X

Week 6 Programming Assignment: Book Cartons

Due on 2020-03-16, 23:59 IST

- Select your language (C/C++/Java/Python2/Python3)
- Paste your code into the submission window.
- There are some public test cases and some (hidden) private test cases.
- "Compile and run" will evaluate your submission against the public test cases.
- "Submit" will evaluate your submission against the hidden private test cases. There are 10 private testcases in all, each with equal weightage. You will get feedback about which private test cases pass or fail, though you cannot see the actual test cases.
- For each private testcase, you will get a status 'Evaluated', 'Not Evaluated' or 'Time Limit Exceeded'.
  - 'Evaluated' does not mean your answer is correct, just that the entire testcase completed and reported some answer.
  - 'Time Limit Exceeded' means your code took too long.
  - 'Not Evaluated' means this testcase was not run. This typically happens to all testcases after the first one that times out.
- Ignore warnings about "Presentation errors".

Book cartons

A university department is shifting its department library to a more spacious room on a newly constructed floor. The books from the library have been packed into m cartons, numbered $1, 2, \ldots, m$, containing $b_1, b_2, \ldots, b_m$ books, respectively, and transported to the new library room.

There are k student volunteers available to unpack the m cartons, where $k \leq m$. Each carton must be assigned to a single volunteer, and every volunteer must

get a non-empty continuous sequence of cartons to unpack.

More formally, we need to find numbers $0 = c_0 < c_1 < c_2 < \ldots < c_k = m$ such that volunteer j, $1 \le j \le k$ unpacks cartons $c_{j-1}+1$ to c_j.

The time each volunteer takes to unpack a carton is directly proportional to the number of books in the carton. The goal is parallelize the unpacking to finish in the fastest possible time. For this, we need to assign cartons such that the maximum number of books assigned to any one volunteer is minimized.

Solution hint

Given a target T, use a greedy strategy to check if there is a legal allocation where no volunteer is assigned more than T books. Find the optimum T using binary search. Note that if a greedy strategy finds an allocation achieving target T using $k' < k$ volunteers, this allocation can always be subdivided to achieve the same target with exactly k volunteers.

Input format

Each test case consists of exactly two lines. The first line has are two integers m and k. The second line has m integers $b_1, b_2, \ldots, b_m$ separated by spaces.

Output format

Your output should be a single line with the input sequence $b_1, b_2, \ldots, b_m$ divided into exactly k parts such that the maximum sum in a single part is as small as possible. Use the slash character (' / ') to separate the parts. There must be exactly one space character between any two successive numbers and between the number and the slash. If there is more than one solution, print the one that minimizes the work assigned to the first volunteer, then to the second volunteer etc. Each volunteer must be assigned at least one carton.

Test Data:

You may assume that $1 \le k \le m \le 500$, always. Also, each carton contains a positive number of books, which is less than 10,000,000.

Sample Input 1:

```
9 3
100 200 300 400 500 600 700 800 900
```

Sample Output 1:

```
100 200 300 400 500 / 600 700 / 800 900
```

Sample Input 2:

Week 8: Linear Programming and Network Flows

Week 8: Intractability

Week 8 Quiz

Text Transcripts

Books

Download Videos

```
5 4
100 100 100 100 100
```

Sample Output 2:

```
100 / 100 / 100 / 100 100
```

Sample Test Cases

	Input	Output
Test Case 1	17 4 100 200 300 400 500 600 700 800 900 800 700 600 500 400 300 200 100	100 200 300 400 500 / 600 700 800 / 900 800 700 / 600 500 40 0 300 200 100
Test Case 2	100 99 100	100 / 100 100
Test Case 3	500 499 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100	100 / 100 /

100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /
100 100 100 100 100	100 / 100 / 100 / 100 / 100 /

	100 100	100 / 100 100
Test Case 4	4 2 1 2 4 8	1 2 4 / 8
Test Case 5	6 2 1 2 4 8 4 2	1 2 4 / 8 4 2
Test Case 6	6 2 1 2 3 3 2 1	1 2 3 / 3 2 1
Test Case 7	2 2 10000000 10000000	10000000 / 10000000

Test Case 8	6 3 1000000 1 1 1 1 1	1000000 / 1 / 1 1 1 1
Test Case 9	6 3 1 1 1 1 1 10000000	1 / 1 1 1 1 / 10000000
Test Case 10	6 6 6 5 4 3 2 1	6 / 5 / 4 / 3 / 2 / 1
Test Case 11	9 3 100 200 300 400 500 600 700 800 900	100 200 300 400 500 / 600 700 / 800 900
Test Case 12	5 4 100 100 100 100 100	100 / 100 / 100 / 100 100

The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.