

CS 280

Spring 2018

John Bowers, Professor
Mike Lam, Professor



Competitive Programming

Course Goals

- Produce well-rounded computer scientists who can:
 - Categorize well-known, classic problems
 - Solve them! (possibly in teams)
- Additional benefits
 - Increase your programming skills
 - Help you prepare for job interviews
- Our ulterior motive
 - Prepare teams for upcoming contests!
 - ACM ICPC Qualifier and Regional contests
 - CCSC Eastern?

Recommended Text

- *Competitive Programming 3*
 - Halim and Halim
 - Available online (\$27 softcover, \$36 hardcover)
 - Not strictly necessary for this course
 - **HIGHLY RECOMMENDED** if you want to be competitive in the ACM contests!






Schedule

- Weekly meeting
 - Discuss a problem category and solution strategies
 - "**Solved**" problem: we will discuss the problem and solution
 - "**In-class**" problem: solve this one in class (or by Friday)
 - Submit on Kattis
 - "**Contest**" problem: solve this one for credit by Friday
 - Submit on Kattis and a (cleaned-up version) on Canvas
 - "**Stretch**" problem(s): solve these for experience and fun (and possibly extra credit)

Code Reviews

- Evaluate correctness, elegance, readability, formatting, and documentation
- Compare to your solutions
 - Discuss ramifications for efficiency and readability
- Graded according to the following rubric:
 - 3: Thorough and insightful
 - 2: Acceptable
 - 1: Deficient
 - 0: No submission

Grading

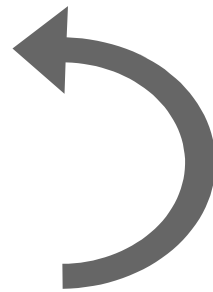
Contest   					
Criteria	Ratings				Pts
Attendance	2.0 pts Present		0.0 pts Absent		2.0 pts
In-class problem (Kattis)	1.0 pts Accepted		0.0 pts Not Accepted		1.0 pts
Contest problem (Kattis)	2.0 pts Accepted		0.0 pts Not Accepted		2.0 pts
Contest problem (Canvas)	2.0 pts Submitted		0.0 pts No submission		2.0 pts
Code review (Canvas)	3.0 pts Insightful and thorough	2.0 pts Acceptable	1.0 pts Deficient	0.0 pts No submission	3.0 pts
					Total Points: 10.0

Grading

- You can get a C simply by attending class, completing the in-class problem, attempting the contest problem (and submitting it on Canvas), and submitting an "acceptable" code review.
- You can get a B by doing all of the above and occasionally solving a contest problem and/or writing an "insightful and thorough" code review.
- To get an A, you'll need to solve most of the contest problems.

Solving Problems

- (Almost) all contest problems consist of
 - Background
 - Problem description
 - Sample input(s)
 - Sample output(s)
- Your task is to
 - Understand the problem
 - Categorize the problem
 - Design a solution
 - Code the solution
 - Debug the solution



Solving Problems

- Understanding is crucial!
 - DO NOT write any code until you understand the problem!

Solving Problems

Solving for Carrots

Carrots are good for you! First of all, they give you good night vision. Instead of having your lights on at home, you could eat carrots and save energy! Ethnomedically, it has also been shown that the roots of carrots can be used to treat digestive problems. In this contest, you also earn a carrot for each difficult problem you solve, or huffle-puff problems as we prefer to call them.



Photo by niznoz

You will be given the number of contestants in a hypothetical contest, the number of huffle-puff problems that people solved in the contest and a description of each contestant. Now, find the number of carrots that will be handed out during the contest.

Input

Input starts with two integers $1 \leq N, P \leq 1000$ on a single line, denoting the number of contestants in the contest and the number of huffle-puff problems solved in total. Then follow N lines, each consisting of a single non-empty line in which the contestant describes him or herself. You may assume that the contestants are good at describing themselves, in a way such that an arbitrary 5-year-old with hearing problems could understand it.

Output

Output should consist of a single integer: the number of carrots that will be handed out during the contest.

Solving Problems

Solving for Carrots

Carrots are good for you! First of all, they give you good night vision. Instead of having your lights on at home, you could eat carrots and save energy! Ethnomedically, it has also been shown that the roots of carrots can be used to treat digestive problems. In this contest, you also earn a carrot for each difficult problem you solve, or huffle-puff problems as we prefer to call them.



Photo by niznoz

You will be given the number of contestants in a hypothetical contest, the number of huffle-puff problems that people solved in the contest and a description of each contestant. Now, find the number of carrots that will be handed out during the contest.

Input

Input starts with two integers $1 \leq N, P \leq 1000$ on a single line, denoting the number of contestants in the contest and the number of huffle-puff problems solved in total. Then follow N lines, each consisting of a single non-empty line in which the contestant describes him or herself. You may assume that the contestants are good at describing themselves, in a way such that an arbitrary 5-year-old with hearing problems could understand it.

Output

Output should consist of a single integer: the number of carrots that will be handed out during the contest.

Sample Input 1

```
2 1
carrots?
bunnies
```

Sample Output 1

```
1
```

Sample Input 2

```
1 5
sovl problmz
```

Sample Output 2

```
5
```

Solving Problems

```
1  import java.io.*;
2  import java.util.*;
3
4  class Main
5  {
6      public static void main(String args[])
7      {
8          Scanner in = new Scanner(System.in);
9
10         int n = in.nextInt();    // # of contestants
11         int p = in.nextInt();    // # of huffle-puff problems
12
13         // number of carrots == number of huffle-puff problems
14         System.out.println(p);
15     }
16 }
```

Week 1

- Set up Kattis accounts
- Download Kattis template files
- Solve and submit "hello"
- Before Friday: solve and submit "fizzbuzz"
 - Submit code on Canvas "Week 1"
- If you can: solve and submit "oddgnome"