

## Задачи:

```
// Шаблоны
template <class T>
T GetMax (T a, T b) {
    T result;
    result = (a>b)? a : b;
    return (result);
}

int main ()
{
    int i=5, j=6, k;
    long l=10, m=5, n;
    k=GetMax<int>(i,j);
    n=GetMax<long>(l,m);
}

// Векторы
std::vector<int> myvector;
for (int i=1; i<=5; i++)
    myvector.push_back(i);

for (std::vector<int>::iterator it =
    myvector.begin() ; it != myvector.end();
    ++it)
    std::cout << ' ' << *it;

// Множества
std::set<int> myset;
std::set<int>::iterator it;

// set some initial values:
for (int i=1; i<=5; i++)
    myset.insert(i*10);    // set: 10 20 30
                           40 50

it=myset.find(20);
myset.erase (it);
myset.erase (myset.find(40));

std::cout << "myset contains:";
for (it=myset.begin(); it!=myset.end();
    ++it)
    std::cout << ' ' << *it;
std::cout << '\n';
```

1. **Шаблоны 1:** Написать шаблонную функцию `T max(T x, T y)`, которая возвращает максимум двух переменных. Проверить её на переменных типа `int`, `float`, `Complex` (возможно потребуется написать/придумать оператор сравнения комплексных чисел), строке в стиле C.
2. **Шаблоны 2::** Написать функцию `char* max(char* x, char* y)`, для строк в стиле C.
3. **Шаблоны 3:** Написать шаблонную функцию `T average(T& arr[], int count)`, которая возвращает среднее значение массива переменных. Проверить её на переменных типа `int`, `float`, `Complex`.
4. **Шаблоны 4:** Написать шаблонный класс `Array<type>` – динамический массив произвольных элементов. Протестировать его.
5. **Шаблоны 5\*:** Написать шаблонный класс `template<typename type, int dims> Tensor` – класс матрицы размерности `dims`. Напишите функции умножения матрицы на число, вычисления следа (сумма диагональных элементов). Протестируйте этот класс, используйте разные типы и размерности.
6. **STL vector 1:** Используйте шаблонный класс `vector` из библиотеки `vector`. Создайте вектор из чисел 4, 8, 42, 16, 23, 42. Используйте функцию `push_back` для добавления чисел в вектор.
7. **STL vector 2:** Создайте вектор из случайных чисел размера  $10^5$ , числа от 0 до  $10^6$ . Для этого целесообразно задать размер вектора в конструкторе или с помощью функции `reserve`.
8. **STL vector 3:** Напечатайте все числа в этом векторе, меньшие 1000. Используйте 3 различных обхода вектора.
9. **STL vector 4:** Напишите функцию `int is_exists(const vector<int>& vec, int x)`, которая будет проверять есть ли в векторе число `x`.
10. **STL vector 5:** Отсортируйте вектор с помощью функции `sort` и напечатайте все числа в этом векторе.
11. **STL set 1:** Используйте шаблонный класс `set` из библиотеки `set`. Создайте вектор из чисел 4, 8, 42, 16, 23, 42. Используйте функцию `insert` для добавления чисел в множество.
12. **STL set 2:** Создайте множество и заполните его случайными числами от 0 до  $10^6$ .

13. **STL set 3:** Напечатайте все числа в этом множестве, меньшие 1000. Используйте 2 различных обхода вектора.
14. **STL set 4:** Проверьте, есть ли в этом множестве число  $x$ . Используйте функцию `find`.