.*VSTU.*

Team Reference Document

01/12/2018

Contents

1. Code Templates

1.1. Basic Configuration.

1.1.1. .vimrc.

```
set cin nu ts=2 sw=2 sts=2 mouse=a
syn on

function! Compile()
    :!g++ -std=gnu++11 -g % -o %<.exe
endfunction

function! Run()
    :!time ./%<.exe
endfunction

map <F4> :call Compile()<cr>
map <F5> :call Run()<cr>
map <C-A> ggVG"+y
```

1.1.2. stress and template.

```
// g++ -std=c++11 main.cpp -o main -D"_DEBUG_TEMICH_"

#include <algorithm>
#include <functional>
#include <iostream>
#include <map>
#include <queue>
#include <set>
#include <sstream>
#include <string>
#include <vector>

using namespace std;

using LL = long long;
using pii = pair<int, int>;

#define X first
#define Y second

struct Solver {
        void solve(istream& cin, ostream& cout) {
                int a, b;
                cin >> a >> b;
                cout << a + b << endl;
        }
};

struct Brute {
        void solve(istream& cin, ostream& cout) {
                int a, b;
                cin >> a >> b;
                while (b--) ++a;
                cout << a << endl;
        }
};
template <typename Solution>
struct SolutionStr {
  string solve(string input) {
    istringstream is(input);
    ostringstream os;
    Solution().solve(is, os);
    return os.str();
  }
};

string gen_input(int it) {
  (void)it;
  return "10 20";
}

void stress() {
  for (int it = 0; it < 1000; ++it) {
    auto input = gen_input(it);
    auto brute_out = SolutionStr<Solver>().solve(input);
    auto sol_out = SolutionStr<Brute>().solve(input);
    if (sol_out != brute_out) {
      cerr << "WA #" << it << endl;
      cerr << "input: " << endl;
      cerr << input << endl;
      cerr << "expected: " << brute_out << endl;
      cerr << "got: " << sol_out << endl;
      exit(1);
    }
  }

  cerr << "OK" << endl;
}

int main() {
        #ifdef _DEBUG_TEMICH_
        stress();
        #endif
        Solver().solve(cin, cout);
}
```

2. Unordered lists

- The individual entries are indicated with a black dot, a so-called bullet.
- ahhhhhhhhhhhhhhhhhhhhhh hhhhhhhhhhhhhhhhhhhhhhhhhh hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh hhhhhhhhhhhhhhhh-hhhhhhhhhhhhhhaa
  hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh
  end
  hhhhhhh nonewline
- The text in the entries may be of any length.

(1)
(2) The numbers starts at 1 with every call to the enumerate environment.