Problem Solving Guidelines

North South University

In a typical programming contest or online judges you are provided with some problems and you have to give your best to solve those one by one. You have to write a solution in any of your preferred programming language and submit that solution for testing. Your solution is considered correct if it follows all the requirements and produce correct answers for given inputs. The more problems you solve the more better rank you get in the contest. Please go through the below details to learn about problem solving strategy.

Solving a Problem

After reading a problem, if you think you have got the solution for that, you have to write your code in the editor(Ex: Codeblocks). For each problem, you'll be given some sample test cases. After you are done with your coding, please check if all the sample test cases matches with your output. If it matches and you are confident with your solution then you have to submit it online to judge the correctness. You can find the submit option in the specific problem page.

What Happens When You Submit a Program?

When you submit a code for a problem, your code gets tested in the judge server to check the correctness of the program. Normally the judges create multiple test files for each of the problems. After receiving a submission, the judge server runs the program for each of the test files separately and check if your code's working perfectly for all of them. Depending on whether your code fails to pass any of the test files, then different types of verdicts are given.

1. Wrong Answer (WA): Your code showed wrong output.
2. Time Limit Exceeded (TLE): Your code could not finish processing the result in the given time.
3. Memory Limit Exceeded (MLE): Your code tried to use more memory than the given limit.
4. Runtime Error (RTE): Your code tried to access unallocated memory or tried to perform illogical operations or ran out of stack memory.
5. Compilation Error (CE): Your code doesn't follow the required syntax of the selected language.
6. Accepted (AC): Your code worked perfectly for each of the test files.

Input Specifications

For each problem, the required input specifications are given separately. Which types of things you'll take as input are explicitly explained in the description. You have to follow the required structure, otherwise, you'll get the wrong verdict. For example, if a program asks you to take a single integer from the input but you wrote a code which takes two integers. Then it'll get a wrong verdict.

Output Specifications

When you are printing something please follow the required specifications given in the problem. For example, if a problem asks you to print the result which is just an integer, please print only the integer. Nothing else. Don’t print things like this: “Please Enter A Number” or “Here’s the Output”. It’ll get a wrong verdict regardless of whether your program worked perfectly or not. Always print a newline character(‘\n’) at the end of the output.

Memory Limit Constraints

For each of the problems, you are given a memory limit. If you use memory more than the limit then you’ll get Memory Limit Error. Generally, the limit is given in MB. One integer takes 4/(10^6) MB. So you can easily calculate how many integer variables you can declare following the memory restriction.

Time Limit Constraints

For each of the problems, you are given a time limit. If your program fails to show the desired output before the required time, it’ll get Time Limit Error. **Generally, 10^8 operations take around 1 second time**.

Tips for the Contestants

1. Try to solve the easier problems first. Generally, the problems which have been solved by more contestants are relatively easier than other problems. You can find the statistics for each of the problems on the dashboard of the contest.
2. Don’t get stuck in a problem for a long time. If you are struggling with a problem, try to switch to a new one and come back to this one later.
3. Before starting the code, discuss your solution idea with your teammates. It really helps to find the correctness of your solution.
4. Don’t worry about the position of your team in the standings in the middle of the contest. Remember, the contest runs for 5 hours. So the final position of your team after this 5 hours is what actually matters.
5. When you get any wrong verdict, tell your teammates to check your code to find the bug. Don’t rush. It’s very natural to get wrong verdicts. You just have to put a little more of your efforts to make it Accepted.

Please look into the sample problems and their solutions to understand the correct procedure.

Sample Problem 1

There will be T (1 <= T <= 10) test cases.
For each case, you’ll be given an integer X (1 <= X <= 100). Just print the number.

Correct C++ Code:

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

int main(){
    int T, tc, X;
    scanf("%d", &T);
    for (tc = 0; tc < T; tc++) {
        scanf("%d",&X);   // see how we take exactly one number per case?
        printf("%d\n",X); // notice how we printed only x and nothing else.
                          // also see how we use the newline at the end of
                          // the output.
    }
    return 0;
}
```

Wrong C++ Code:

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

int main(){
    int T, tc, X;
    scanf("%d", &T);
    for (tc = 0; tc < T; tc++) {
        printf("Please Enter a Number"); // Wrong, the problem didn't ask you to
                                         // print this.

        scanf("%d",&X);
        printf("Here's the Number"); // Wrong, the problem didn't ask you to
                                     // print this.

        printf("%d\n",X);
    }
    return 0;
}
```

Sample Problem 2

You’ll be given two integers X and Y, just print the sum of X and Y. Here (1 <= X, Y <= 100).

Correct C++ Code:

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

int main(){
    int X, Y;
    scanf("%d %d",&X,&Y);
    int sum = X+Y;
    printf("%d\n",sum);
}
```

Wrong C++ Code:

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

int main(){
    int X, Y, Z;
    scanf("%d %d %d",&X,&Y,&Z); // Wrong, the problem asked to take input of
                                // just two numbers
    int sum = X+Y;
    printf("%d",sum);           // Wrong, you didn't print the new line
                                // character('\n') at the end
    return 0;
}
```

Sample Problem 3

Print “North South University” on a single line. No input is provided.

Correct C++ Code:

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

int main(){
    printf("North South University\n");
    return 0;
}
```

Sample Problem 4

You will be given an integer N in the first line. In the next line, you’ll be given N integers. Please print the maximum value among those N integers.

Correct C++ Code:

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

```

```
int main(){
    int N;
    scanf("%d", &N);
    int A[N+10]; /// Look here, we took the array size N+10 for safety.
    for(int i = 0; i<N; i++){
        scanf("%d", &A[i]);
    }
    int maxNum = A[0];
    for(int i = 0; i<N; i++){
        maxNum = max(maxNum, A[i]);
    }
    printf("%d\n", maxNum);
    return 0;
}
```

Sample Problem 5

You will be given three integers L, R and K. You have to print the number of integers in the range L to R which gets divided by K.

Correct C++ Code:

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

int main(){
    int L, R, K;
    scanf("%d %d %d", &L, &R, &K);
    int result = 0;
    for(int i = L; i<=R; i++){
        if(i%K == 0){
            result++;
        }
    }
    printf("%d\n", result);
    return 0;
}
```