CS 92 Competitive Programming

Silver Level

LECTURE 1: OVERVIEW OF C++ LANGUAGE FOR USACO

DR. ERIC CHOU

IEEE SENIOR MEMBER

Template

SECTION 1

Template for Command Line Input

command.cpp

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main(){
    int N=0;
    cin >> N;

    cout << N << endl;
    return 0;
}
```

Template without Calling any Function

template.cpp

```
#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;
int main(){
    ifstream fin("a.in");
    ofstream fout("a.out");
    int N=0;
    fin >> N;

    fout << N << endl;
    fin.close();
    fout.close();
    return 0;
}
```

Template Dealing with Strings

string.cpp

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <cstdlib>
using namespace std;
template <typename T> string str(const T& n){
 ostringstream stm; stm << n; return stm.str() ;
}
int main(){
    ifstream fin("a.in");
    ofstream fout("a.out");
    string s("");
    fin >> s;
    fout << s << endl;
    fin.close();
    fout.close();
    return 0;
}
```

Template Dealing with Strings

string_command.cpp

```
#include <iostream>
#include <cstdlib>
#include <sstream>
#include <string>
using namespace std;
template <typename T> string str(const T& n){
 ostringstream stm; stm << n; return stm.str() ;
}
int main(){
    string s("");
    cin >> s;
    cout << s << endl;
    return 0;
}
```

Important STL Libraries

```
#include <algorithm>
#include <vector>
#include <string>
#include <map>
#include <pair>
#include <array>
```

Advanced Template – File Mode

p.cpp

- In USACO competition, you may need to switch between file mode and console mode pretty often.
- You can use (file mode):

```
#define FILEMODE
#define PROJECT ""
```

- You can use (console mode):

```
//#define FILEMODE
#define PROJECT ""
```

Advanced Template – File Mode

p.cpp

- The files are open with re-direction from console to file. So, you may just use cout to print your data with proper setting for the modes.

```
#ifdef FILEMODE
  freopen(PROJECT ".in", "r", stdin);
  freopen(PROJECT ".out", "w", stdout);
#endif
```

Advanced Template – File Mode

p.cpp

- The files are open with re-direction from console to file. So, you may just use cout to print your data with proper setting for the modes.

```
#ifdef FILEMODE
  freopen(PROJECT ".in", "r", stdin);
  freopen(PROJECT ".out", "w", stdout);
#endif
```

Advanced Template – STL Libraries

p.cpp

- For one, if you don't care about compilation speed, you can simply include the whole standard library (C++ STL) for ease of access (however, note that this is a nonstandard library that is not available on many compilers; it is available for many contest compilers, like that of the USACO). Below is the exact include to do this:

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

using namespace std;
```

Advanced Template – STL Libraries

p.cpp

- This will allow you to use all C++ STL features without including the libraries one by one. However, when I program, I **prefer** to include each library separately, as it improves compiler speed and keeps my thoughts fairly clear. You could also pre-include a list of libraries you commonly use if you prefer to do this.

```
typedef  long  long    lint;      // long integer
typedef vector<lint> vlint;     // vector of long integer
typedef vector<int> vi;         // vector of int
typedef vector<string> vs;      // vector of string
```

Advanced Template – Constants/Macros

p.cpp

```
#define MAXINT      2147483647
#define MININT      -2147483648
#define MOD         1000000007
#define pb          push_back
#define mp          make_pair
#define space       << " " <<
#define spacef      << " "
#define newline     << "\n"
```

Advanced Template – Constants/Macros

p.cpp

```
#define fo(i,a,b)       for(lint i = a; i <= b; i++)
#define foo(i,a,b,d) for(lint i = a; i <= b; i+=d)
#define print(x)       for(auto i : x) cout << i spacef
#define println(x)    for(auto i: x) cout << i spacef; cout newline;
#define reset(x, n)  for (int i=0; i<n; i++) x[i]=0;
#define mmax(x,i)  x = max(x,i)
#define mmin(x,i)   x = min(x,i)
```

Training Strategy

SECTION 1

USACO Bronze for Basic C++ Competitive Programming

- It takes a while to master C++ for competitive programming, especially for students who learned Java before C++.
- Therefore, rewrite USACO Bronze programs using C++ is a good way to improve C++ mastery.

Important Data Structures

- String
- Array
- Struct
- Vector
- Set
- Map
- Pair
- Class
- Math
- Algorithm (Sort/Search)
- Pointers
- Upper bounds
- Number

Problem 1

SECTION 1

USACO 2014 December Contest, Bronze
Problem 2. Crosswords

Return to Problem List

Contest has ended.

Submitted; Results below show the outcome for each judge test case

*	*	*	*	*	*	*	*	*	*
1 1.2mb 1ms	2 1.2mb 1ms	3 1.2mb 1ms	4 1.2mb 1ms	5 1.2mb 1ms	6 1.2mb 1ms	7 1.2mb 1ms	8 1.2mb 1ms	9 1.2mb 9ms	10 1.2mb 1ms

Problem

INPUT: (file crosswords.in)
The first line of input contains N and M separated by a space.
The next N lines of input each describe a row of the grid. Each contains M characters, which are either '.' (a clear cell) or '#' (a blocked cell).

SAMPLE INPUT:

```
5 3
...
#..
...
..#
.##
```

Process

Step 1: We determine if a each cell begins a horizontal or vertical clue. If a cell begins a horizontal clue, it must be clear, its neighboring cell to the left must be blocked or outside the crossword grid, and the two cells on its right must be clear (that is, a horizontal clue can only represent a word of 3 or more characters).

The rules for a cell beginning a vertical clue are analogous: the cell above must be blocked or outside the grid, and the two cells below must be clear.

Process

Step 2: We assign a number to each cell that begins a clue. Cells are assigned numbers sequentially starting with 1 in the same order that you would read a book; cells in the top row are assigned numbers from left to right, then the second row, etc. Only cells beginning a clue are assigned numbers.

Detection of the Clue Heads

```
...        !!!        123
#..        #..        #..
...        !..        4..
..#        ..#        ..#
.##        .##        .##
```

Loading 2D Character Array

```
int row = 0, col=0;
cin >> row >> col;
char m[row][col];
fo(i, 0, row){
    string s = "";
    cin >> s;
    fo(j, 0, col){
      m[i][j] = s.at(j);
    }
}
```

Traverse Through the Game Board

```
int c=0;
fo(i, 0, row){
   fo(j, 0, col){
     if (m[i][j] == '.'){ // clear
       if ((j==0|| m[i][j-1]=='#') &&
            j<col-2 && m[i][j+1] =='.' && m[i][j+2]=='.') {
          c++; m[i][j] = '!'; continue;
         }
       if ((i==0|| m[i-1][j]=='#') &&
          i<row-2 && m[i+1][j] =='.' && m[i+2][j]=='.') {
          c++; m[i][j] = '!'; continue;
       }
     }
   }
}
```

Output for the Results:

```
cout << c newline;
fo(i, 0, row){
    fo(j, 0, col){
      if (m[i][j]=='!') cout << (i+1) space (j+1) newline;
    }
}
```