COS 126	General Computer Science	Spring 2017

Programming Exam 1

Instructions. This exam has one question. You have 50 minutes. The exam is *open course materials*, which includes the course textbook, the companion booksite, the course website, your course notes, and code you wrote for the course. Accessing other information or communicating with a non-staff member (such as via email, text, Facebook, Piazza, phone, or Snapchat) is prohibited.

Submission. Submit your solution via the link on the *Class Meetings* page. Click the *Check All Submitted Files* button to verify your submission. You may submit multiple times.

Grading. Your program will be graded for correctness, clarity (including comments), design, and efficiency. You will receive partial credit for a program that correctly implements some of the required functionality. You will receive a substantial penalty if your program does not compile or if you do not follow the prescribed input/output specifications.

Discussing this exam. Discussing or communicating the contents of this exam before solutions have been posted is a violation of the Honor Code.

This exam. You must turn in this exam. Print your name, NetID, precept, and the room in which you are taking the exam in the space below. Also, write and sign the Honor Code pledge. You may fill in this information now.

Name:

NetID:

Exam Room:

Precept:

"I pledge my honor that I will not violate the Honor Code during this examination."

Signature

Problem. Write a program `ShannonEntropy.java` that prints the *Shannon entropy* of a sequence of n integers between 1 and m (inclusive).

Suppose that the sequence of n integers contains x_i occurrences of the integer i. Let $p_i = x_i/n$ be the proportion of times that the integer i appears. Then, the *Shannon entropy* of the sequence is given by the formula:

$$H \quad = \quad -(p_1 \log_2 p_1 \; + \; p_2 \log_2 p_2 \; + \; \ldots \; + \; p_m \log_2 p_m)$$

Treat the term $p_i \log_2 p_i$ as 0 whenever $x_i = 0$. Note that $\log_2 z$ denotes the binary logarithm of z.

Recall that `Math.log(z)` returns $\log_e z$ and that $\log_2 z = \log_e z / \log_e 2$.

Step-by-step calculation (for reference). Here is an example with $m = 6$ and $n = 16$.

3 2 6 2 4 3 2 1 2 2 1 3 2 3 2 2

The following table shows the frequencies x_i, the proportions p_i, the $-p_i \log_2 p_i$ terms, and the Shannon entropy:

i	x_i	p_i	$-p_i \log_2 p_i$
1	2	1/8	3/8
2	8	1/2	1/2
3	4	1/4	1/2
4	1	1/16	1/4
5	0	0	0
6	1	1/16	1/4
	16	1	15/8

$$\textit{Shannon entropy} \;=\; 3/8 + 1/2 + 1/2 + 1/4 + 0 + 1/4 = 15/8 = 1.875$$

Input specification. The input from standard input consists of two positive integers m and n, followed by a sequence of n integers between 1 and m (inclusive), separated by whitespace.

```
% more dice.txt
6   ← m
16  ← n
3 2 6 2 4 3 2 1 2 2
1 3 2 3 2 2
```

16 integers between 1 and 6

```
% more letters.txt
26
10000
19 21  8 15  1  7  4  4 22 15
 5 15  6  1 12 15 22 16  3 20
 8 14 20 20 12 20 21 20 18  9
20 19  5 20 20  5  4 15 12 14
14  9 15  6  1 15  7  1 14 15
 7  1 23  3 18 ...
```

The data files `dice.txt` and `letters.txt` are available via the *Class Meetings* page.

As usual, your program must handle all inputs that conform to the input specification (but we will not test non-conforming inputs).

Output specification. The output to standard output consists of one floating-point number—the Shannon entropy of the sequence of integers.

```
% java-introcs ShannonEntropy < dice.txt
1.875

% java-introcs ShannonEntropy < letters.txt
4.169566323759212
```

Do *not* print any other output to standard output.

Submission. Submit `ShannonEntropy.java` via the link on the *Class Meetings* page.