Your Exceptional Server submission got 30.00 points. Share Tweet

Try the Next Challenge

Exceptional Server

by Dalimil

Problem Submissions Leaderboard Discussions

Your friend set up a small computational server that performs complex calculations.
It has a function that takes 2 large numbers as its input and returns a numeric result. Unfortunately, there are various exceptions that may occur during execution.

Complete the code in your editor so that it prints appropriate error messages, should anything go wrong. The expected behavior is defined as follows:

- If the *compute* function runs fine with the given arguments, then print the result of the function call.
- If it fails to allocate the memory that it needs, print `Not enough memory`.
- If any other standard C++ exception occurs, print `Exception: S` where S is the exception's error message.
- If any non-standard exception occurs, print `Other Exception`.

Input Format

The first line contains an integer, T, the number of test cases.
Each of the T subsequent lines describes a test case as 2 space-separated integers, A and B, respectively.

Constraints

$1 \leq T \leq 10^3$
$0 \leq A, B \leq 2^{60}$

Output Format

For each test case, print a single line containing whichever message described in the *Problem Statement* above is appropriate. After all messages have been printed, the locked stub code in your editor prints the server load.

Sample Input

```
2
-8 5
1435434255433 5
```

Sample Output

```
Exception: A is negative
Not enough memory
2
```

Explanation

-8 is negative, hence 'Exception: A is negative' is thrown. Since the second input is too large, 'not enough memory' is displayed. 2 is the server load.

Submitted 9405 times

Medium | Max Score 30

Need Help?

View Discussions

View Top Submissions

Rate This Challenge:

Download problem statement

Download sample test cases

Suggest Edits

Current Buffer (saved locally, editable) C++14

```
#include <iostream>
#include <exception>
#include <string>
#include <stdexcept>
#include <vector>
#include <cmath>
using namespace std;

class Server {
private:
    static int load;
public:
    static int compute(long long A, long long B) {
        load += 1;
        if(A < 0) {
            throw std::invalid_argument("A is negative");
        }
        vector<int> v(A, 0);
        int real = -1, cmplx = sqrt(-1);
        if(B == 0) throw 0;
        real = (A/B)*real;
        int ans = v.at(B);
        return real + A - B*ans;
    }
    static int getLoad() {
        return load;
    }
};
int Server::load = 0;

int main() {
    int T; cin >> T;
    while(T--) {
        long long A, B;
        cin >> A >> B;
try
        {
            std::cout<< Server::compute(A,B)<<std::endl;
        }
        catch (std::bad_alloc& error)
        {
            std::cout<<"Not enough memory"<<std::endl;
        }
        catch (std::exception& error)
        {
            std::cout<<"Exception: "<<error.what()<<std::endl;
        }
```

```
48          catch(...)
49          {
50              std::cout<<"Other Exception"<<std::endl;
51          }
52      }
53      cout << Server::getLoad() << endl;
54      return 0;
55  }
```

Line: 42 Col: 42

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

Challenge your friends:

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9

You've earned 30.00 points. Next Challenge