

CMU Programming Contest Rules

Allowed Materials

- Participants are allowed to bring paper copies of APIs, notes, textbooks, and code with them.
- No external electronic data may be accessed at any time. This includes phones, USB drives, the internet, etc.
- Teams may not consult with coaches or other external sources during the competition.

Contest Rules

- The contest will consist of 8 problems to be solved in 3 hours.
- Allowed programming languages are C, C++, Java, and Python.
- The problems may be solved in any order.
- All solutions will be submitted electronically. Solutions will be tested by the judges and feedback will be sent accordingly.
- If a question is unclear, participants may request clarification from the judges. If a clarification is warranted, it will be posted on the contest site.
- Participants may print out their code. Any printed pages will be delivered by a contest volunteer. Participants should not pick up their printouts themselves.
- Any unauthorized behavior will result in automatic disqualification. This includes but is not limited to: looking at other teams' work/solutions, using disallowed external resources, or attempting to hack the contest environment.
- Judges decisions are final.

Program Output

- All questions must read data from standard input and write to standard output. (Examples included at the end)
- Output will only be accepted if it is formatted exactly as specified in the problem statement (No preceding or trailing spaces, extra characters, typos, etc.)

- There is a runtime limit specified at the top of the page on all problems. Any programs running longer than this time will be cut-off.

Judge Feedback

- After receiving solutions, the following types of feedback will be sent back:
 - o Accepted: The solution is correct.
 - o Compilation Error: The program did not compile.
 - o Run-time Error: The program raised an error during runtime.
 - o Incorrect Output: The solution outputs the wrong answer for a test case
 - o Time Limit Exceeded: The program ran over the time limit and was cut-off.

Scoring

- Teams are ranked based on number of problems solved correctly.
- In the event of a tie, the following tiebreaker is used:
 - o For each correct submission, teams will receive 1 point per minute taken to solve the problem, plus an additional 20 point penalty for each incorrect submission (not including compilation errors) of that problem.
 - o Incorrect submissions for unsolved problems do not count towards a team's points.
 - o The team with fewer points will be deemed the winner.
- In the case of a tie in both of the previous methods, the winner will be determined by solution elegance determined at the judges' discretion. (This is extremely unlikely, and you should not worry too much about this).

Standard Input/Output

Below are some examples of how to read from standard input and write to standard output, which you will need to do for each problem.

Solutions to the following problem.

Write a program that takes a single integer, n , followed by n lines, and outputs n lines, each containing the corresponding input line doubled.

Solution in C

```
#include <stdio.h>

int main(void)
{
    int datalines, number, result;
    scanf("%d", &datalines);

    for (int i = 0; i < datalines; i++)
    {
        scanf("%d", &number);
        result = number * 2;
        printf("%d\n", result);
    }

    return 0;
}
```

Solution in C++

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

int main()
{
    int datalines, number, result;
    cin >> datalines;

    for (int i = 0; i < datalines; i++)
    {
        cin >> number;
        result = number * 2;
        cout << result << endl;
    }
}
```

```
    }  
    return 0;  
}
```

Solution in Java

```
import java.util.*;  
  
public class Problem1  
{  
    public static void main(String[] args)  
    {  
        int datalines, number, result;  
  
        Scanner in = new Scanner(System.in);  
        datalines = in.nextInt();  
  
        for (int i = 0; i < datalines; i++)  
        {  
            number = in.nextInt();  
            result = number * 2;  
            System.out.println(result);  
        }  
    }  
}
```

Solution in Python 3

```
datalines = int(input())  
for i in range(datalines):  
    number = int(input())  
    result = number * 2  
    print(result)
```