P THINK FAST S

# Competitive Programming

From Problem 2 Solution in O(1)

## Online Judges

**Mostafa Saad Ibrahim**
PhD Student @ Simon Fraser University

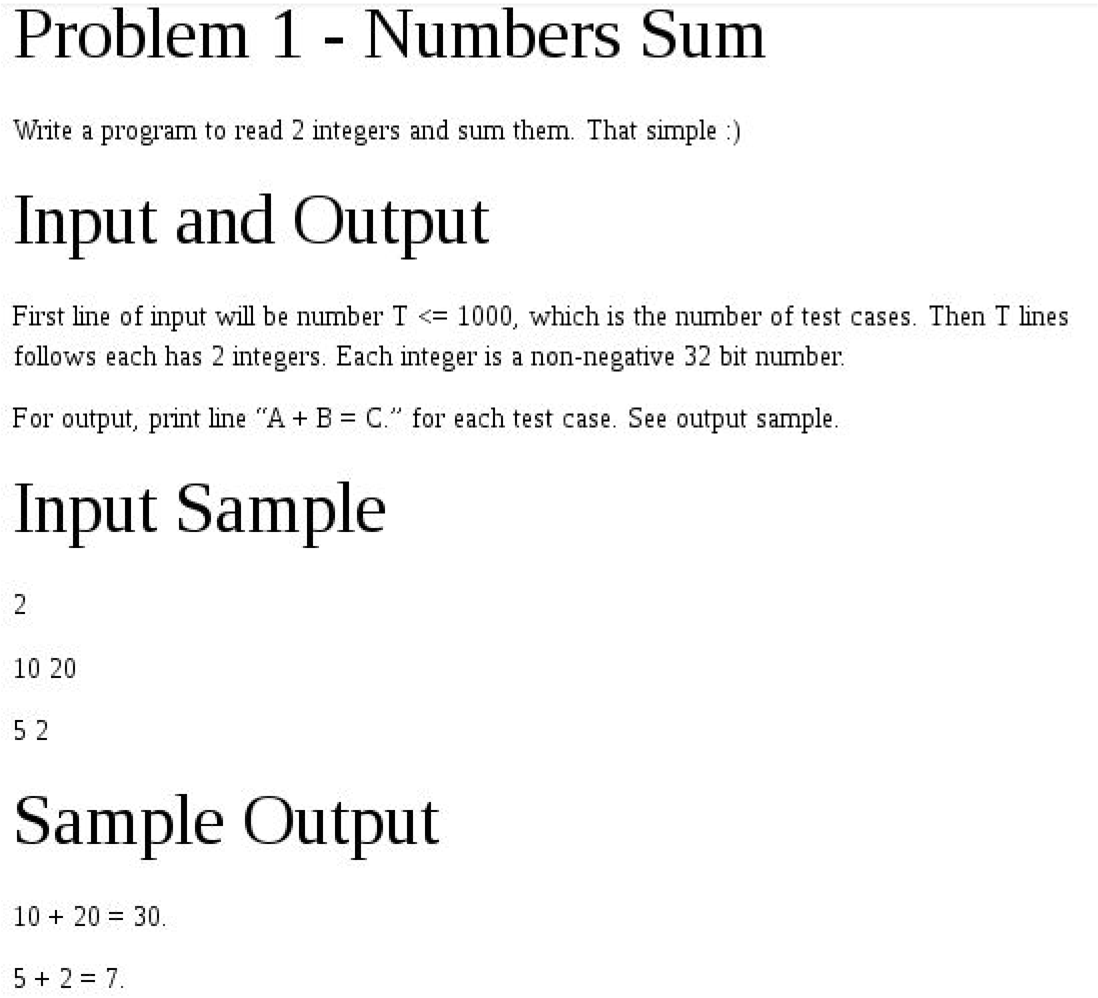
# Online Judges

- An online system to evaluate **automatically** your submitted solutions.
- You submit **code** for some problem using some **language** (e.g. c++)
- OJ compiles code against this language
- If code compiled, it runs it against MANY test cases invented by problem setter
- The OJ compares **your results** with the **problem setter results**...and accept/reject

# Online Judges

- Topcoder and Codeforces
  - Frequent contests. Each is 2 divisions / sorted problems.
  - They are also the best for a **junior**
- UVA, SPOJ, LiveArchive, Timus, PKU, ZOJ, TJU, SGU, CodeJam, Project Euler, MORE
  - Useful for **semi-senior**
- Codechef, Hackerrank, Codility
  - They are growing and seems will be destination for many
- Usaco (IOI style)
  - May beuseful for **senior** to try. Blocking Sorted Problems.

# Problem Statement



## Problem 1 - Numbers Sum

Write a program to read 2 integers and sum them. That simple :)

## Input and Output

First line of input will be number T <= 1000, which is the number of test cases. Then T lines follows each has 2 integers. Each integer is a non-negative 32 bit number.

For output, print line "A + B = C." for each test case. See output sample.

## Input Sample

```
2

10 20

5 2
```

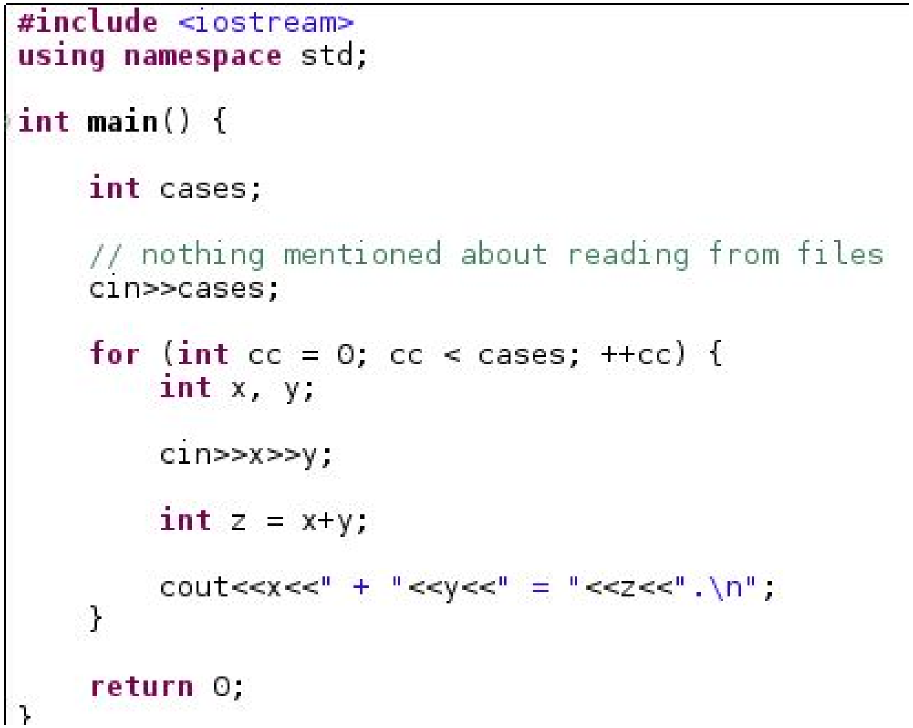
## Sample Output

```
10 + 20 = 30.

5 + 2 = 7.
```

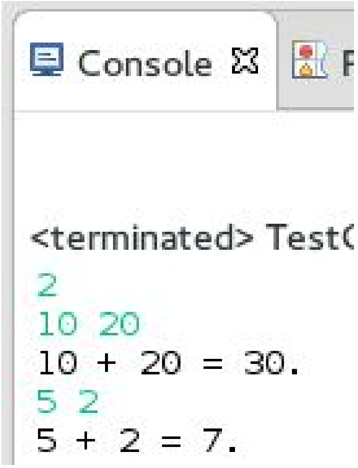
# Following the statement

- It is very important to realize these 2 points
  - Judge is automatic
  - It compares your result with predefined results
- If problem says read from file named input.txt, then you must read from it. If it says read from console, follow. If says write to file named sum.txt...follow...or your code fails
- When it says print A + B = C.
  - Then must print number, “ + ” number,” = “, number, “.”
  - FOLLOW strictly what it asks you to do it.

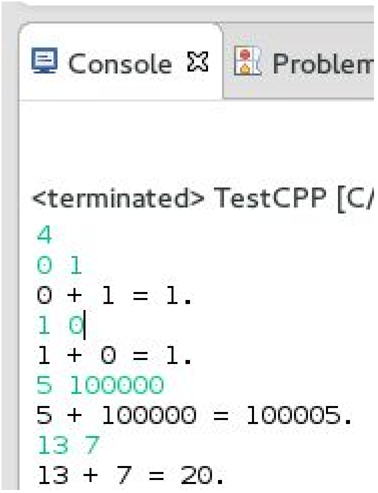
# Let’s write a solution



```
#include <iostream>
using namespace std;

int main() {

    int cases;

    // nothing mentioned about reading from files
    cin>>cases;

    for (int cc = 0; cc < cases; ++cc) {
        int x, y;

        cin>>x>>y;

        int z = x+y;

        cout<<x<<" + "<<y<<" = "<<z<<".\n";
    }

    return 0;
}
```



```
<terminated> Test(
2
10 20
10 + 20 = 30.
5 2
5 + 2 = 7.
```

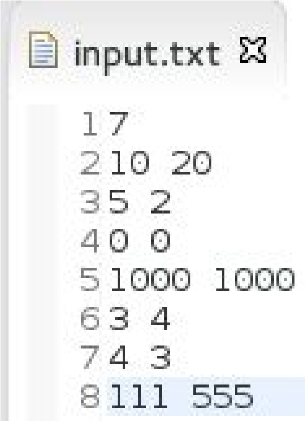


```
<terminated> TestCPP [C/
4
0 1
0 + 1 = 1.
1 0
1 + 0 = 1.
5 100000
5 + 100000 = 100005.
13 7
13 + 7 = 20.
```

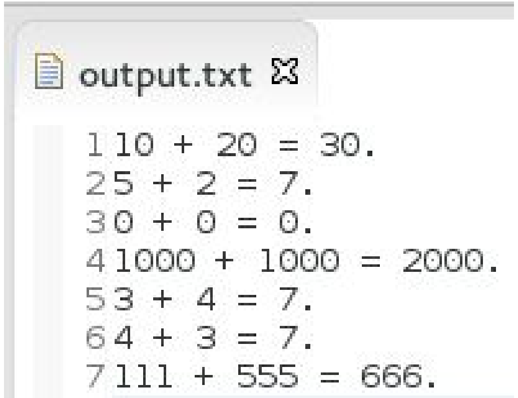
# Problem setter test cases

- Sample input is usually 2-3 cases
- However, the real success is to do well on the hidden test cases of the problem setter
- The cases ensure your code is really correct and consider time/memory/all scenarios
- Some online judges never tell you the test cases (e.g. UVA), while others tell you after the contest (e.g. Topcoder and Codeforces)

# Problem setter test cases



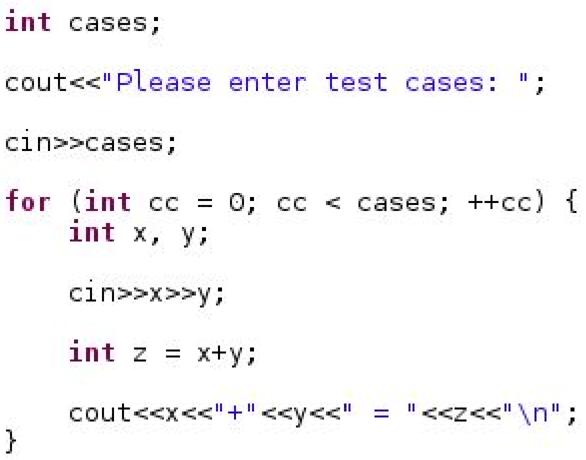
```
7
10 20
5 2
0 0
1000 1000
3 4
4 3
111 555
```



```
10 + 20 = 30.
5 + 2 = 7.
0 + 0 = 0.
1000 + 1000 = 2000.
3 + 4 = 7.
4 + 3 = 7.
111 + 555 = 666.
```

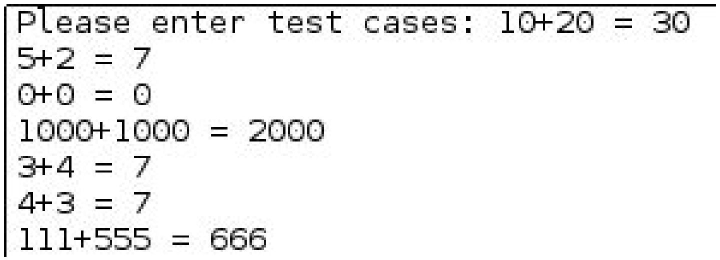
If your output is SAME as judge test cases, you get
**Accepted Answer**
otherwise
**Rejected Answer**

# Correct logic, but rejected



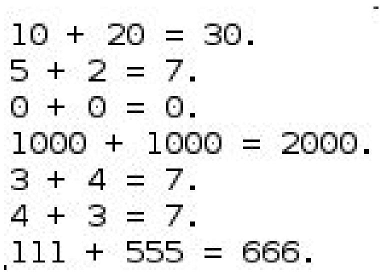
```
int cases;

cout<<"Please enter test cases: ";

cin>>cases;

for (int cc = 0; cc < cases; ++cc) {
    int x, y;

    cin>>x>>y;

    int z = x+y;

    cout<<x<<"+"<<y<<" = "<<z<<"\n";
}
```

Your output



```
Please enter test cases: 10+20 = 30
5+2 = 7
0+0 = 0
1000+1000 = 2000
3+4 = 7
4+3 = 7
111+555 = 666
```

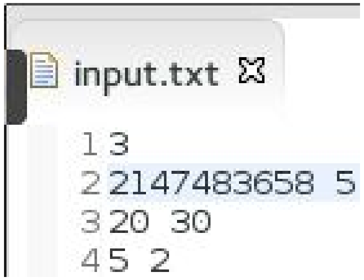
Judge predefined output



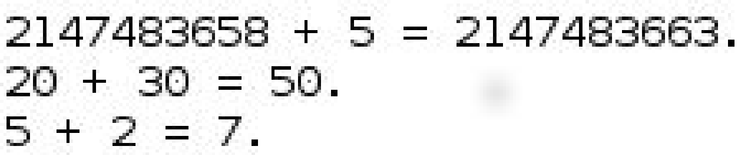
```
10 + 20 = 30.
5 + 2 = 7.
0 + 0 = 0.
1000 + 1000 = 2000.
3 + 4 = 7.
4 + 3 = 7.
111 + 555 = 666.
```

# Challenging Cases - 1

Judge Input and Output

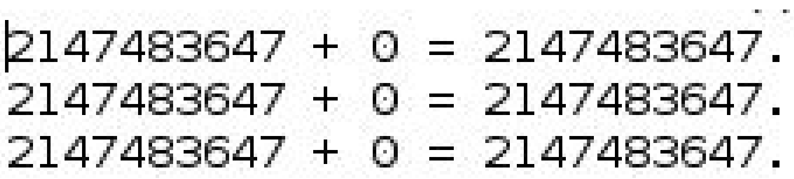


```
3
2147483658 5
20 30
5 2
```



```
2147483658 + 5 = 2147483663.
20 + 30 = 50.
5 + 2 = 7.
```

Your output



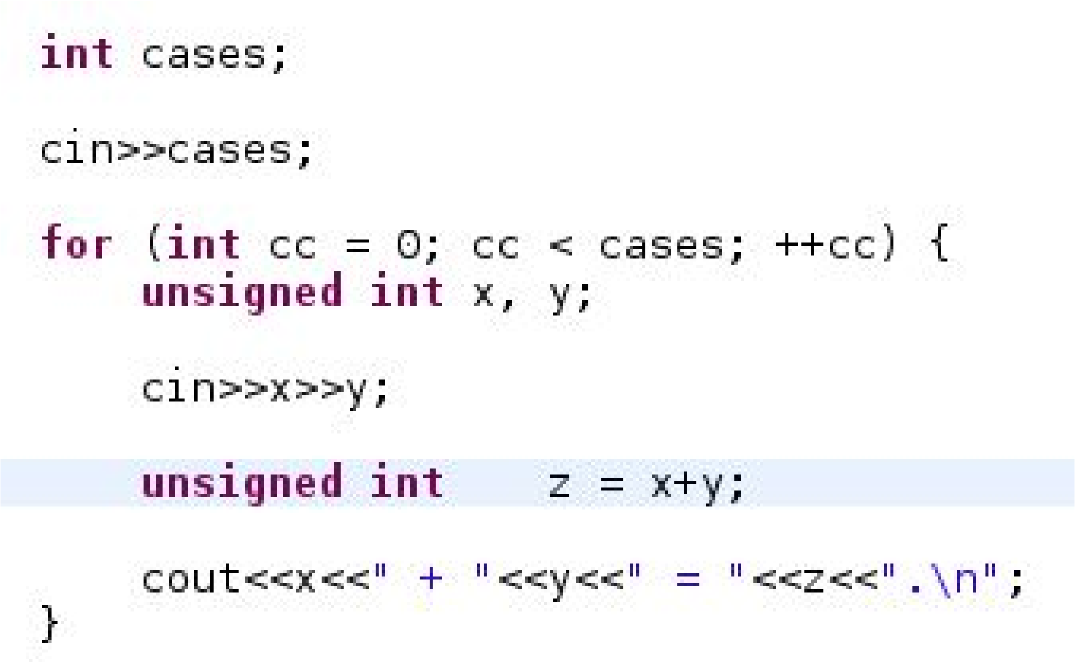
```
2147483647 + 0 = 2147483647.
2147483647 + 0 = 2147483647.
2147483647 + 0 = 2147483647.
```

What is wrong?!

Problem statement said non negative integers in 32 bit

This means input is unsigned integer (bigger range) NOT signed integer!

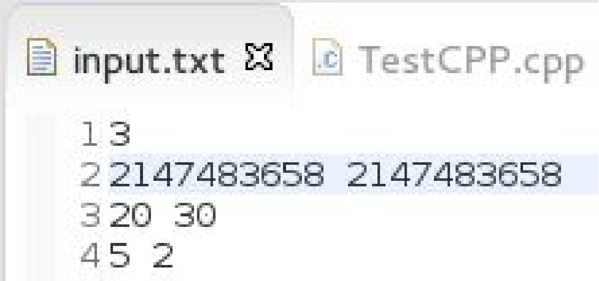
# Code Fix 1



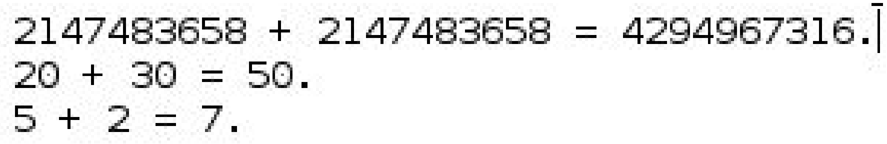
```
int cases;

cin>>cases;

for (int cc = 0; cc < cases; ++cc) {
    unsigned int x, y;

    cin>>x>>y;

    unsigned int    z = x+y;

    cout<<x<<" + "<<y<<" = "<<z<<".\n";
}
```

# Challenging Cases - 2

Judge Input and Output

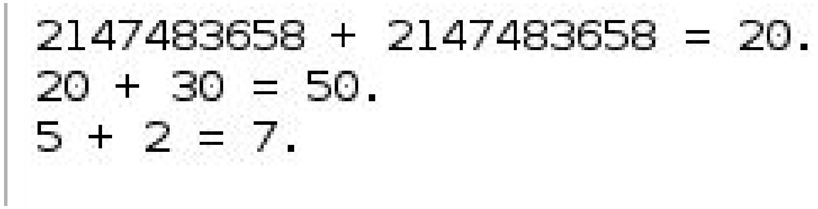


```
3
2147483658 2147483658
20 30
5 2
```



```
2147483658 + 2147483658 = 4294967316.
20 + 30 = 50.
5 + 2 = 7.
```

Your output



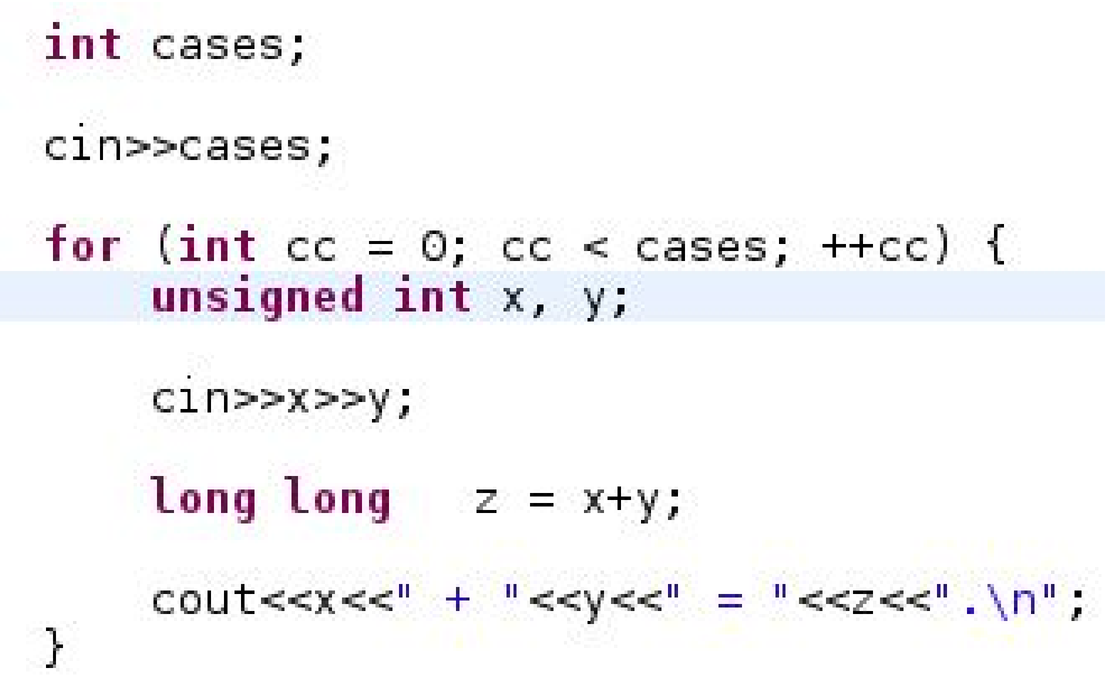
```
2147483658 + 2147483658 = 20.
20 + 30 = 50.
5 + 2 = 7.
```

What is wrong?!

When we sum 2 unsigned integer, then result will be bigger than unsigned integer!

We need z to be of something bigger….use long long

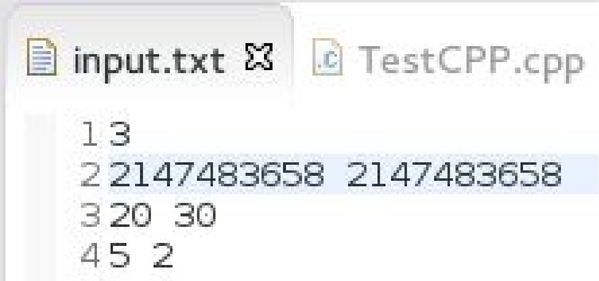
# Code Fix 2



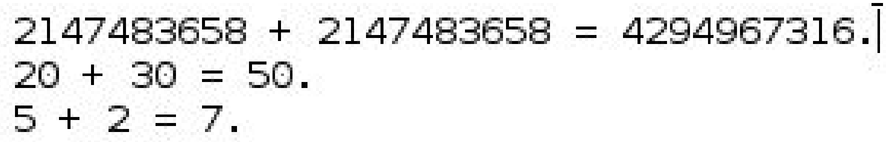
```
int cases;

cin>>cases;

for (int cc = 0; cc < cases; ++cc) {
    unsigned int x, y;

    cin>>x>>y;

    long long   z = x+y;

    cout<<x<<" + "<<y<<" = "<<z<<".\n";
}
```

# Challenging Cases - 2 again

Judge Input and Output

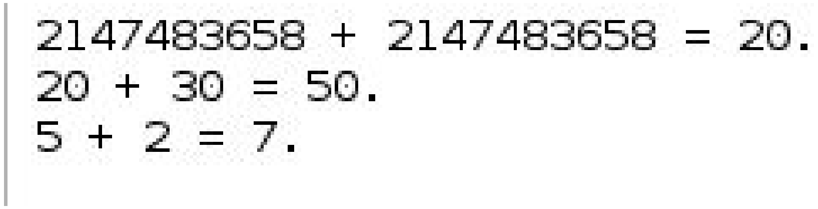


```
3
2147483658 2147483658
20 30
5 2
```



```
2147483658 + 2147483658 = 4294967316.
20 + 30 = 50.
5 + 2 = 7.
```

Your output



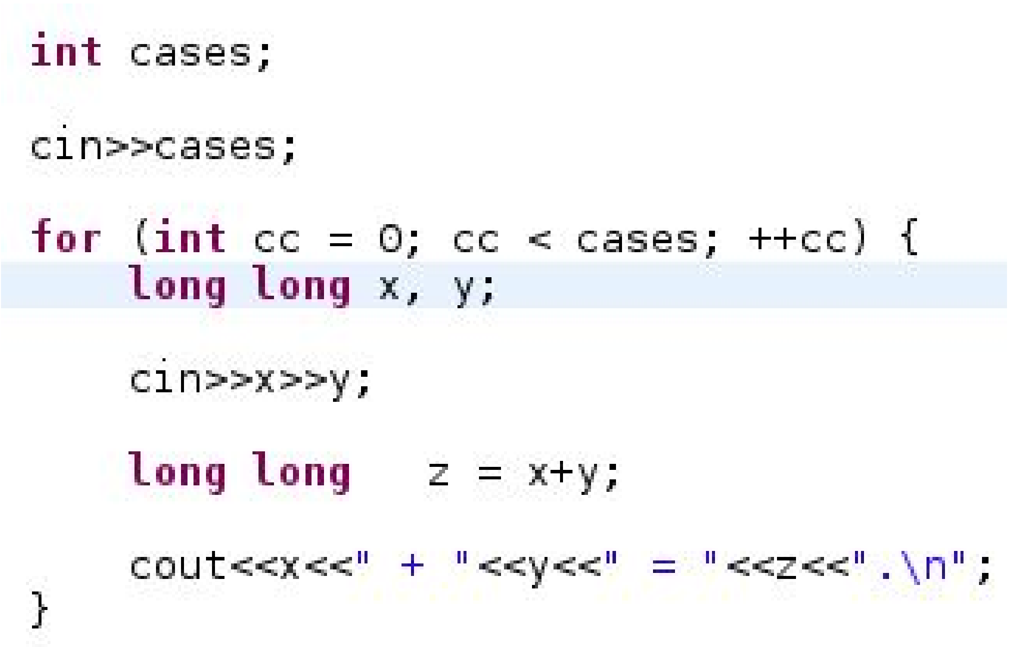
```
2147483658 + 2147483658 = 20.
20 + 30 = 50.
5 + 2 = 7.
```

We fixed that!
What is wrong!!!!!!!

Sum of the 2 unsigned will be unsigned..no automatic conversion will happen

Either use long long for x, y or CAST to long long

# Code Fix 3



```
int cases;

cin>>cases;

for (int cc = 0; cc < cases; ++cc) {
    long long x, y;

    cin>>x>>y;

    long long   z = x+y;

    cout<<x<<" + "<<y<<" = "<<z<<".\n";
}
```

# Done :)

- You got many rejections for a trivial problem!
  - Mistake in input data type
  - Mistake in output data type
  - Mistake in data types conversions!
- Competitions makes you perfect at these things
- You learn to be careful while coding
- And more gains as we said before :)

# Printing Blank Lines

- Be careful from exact nature of printing between test cases.
- Print each test case answer on line
  - cout<<case_answer<<”\n”;
- Print blank line after each test case
  - cout<<case_answer<<”\n\n”;
- Print blank line between consecutive test cases
  - cout<<case_answer<<”\n”;
  - if(not last test case)
    - cout<<”\n”;

# Printing Separating Spaces

- Let say space is * (for clarification)
- What if problems ask you to print numbers from 1 to 5. Print space after each number.
  - Answer 1: 1*2*3*4*5*
- What if problems ask you to print numbers from 1 to 5. Print space **between** each number.
  - Answer 2: 1*2*3*4*5
- **Take home message**: Be careful from printing instructions to avoid crazy WAs

# Judge responses

- Judge actually sends different type of other responses, based on code status
- Accepted (AC), Wrong Answer (WA)
- Compile Error (CE)
  - Your code doesn’t compile. Probably using wrong language in your choice..or wrong compiler version
- Presentation Error (PE)
  - If your program is wrong such that calculation is correct but displaying is wrong (e.g. missing dot, extra line)
  - However, judge may also send WA (wrong answer)

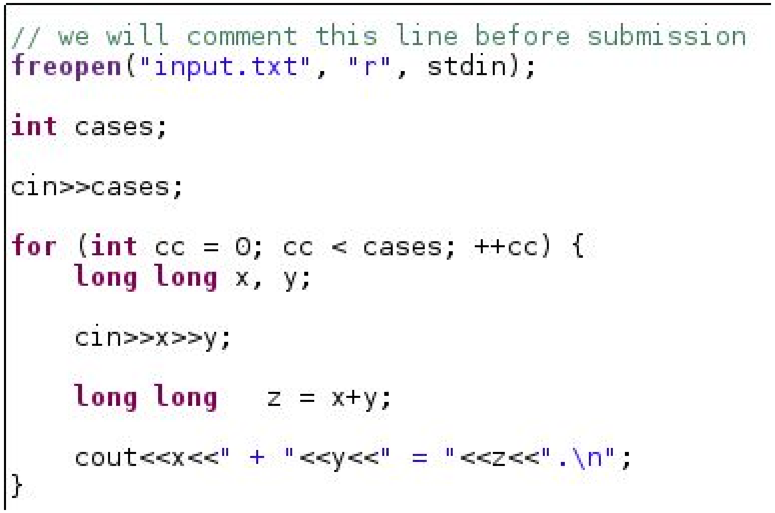
# Judge responses

- Time Limit Exceeded (TLE)
  - You need time more than allowed by judge
- Memory Limit Exceeded (MLE)
  - Reserved too much memory (e.g. 1e9) or memory leak
- Run Time Error (RTE)
  - Code crashes...E.g. Index out of boundary
- Above list are most common, but some OJs may define their own one. See UVA list

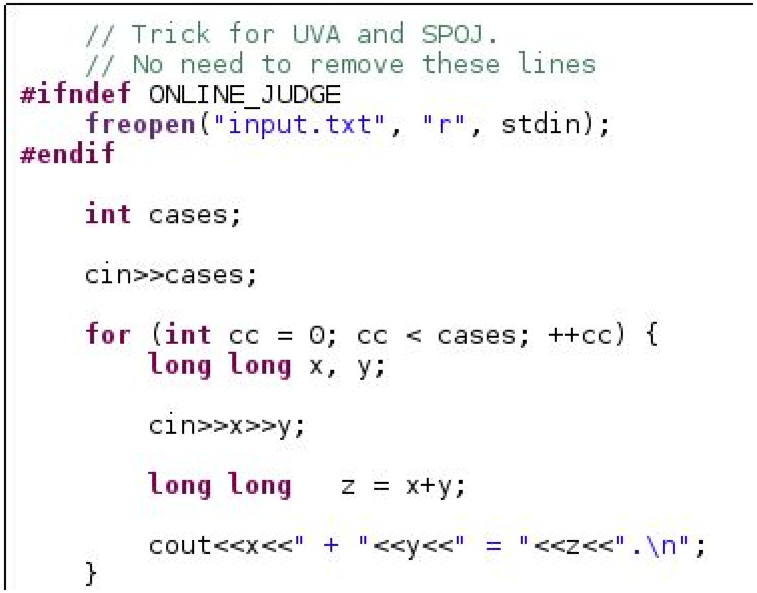
# freopen for easy testing

- Putting test cases on file and increasing them is very practical instead of keep writing them.
- If problem already asks to read from file, good
- Otherwise, freopen can switch cin stream to filestream
- We use this trick to use files always, and before submission comment file line

# freopen for easy testing



```
// we will comment this line before submission
freopen("input.txt", "r", stdin);

int cases;

cin>>cases;

for (int cc = 0; cc < cases; ++cc) {
    long long x, y;

    cin>>x>>y;

    long long   z = x+y;

    cout<<x<<" + "<<y<<" = "<<z<<".\n";
}
```



```
    // Trick for UVA and SPOJ.
    // No need to remove these lines
#ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
#endif

    int cases;

    cin>>cases;

    for (int cc = 0; cc < cases; ++cc) {
        long long x, y;

        cin>>x>>y;

        long long   z = x+y;

        cout<<x<<" + "<<y<<" = "<<z<<".\n";
    }
```

# Codeforces VS Topcoder

- As we said, these 2 sites are very helpful for juniors. Problems ordered by level.
  - You need to work on both...which to start with?
- They are helpful too for seniors, however
  - Seniors, must work on other judges to see variety of types
  - Most important, you must train to identify problem level by yourself.
  - UVA, SPOJ, LiveArchive, Timus are good selection

# Codeforces VS Topcoder

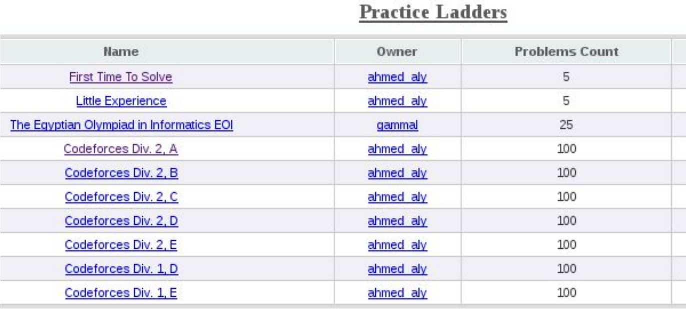
- Topcoder
  - A 75 minutes SRM with 3 problems for 2 divisions
  - is a better start for juniors? [debate]
  - No read/write. Just function to implement
  - Very clear problem statement. Strong Editorials (old ones)
  - Great forums to get help
  - More encourage on contestant speed
  - Little problem: It uses arena NOT web based
    - Once you setuped it, no problems any more
    - Gives you space to chat with others
    - They have a web interface, but not so good

# Codeforces VS Topcoder

- Codeforces
  - A 2 hours round with 5 problems for 2 divisions
  - Nowadays, Very regular than topcoder
  - Problem variety is bigger & better, due to less limits
  - Problems ideas less time to repeat (vs TC more repeat)
  - Style close to ACM ICPC
    - ICPC testing is over 1 file usually of all test cases. So time limit is overall limit. Codeforces time limit per problem.
    - Codeforces allows many languages..ICPC C/C++/Java
  - Better Hacking (Challenge code) rules [debate].
  - Virtual Competitions

# A2oj Online Judge

- Created by Ahmed Aly, a senior
- Monitor your submissions for Other online judges
- Can create virtual contests, contests corss other online judges..and more
- Has ladder feature for beginners
  - Sorted blocking problems
  - Like USACO..but better



Practice Ladders

| Name | Owner | Problems Count |
|---|---|---|
| First Time To Solve | ahmed_aly | 5 |
| Little Experience | ahmed_aly | 5 |
| The Egyptian Olympiad in Informatics EOI | gammal | 25 |
| Codeforces Div. 2, A | ahmed_aly | 100 |
| Codeforces Div. 2, B | ahmed_aly | 100 |
| Codeforces Div. 2, C | ahmed_aly | 100 |
| Codeforces Div. 2, D | ahmed_aly | 100 |
| Codeforces Div. 2, E | ahmed_aly | 100 |
| Codeforces Div. 1, D | ahmed_aly | 100 |
| Codeforces Div. 1, E | ahmed_aly | 100 |

# تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً