LAMBDA EXPRESSION λ

.:: SET 01::.

```cpp
 1  #ifndef __SORTS_H__
 2  #define __SORTS_H__
 3  #include <iostream>
 4  #include <functional>
 5  using namespace std;
 6
 7  template <typename T>
 8  void imprimirArr(T* a, int n) {
 9      cout << "[";
10      for (int i = 0; i < n; ++i) {
11          cout << (i? ", " : "") << a[i];
12      }
13      cout << "]\n";
14  }
15
16  template <typename T>
17  void intercambio(T* a, int n, bool(*mayorQue)(T, T)) {
18                                              imprimirArr(a, n); // testing step by step
19      for (int i = 0; i < n - 1; ++i) {
20          for (int j = i + 1; j < n; ++j) {
21              if (mayorQue(a[i], a[j])) {
22                  T temp = a[j];
23                  a[j] = a[i];
24                  a[i] = temp;
25                                              imprimirArr(a, n); // testing step by step
26              }
27          }
28      }
29  }
```

```cpp
template <typename T>
void burbuja(T* a, int n, function<bool(T, T)>mayorQue) {
                                                imprimirArr(a, n); // testing step by step
    for (int i = 0; i < n - 1; ++i) {
        for (int j = 0; j < n - i - 1; ++j) {
            if (mayorQue(a[j], a[j + 1])) {
                T temp = a[j];
                a[j] = a[i];
                a[i] = temp;
                                                imprimirArr(a, n); // testing step by step
            }
        }
    }
}

#endif
```

```cpp
#include <iostream>
#include <functional>
#include <stdlib.h>

#include "sorts.h"

using namespace std;

int main() {
    auto mayQ = [](float a, float b) { return a > b; };

    float a1[10],a2[10],a3[10], a4[10], a5[10];

    srand(time(0));

    for (int i = 0; i < 10; ++i) {
        a1[i] = rand() % 100;
        a5[i] = a4[i] = a3[i] = a2[i] = a1[i];
    }
    cout << "Paso a paso intercambio\n";
    intercambio<float>(a1, 10, mayQ);
    //----
    cout << "Paso a paso burbuja\n";
    burbuja<float>(a2, 10, mayQ);
    return 0;
}
```