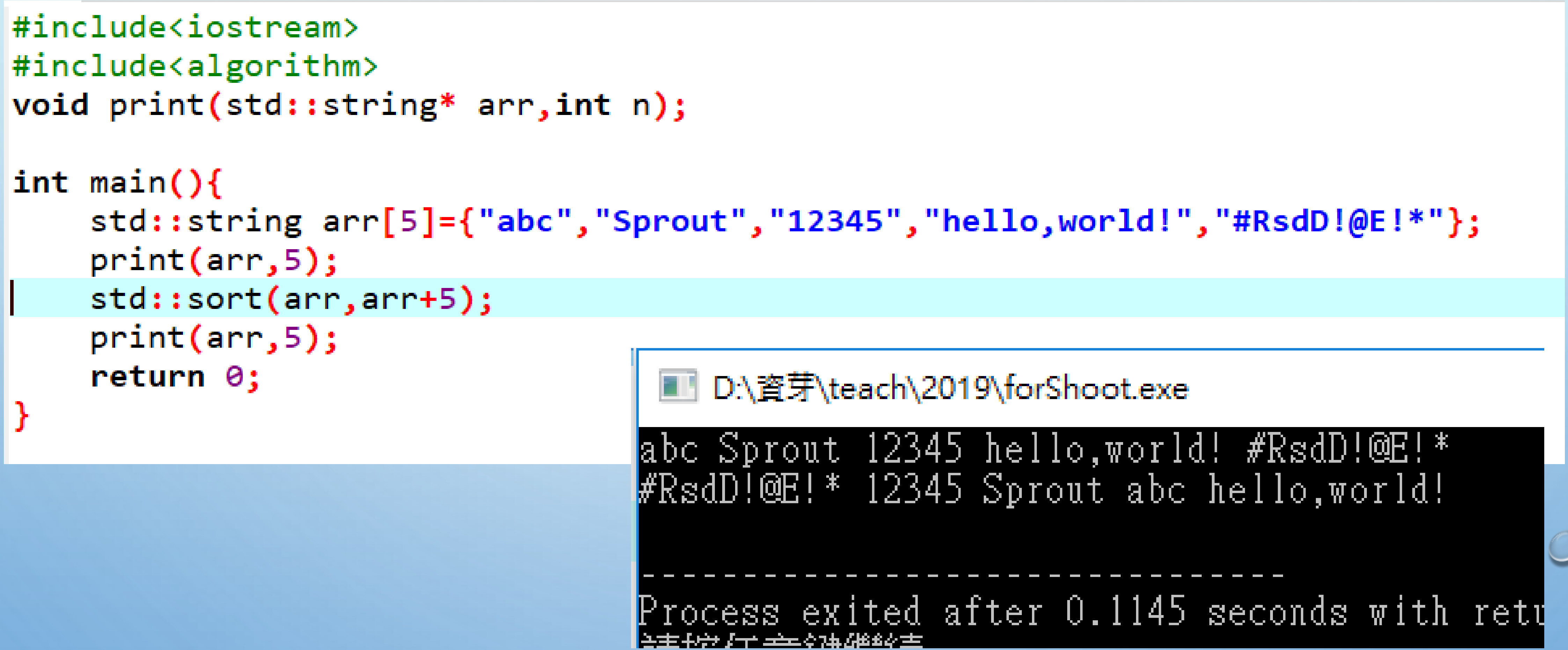
std::vector用法

函式庫	**#include <vector>**

一般宣告
先宣告N個東西
先宣告N個c的東西
	std::vector<T> V;
	std::vector<T> V(N);
	std::vector<T> V(N,c);

enqueue(放屁股)	**V.push_back(T);**
dequeue(拿屁股)	V.pop_back();
清空整個vector	**V.clear();**
現在裡面有幾個?	**V.size();**
現在是不是空的?	V.empty();
在vector的iter插入x	V.insert(iter,x);
在iter刪除東西	V.erase(iter);
存取第k個東西	**V[k]**
開頭的iterator	V.begin()
結尾的iterator（指向最後一個元素+1）	V.end()

範例 (with string)



```
#include<iostream>
#include<algorithm>
void print(std::string* arr,int n);

int main(){
    std::string arr[5]={"abc","Sprout","12345","hello,world!","#RsdD!@E!*"};
    print(arr,5);
    std::sort(arr,arr+5);
    print(arr,5);
    return 0;
}
```

D:\資芽\teach\2019\forShoot.exe

```
abc Sprout 12345 hello,world! #RsdD!@E!*
#RsdD!@E!* 12345 Sprout abc hello,world!


--------------------------------
Process exited after 0.1145 seconds with retu
請按任意鍵繼續
```

範例 (with string)

- ASCII

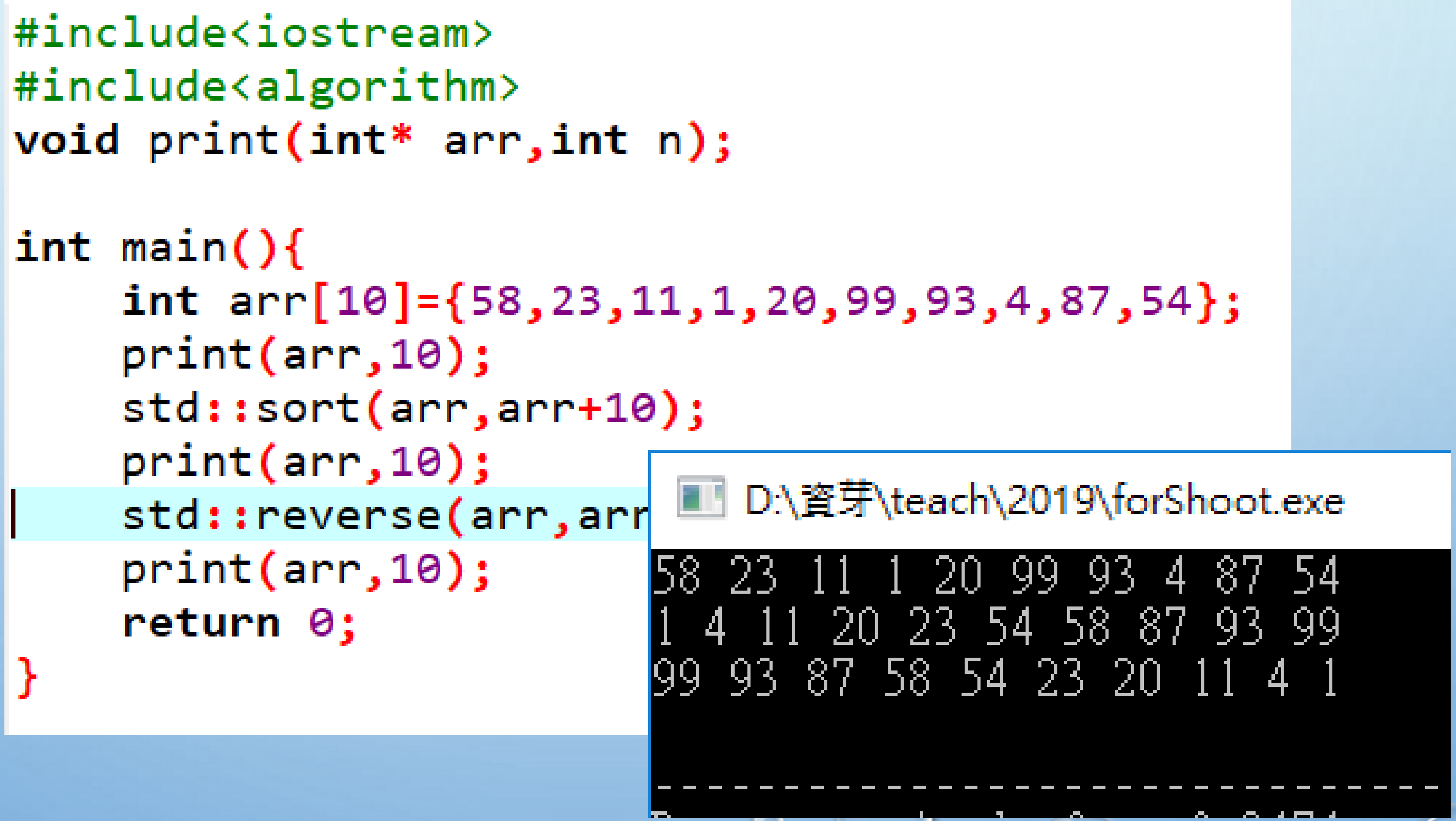
a	S	1	h	#
97	83	49	104	35

- “ #RsdD!@E! ” < “ 12345 ” < “ Sprout ” < “ abc ” < “ hello,world! ”
- 只要可以比大小就可以直接使用sort

Q：如果想要降冪排序怎麼辦?

reverse

- std :: reverse (begin , end) ; ← 跟 sort 類似
- 無回傳值

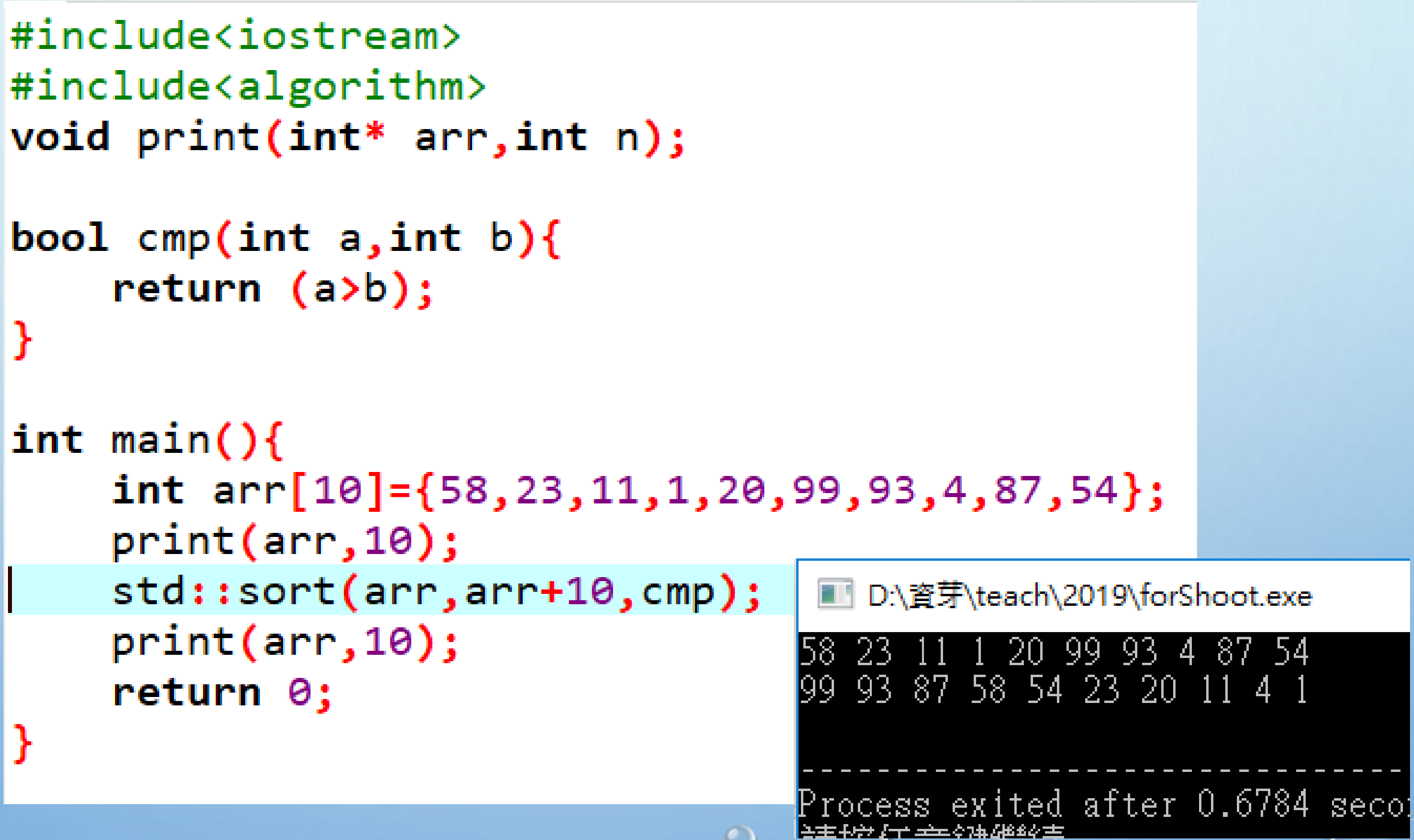


```
#include<iostream>
#include<algorithm>
void print(int* arr,int n);

int main(){
    int arr[10]={58,23,11,1,20,99,93,4,87,54};
    print(arr,10);
    std::sort(arr,arr+10);
    print(arr,10);
    std::reverse(arr,arr
    print(arr,10);
    return 0;
}
```

D:\資芽\teach\2019\forShoot.exe

```
58 23 11 1 20 99 93 4 87 54
1 4 11 20 23 54 58 87 93 99
99 93 87 58 54 23 20 11 4 1


--------------------------------
```

第三個參數-比較函數

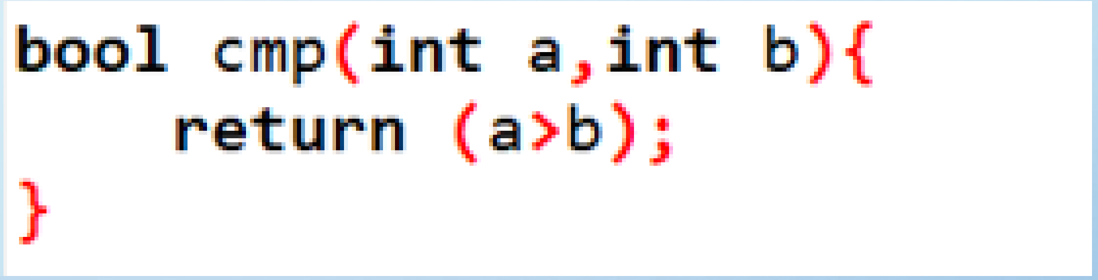


```
#include<iostream>
#include<algorithm>
void print(int* arr,int n);

bool cmp(int a,int b){
    return (a>b);
}

int main(){
    int arr[10]={58,23,11,1,20,99,93,4,87,54};
    print(arr,10);
    std::sort(arr,arr+10,cmp);
    print(arr,10);
    return 0;
}
```

```
58 23 11 1 20 99 93 4 87 54
99 93 87 58 54 23 20 11 4 1


--------------------------------
Process exited after 0.6784 seco
請按任意鍵繼續
```

第三個參數-比較函數

- sort 函式的 隱藏(?)參數，是一個特定型態的函式
- bool 型態的回傳值
- true：排序後a會在b前面
- false：排序後a會在b後面

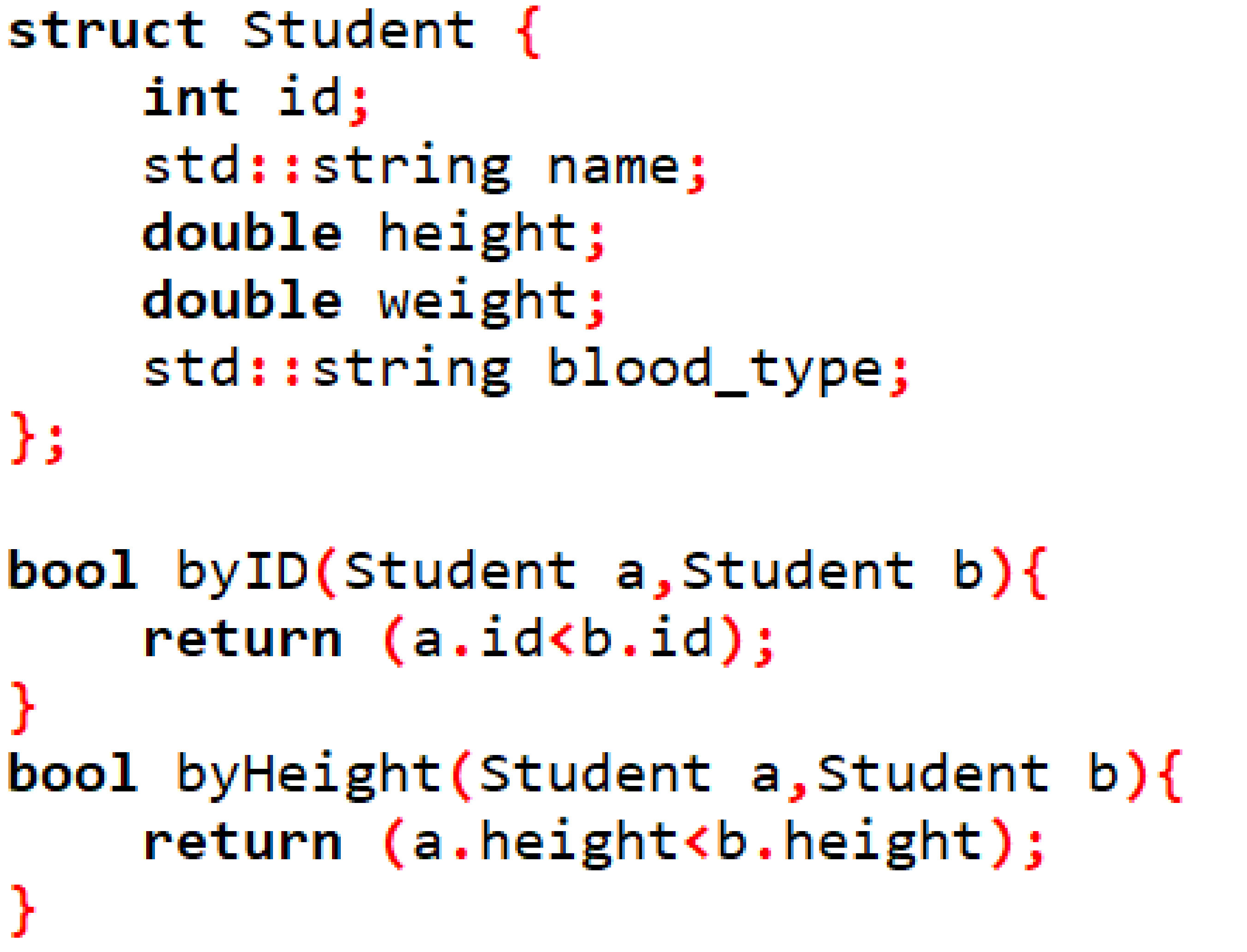


```
bool cmp(int a,int b){
    return (a>b);
}
```

Q：排序不能直接比大小的東西?

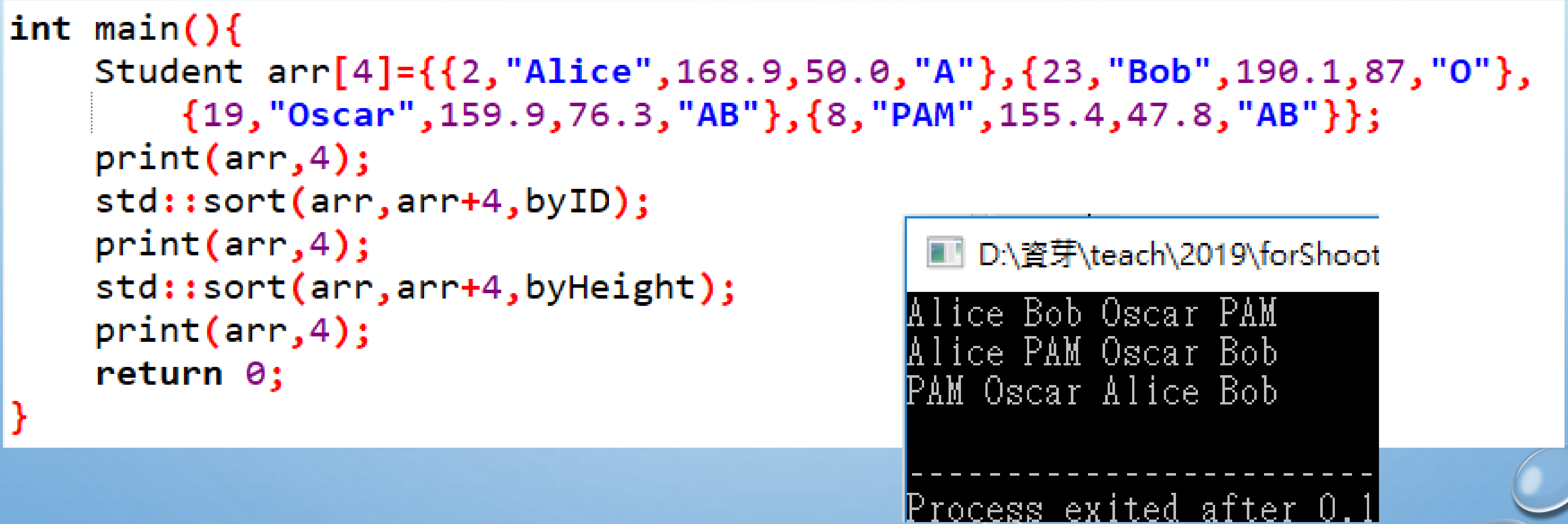
EX: STRUCT、CLASS

再次有請比較函數



```
struct Student {
    int id;
    std::string name;
    double height;
    double weight;
    std::string blood_type;
};

bool byID(Student a,Student b){
    return (a.id<b.id);
}
bool byHeight(Student a,Student b){
    return (a.height<b.height);
}
```

再次有請比較函數



```
int main(){
    Student arr[4]={{2,"Alice",168.9,50.0,"A"},{23,"Bob",190.1,87,"O"},
        {19,"Oscar",159.9,76.3,"AB"},{8,"PAM",155.4,47.8,"AB"}};
    print(arr,4);
    std::sort(arr,arr+4,byID);
    print(arr,4);
    std::sort(arr,arr+4,byHeight);
    print(arr,4);
    return 0;
}
```

D:\資芽\teach\2019\forShoot

```
Alice Bob Oscar PAM
Alice PAM Oscar Bob
PAM Oscar Alice Bob


--------------------------------
Process exited after 0.1
```

練習-147氣泡排序練習

stl :: next_permutation

好用的枚舉函數

排列

- 我們都知道 N 個東西可以產生 N ! 種排列
- 排列們又可依字典序排列好
- 比方說 { 1, 2, 3 } 有 3!=6 種排列，依字典序排列如下：
  - { 1, 2, 3 }
  - { 1, 3, 2 }
  - { 2, 1, 3 }
  - { 2, 3, 1 }
  - { 3, 1, 2 }
  - { 3, 2, 1 }

下一個排列

- 當排列的狀態是第P個的時候，第P+1個就是它的下一個排列，比如
- { 1, 2, 3 }
- { 1, 3, 2 } ← 現在的排列
- { 2, 1, 3 } ← 下一個排列
- { 2, 3, 1 }
- { 3, 1, 2 }
- { 3, 2, 1 }

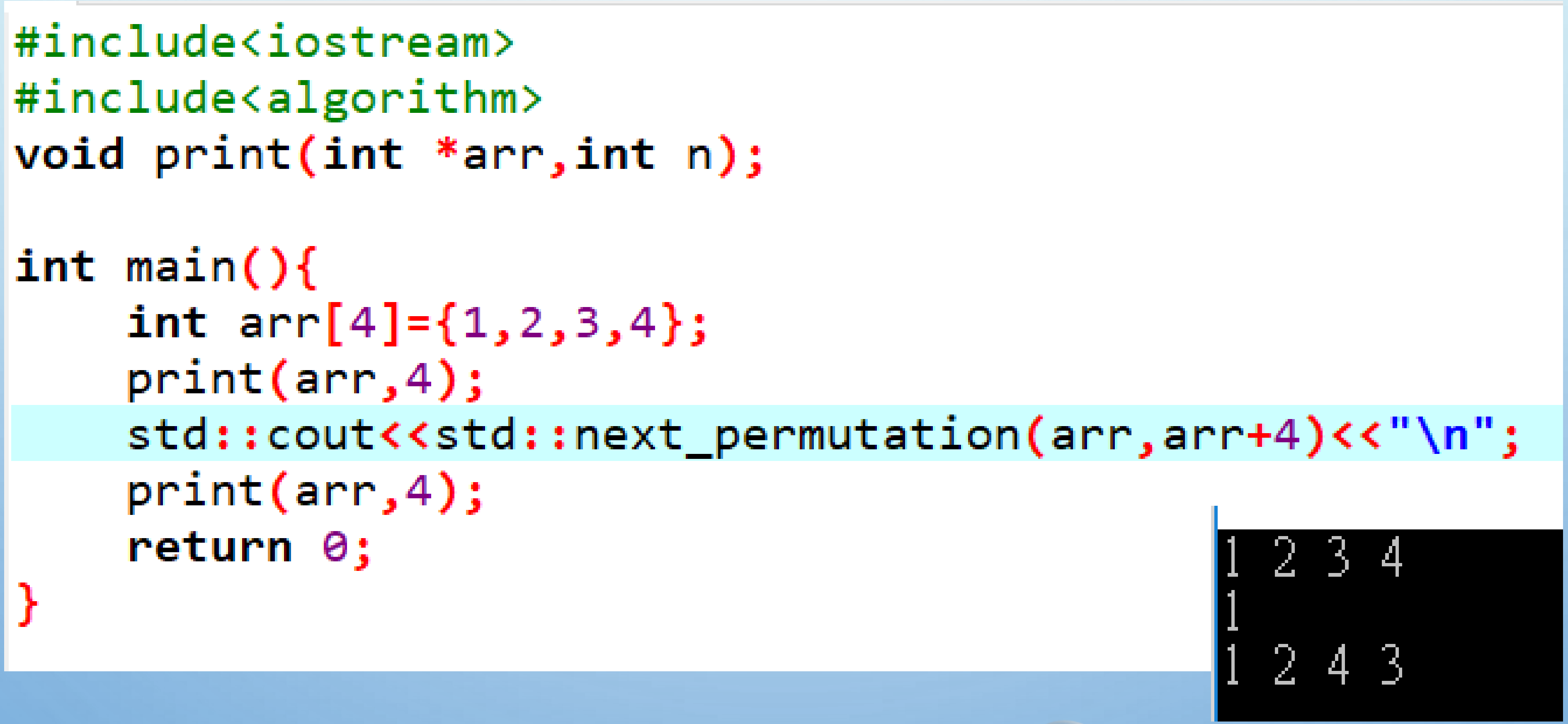
下一個排列

- 當排列的狀態是第P個的時候，第P+1個就是它的下一個排列，比如
- { 1, 2, 3 }
- { 1, 3, 2 }
- { 2, 1, 3 } ← 現在的排列
- { 2, 3, 1 } ← 下一個排列
- { 3, 1, 2 }
- { 3, 2, 1 }

stl :: next_permutation

- 定義於〈algorithm〉標頭檔中 → # include < algorithm >
- 格式：
- std :: next_permutation (begin , end)
- 排成下一個排列 (依字典序)
- bool型態的回傳值，代表還有沒有下一個排列

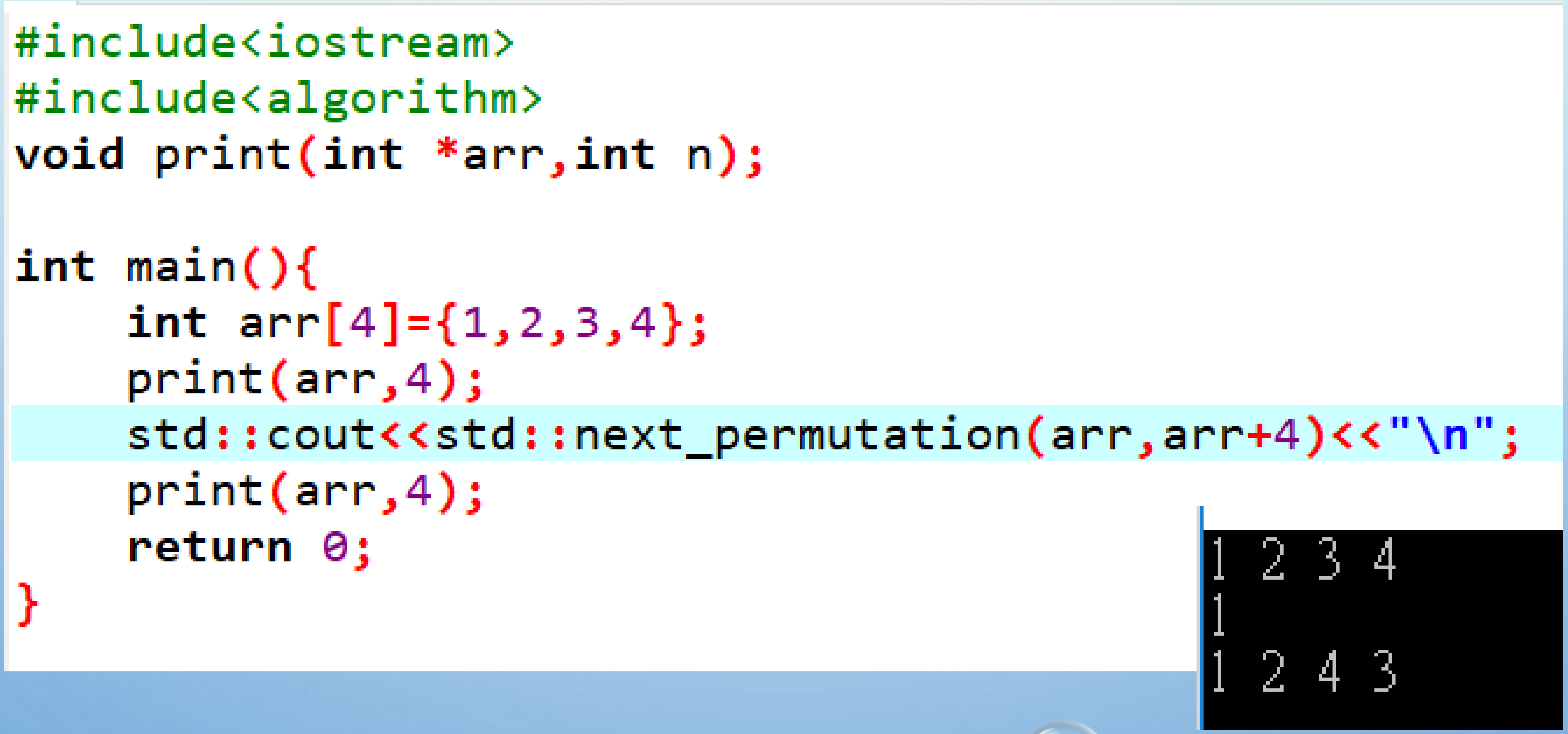
範例(with array)



```cpp
#include<iostream>
#include<algorithm>
void print(int *arr,int n);

int main(){
    int arr[4]={1,2,3,4};
    print(arr,4);
    std::cout<<std::next_permutation(arr,arr+4)<<"\n";
    print(arr,4);
    return 0;
}
```

```
1 2 3 4
1
1 2 4 3
```

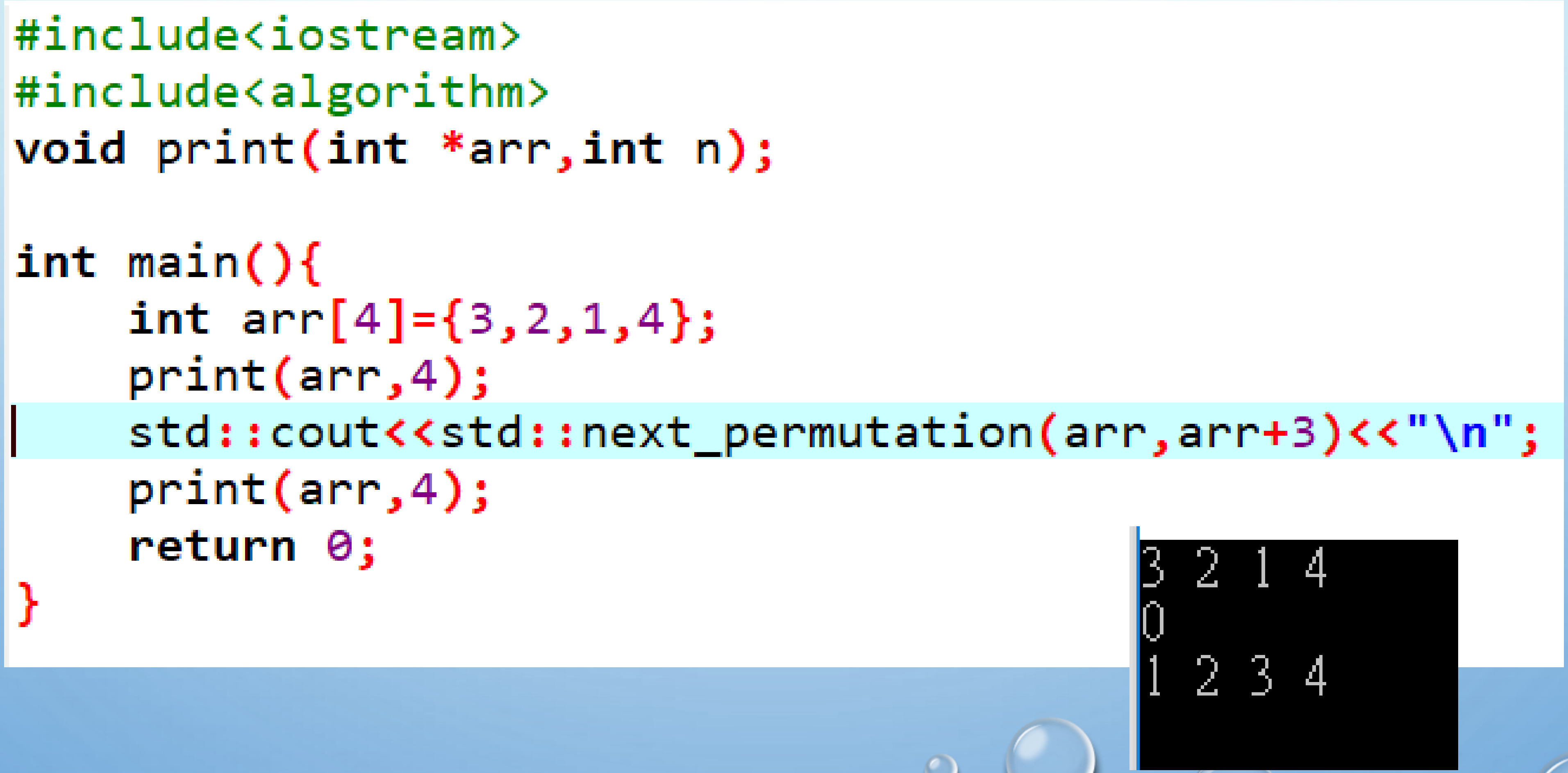
範例(with array)



```
#include<iostream>
#include<algorithm>
void print(int *arr,int n);

int main(){
    int arr[4]={1,2,3,4};
    print(arr,4);
    std::cout<<std::next_permutation(arr,arr+4)<<"\n";
    print(arr,4);
    return 0;
}
```

```
1 2 3 4
1
1 2 4 3
```

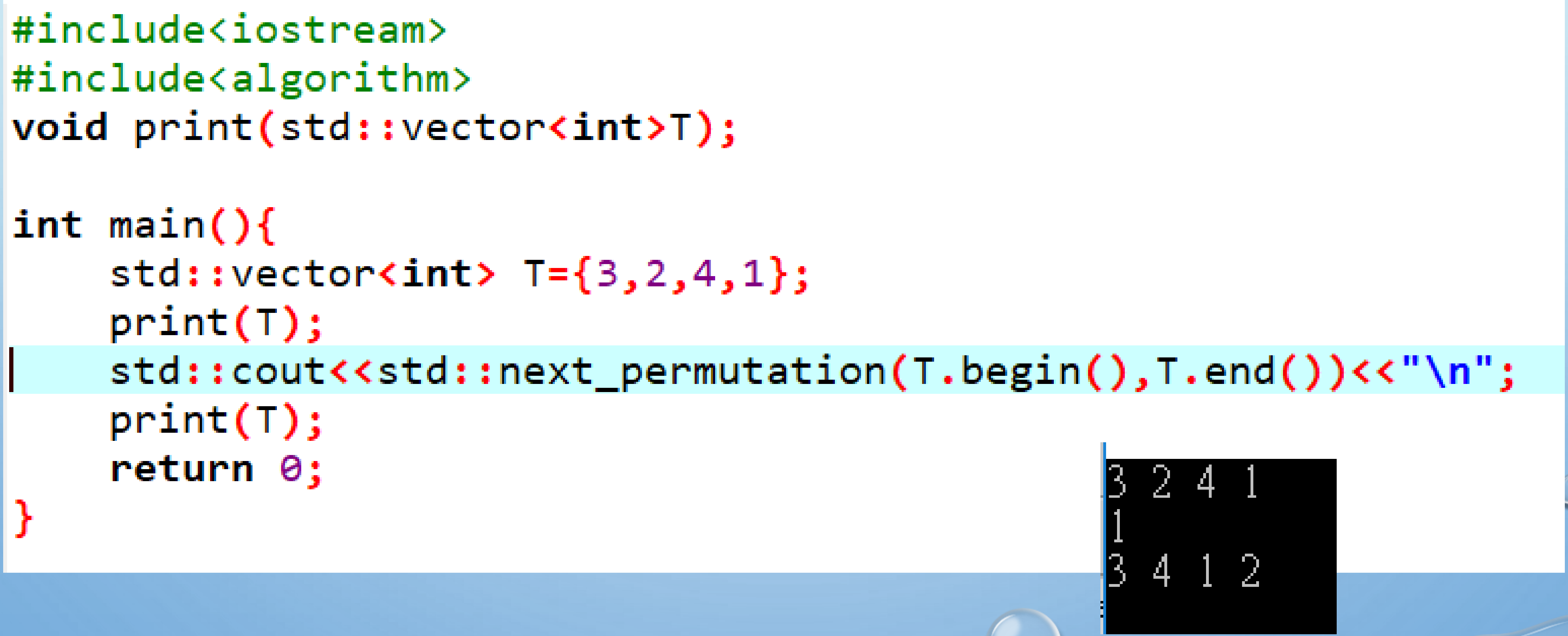
範例(with array)



```
#include<iostream>
#include<algorithm>
void print(int *arr,int n);

int main(){
    int arr[4]={3,2,1,4};
    print(arr,4);
    std::cout<<std::next_permutation(arr,arr+3)<<"\n";
    print(arr,4);
    return 0;
}
```

```
3 2 1 4
0
1 2 3 4
```

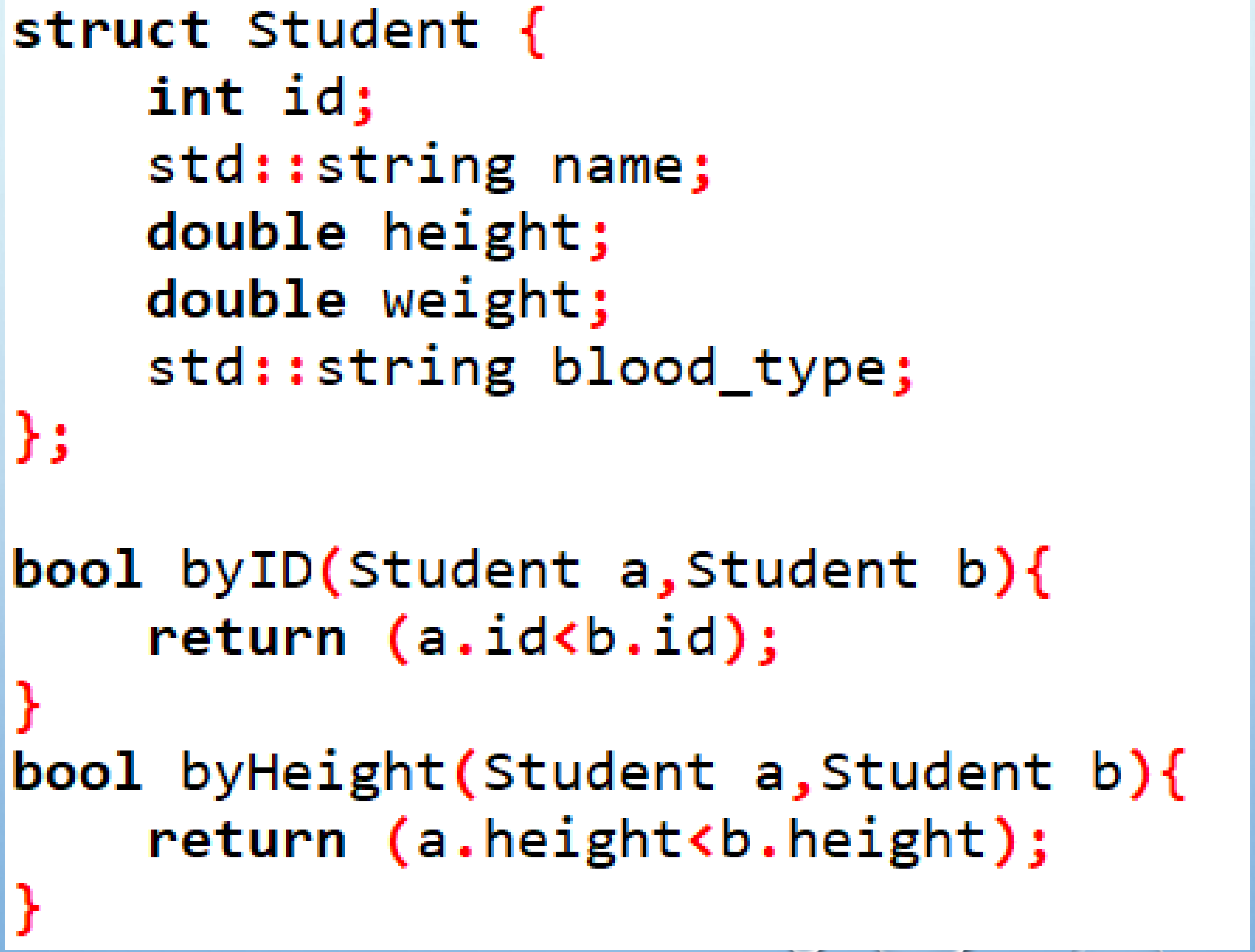
範例(with vector)



```
#include<iostream>
#include<algorithm>
void print(std::vector<int>T);

int main(){
	std::vector<int> T={3,2,4,1};
	print(T);
	std::cout<<std::next_permutation(T.begin(),T.end())<<"\n";
	print(T);
	return 0;
}
```

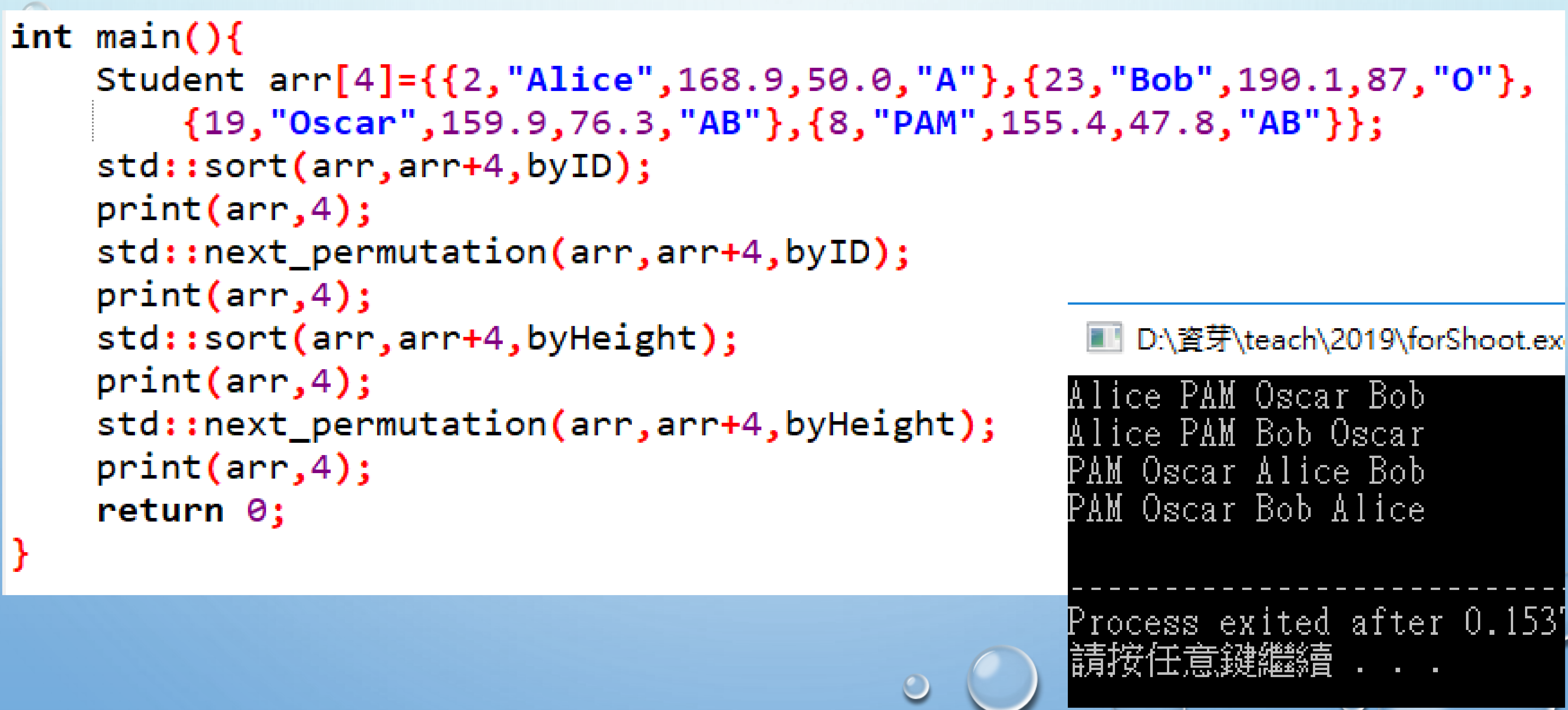
```
3 2 4 1
1
3 4 1 2
```

範例(with struct)



```
struct Student {
    int id;
    std::string name;
    double height;
    double weight;
    std::string blood_type;
};

bool byID(Student a,Student b){
    return (a.id<b.id);
}
bool byHeight(Student a,Student b){
    return (a.height<b.height);
}
```

範例(with struct)



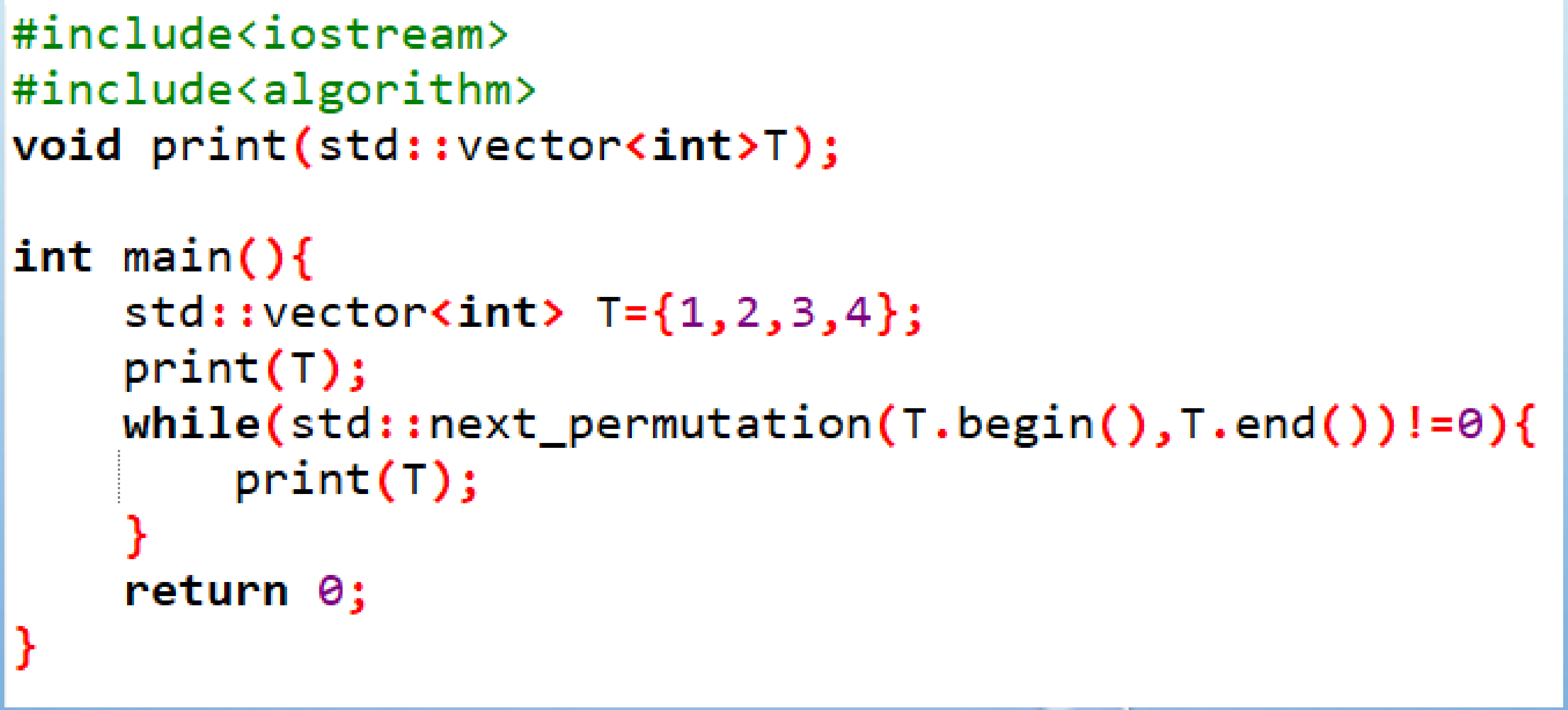
```
int main(){
    Student arr[4]={{2,"Alice",168.9,50.0,"A"},{23,"Bob",190.1,87,"O"},
        {19,"Oscar",159.9,76.3,"AB"},{8,"PAM",155.4,47.8,"AB"}};
    std::sort(arr,arr+4,byID);
    print(arr,4);
    std::next_permutation(arr,arr+4,byID);
    print(arr,4);
    std::sort(arr,arr+4,byHeight);
    print(arr,4);
    std::next_permutation(arr,arr+4,byHeight);
    print(arr,4);
    return 0;
}
```

D:\資芽\teach\2019\forShoot.ex

```
Alice PAM Oscar Bob
Alice PAM Bob Oscar
PAM Oscar Alice Bob
PAM Oscar Bob Alice


--------------------------------
Process exited after 0.153
請按任意鍵繼續 . . .
```

全排列

- 開個迴圈跑到回傳0為止



```cpp
#include<iostream>
#include<algorithm>
void print(std::vector<int>T);

int main(){
    std::vector<int> T={1,2,3,4};
    print(T);
    while(std::next_permutation(T.begin(),T.end())!=0){
        print(T);
    }
    return 0;
}
```

更多神奇好用的東西

HTTP://WWW.CPLUSPLUS.COM/REFERENCE/ALGORITHM/

練習題-153 文字轉轉轉