**Template.cpp**

```
#include <bits/stdc++.h>

using namespace std;

#define endl '\n'
#define int long long

void solve() {}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);

    int tc = 1;
    cin >> tc;
    for (int i = 1; i <= tc; ++i) {
        cerr << "TestCase #" << i << ": \n";
        solve();
    }
}
```

**Debug.h (C++20)**

```
template<typename T>
void __print(const T& x) {
    if constexpr (is_arithmetic_v<T> or is_same_v<T, string>) cerr
<< x;
    else {
        cerr << '{';
        if constexpr (requires {x.first;})
            __print(x.first), cerr << ", ", __print(x.first);
        else {
            int f = 0; for(auto i : x) cerr << (f ++ ? ", " : ""), __print(i);}
        cerr << '}';
    }
}
template <typename... A>
void _print(const A&... a) {((__print(a), cerr << ", "), ...);}
#define debug(...) cerr << "[" << #__VA_ARGS__ << "] = [";
_print(__VA_ARGS__); cerr << "]\n";
```

**Debug.h (C++17)**

```
struct {
   template<typename T>
   void __print(const T& x) {
      if constexpr (is_arithmetic_v<T> or is_same_v<T, string>)
cerr << x;
      else {
         cerr << '{';
         int f = 0; for(auto i : x) cerr << (f ++ ? ", " : ""), __print(i);
         cerr << '}';
      }
   }
   template<typename X, typename Y>
   void __print(const pair<X, Y>& x){
      cerr << '{', __print(x.first), cerr << ", ", __print(x.second), cerr
<< '}';
   }
   template <typename... A>
   void _print(const A&... a) {((__print(a), cerr << ", "), ...);}
}_d;
#define debug(...) cerr << "[" << #__VA_ARGS__ << "] = [";
_d._print(__VA_ARGS__); cerr << "]\n";
```

**.vimrc (put in home directory ~/)**

```
inoremap jj <Esc>
```

**PBDS**

```
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
template <typename T> using ost = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
```

**FenwickTree**

```
template <typename T, const int N>
struct FenwickTree {
   T bit[N];
   int n;
   void set(int idx, T val, bool add = false) {
      for (; idx < n; idx = idx | (idx + 1))
         bit[idx] = (add ? bit[idx] : 0) + val;
   }
   T query(int r) {
      T ret = 0;
      for (; r >= 0; r = (r & (r + 1)) - 1)
         ret += bit[r];
      return ret;
   }
};
```