

什么是匿名函数？

C++中的匿名函数，类似于python的lambda函数，也就是在句中定义和声明的一个临时函数，仅在调用时才会创建函数对象，无需在头文件中声明。

```
1 [capture](parameters)->return-type{body}
2
3 [捕获列表](参数列表)->返回类型-{函数主体}
```

capture：捕获列表

- `[]`：捕获列表为空。在函数内无法使用外部变量。
- `[a]`：捕获列表为按值传递形式。在函数内仅能使用传递的变量值，无法改变变量。值在匿名函数生成时便已经确定，后续修改不会影响函数内的变量
- `[&a]`：按应用传递。可改变变量。
- `[=]` 用到的任何外部变量都隐式按值捕获
- `[&]` 用到的任何外部变量都隐式按引用捕获
- `return-type`：返回类型

```
1 #include <iostream>
2 #include <cstdio>
3 using namespace std;
4
5 int main() {
6
7     // auto f = [](int a, int b)->int{return a + b;};
8     // auto f = [](int a, int b){return a + b;};
9
10    int x = -1;
11
12    auto f = [&](int a, int b){
13        x = a + b;
14        // cout << x << endl;
15        return a + b;
16    };
17
18    cout << f(3, 4) << endl;
19    cout << x << endl;
20
21    return 0;
22 }
```

```
1 #include <iostream>
2 #include <cstdio>
3 #include <vector>
4 #include <algorithm>
5 using namespace std;
6
7 typedef pair<int, int> PII;
8
```

```

9  vector<PII> q;
10
11  int main() {
12
13      int n;
14      cin >> n;
15      for(int i = 1; i <= n; i++){
16          int x, y;
17          cin >> x >> y;
18          q.push_back({x, y});
19      }
20
21      sort(q.begin(), q.end(), [&](const PII& a, const PII& b){
22          if(a.first == b.first) return a.second < b.second;
23          return a.first > b.first;
24      });
25
26      for(auto x : q)
27          cout << x.first << " " << x.second << endl;
28
29      return 0;
30  }
31
32  /*
33  8
34  3 1
35  1 2
36  2 9
37  5 4
38  4 5
39  2 5
40  7 7
41  2 3
42  */

```