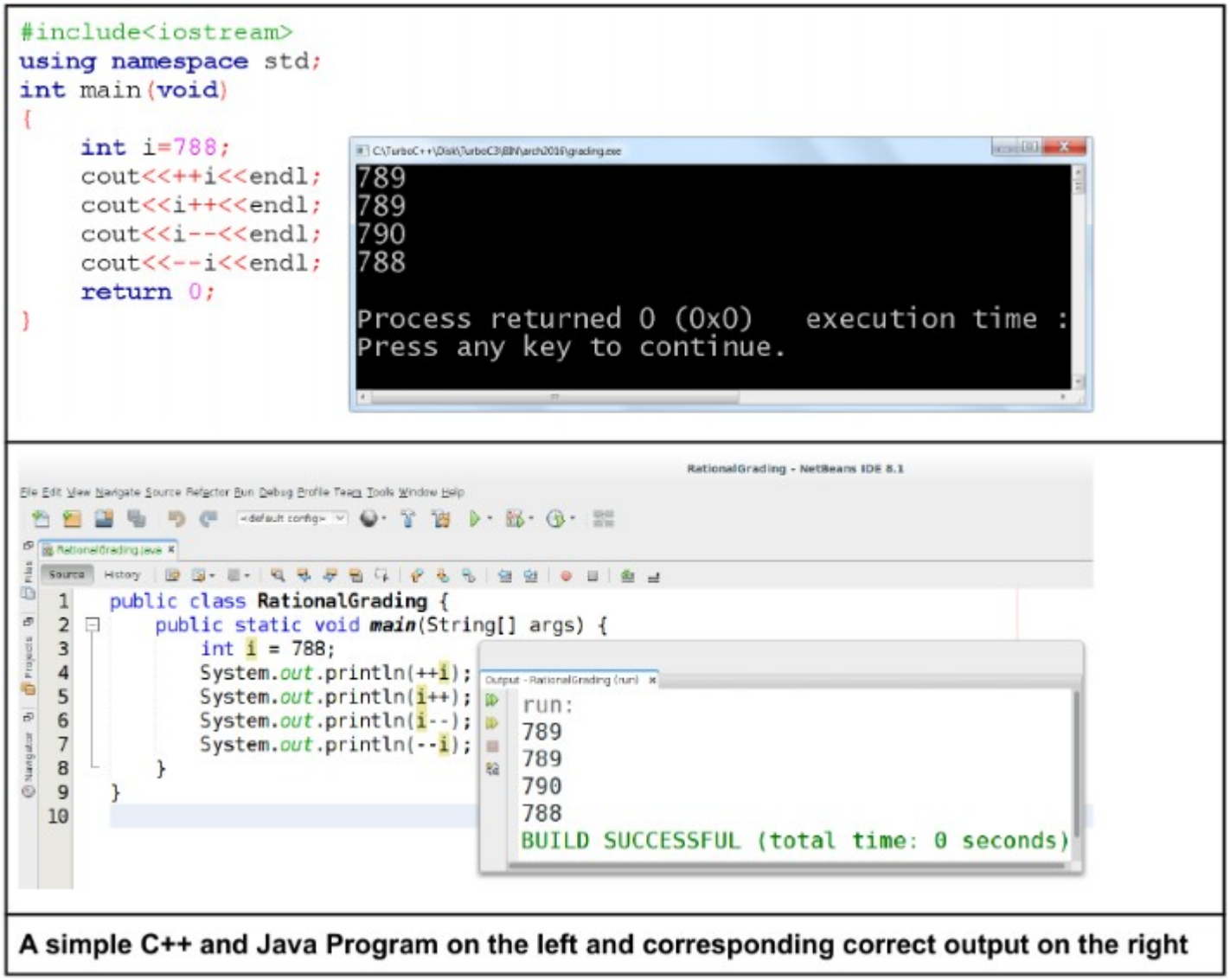
CSEC-ASTU Competitive Programming Contest 2021

Problem N02: Rational Grading

Time limit: 1s

Grading exam scripts of a programming course is often a pain and to reduce the pain teachers often give full marks only when the examinee's output matches exactly with the correct one. Then the mark can be easily calculated as a percentage of matching. But in some scenario, this approach may not be right — especially in those cases where the output of each step depends on that of the previous one. For example in the figure below you can see a program and its correct output. The 2nd, 3rd and 4th output depends on the output of the previous lines. So if the output of the first line is wrong and the next three lines are correct the examinee should not get 3 marks because he has made mistakes in all four cases (Or made a mistake while copying from his friend).



A simple C++ and Java Program on the left and corresponding correct output on the right

So to grade such a program we adopt the following new policy — we should judge the correctness of each line based on the output of the immediate previous line. For example, if for the program above if someone's output is:

```
1000
1000
1001
999
```

He/she should get 3 marks. Because, the output of the first line is incorrect but based on that line's output, the output of next three lines are correct. Given a program's description and its output, you job is to calculate the mark that the examinee will get.

Input

There are at most 1001 test cases. The description of each test case is given below

Each test case starts with two integers ivalue ($0 < ivalue_{10} \leq 1000000_{10}$) and t ($0 < t \leq 30$). Here ivalue denotes the initial value of the variable i. This value can be in decimal (leftmost digit is not zero), octal (leftmost digit is zero) or hexadecimal (starts with '`0x`' and non-numeric digits will be in uppercase). t denotes the total number of printing statements in the program. Each of the next t lines contains an expression that the printing function will print followed by the output for the expression written by the examinee. The expressions can be anyone from the following instruction set, s = {`i++, ++i, i--, --i, i`}, the output value will be between 0 and 1000000 (inclusive) and of course in decimal.

Output

Input is terminated by a line containing two zeroes.

Illustration of 2nd Sample Input

Expression	Current Value of i before printing based on $ivalue_{10}$ or the value in the previous line	Printed Value (By student)	Correct Expected Value	Marks obtained	Value of i after being printed
`++i`	0766 = 502	*789*	*503*	*0*	789
`i++`	789	789	789	1	790
`i--`	790	790	790	1	789
`--i`	789	788	788	1	788
Total Marks obtained				3	

CSEC-ASTU Competitive Programming Contest 2021

Sample Input 1	Sample Output 1
766 4 ++i 767 i++ 767 i-- 768 --i 766 0766 4 ++i 789 i++ 789 i-- 790 --i 788 0x766 4 ++i 1895 i++ 1895 i-- 1896 --i 1894 0 0	4 3 4