Silver Jubilee

All of you know that Department of CSE at BUET is going to celebrate its Silver Jubilee A.K.A 25 years anniversary. Being a prospective student of the department you want to work in the decoration team. The team takes walls and draws graffitis on it. Of course the graffitis are based on several aspects of cse department. However, they are not equal in size. Though all the graffiti on a wall have equal height, but they have in general varying lengths. Several students are interested in this painting work and we want to give all of them opportunity to draw graffitis. However there are some rules. One graffiti can be assigned to a single student only, and every student must get a continuous sequence of graffitis. All the graffiti aren't of equal length or of equal complexity. Some are small, while some are large. Some are simple to draw, while some are complex. Time needed for a student to complete a graffiti is proportional to its length and complexity. You have a wall and a sequence of graffitis. You have to assign every student a consecutive sequence of graffitis. So they will start at the same time at different locations on the wall. The time needed to complete the job is determined by the student who was assigned the most work. Therefore, our goal is to minimize the maximum workload assigned to a single student. Your task is to find that workload, and also the optimal assignment.

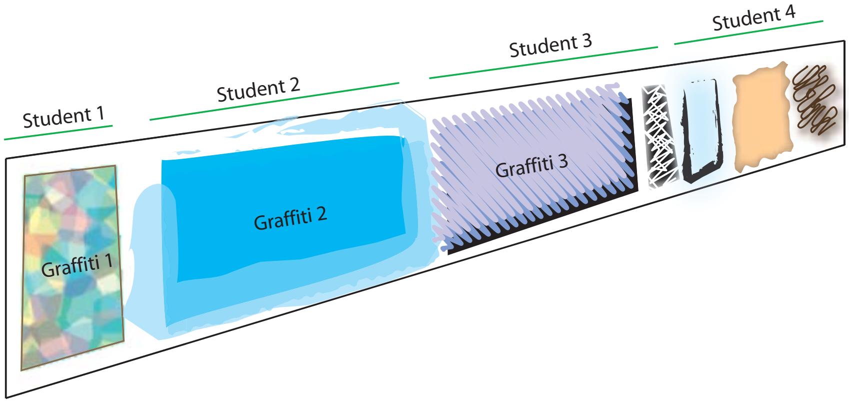

Figure 1: An example graph

Imagine you have m graffitis (numbered $1, 2, \ldots, m$) that may take different time to complete ($p_1, p_2, \ldots, p_m$). Your task is to divide these graffiti job among k students, $k \leq m$. Remember, each graffiti can be assigned to a single student only, and every student must get a continuous sequence of graffiti jobs.

Input Specification

The input consists of multiple test cases. Each case consists of exactly two lines. At the first line, there are two integers m and k, $1 \leq k \leq m \leq 500$. At the second line, there are integers $p_1, p_2, \ldots p_m$ separated by spaces. All these values are positive and less than 10000000. There are blank lines between consecutive cases.

You should terminate your program when `0 0` is input for both m and k.

Output Specification

For each case, print exactly two lines. First line contains one integer which denotes the minimum value of the maximum workload. Second line must contain the input succession $p_1, p_2, \ldots p_m$ divided into exactly k parts such that the maximum sum of a single part should be 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 student, then to the second student etc. But each student must be assigned at least one graffiti.**

There should be one blank line between distinct cases.

Sample Input

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

5 4
100 100 100 100 100

12 5
48 65 53 3 90 3 90 30 52 75 54 22

25 8
46 18 71 6 35 59 35 52 83 62 85 71 72 88 46 100 8 5 89 85 61 80 59 43 39

12 3
10 10 10 10 10 10 10 10 10 10 10 10

10 3
2 2 2 3 3 3 4 4 4 5

20 5
1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2

0 0
```

Sample Output

```
1700
100 200 300 400 500 / 600 700 / 800 900

200
100 / 100 / 100 / 100 100

146
48 65 / 53 3 90 / 3 90 30 / 52 75 / 54 22

221
```

```
46 / 18 71 6 35 59 / 35 52 83 / 62 85 71 / 72 88 46 / 100 8 5 89 / 85 61 / 80 59 43 39

40
10 10 10 10 / 10 10 10 10 / 10 10 10 10

12
2 2 2 3 3 / 3 4 4 / 4 5

21
1 2 3 4 5 / 6 7 8 / 9 1 2 3 4 / 5 6 7 / 8 9 1 2
```

Remarks

Algorithm You should use `Binary Search` over the solution space to complete this assignment. At first find the minimum of the maximum workload using binary search. Then carefully make the assignment, so that the output conditions are satisfied.

Input Output Explanation In the first case the optimal assignment among three students are `{100, 200, 300, 400, 500}, {600,700}, (800,900}`. First student has a workload of 1500 pages, second 1300 pages, and the third student 1700 having max. So the answer is 1700 here. For the third case, the work is divided as `{48, 65} {53, 3, 90} {3, 90, 30} {52,75} {54, 22}` among 5 students. As you can see, the maximum of these assignments are `146 (53+3+90)`. No other assignment can make the maximum workload less than 146. So 146 is the optimal answer.

Extra Credit For the minimum value of maximum workload, there can be several assignments. For example, see the second test case in sample input.

`100 / 100 / 100 / 100 100`, or
`100 100 / 100 / 100 / 100`, or
`100 / 100 100 / 100 / 100`, or
`100 / 100 / 100 100 / 100`,
all assignments result in a maximum workload of `200`. However, first one is the correct answer for the mandatory portion as we said, "If there is more than one solution, print the one that minimizes the work assigned to the first student, then to the second student etc". To get some bonus credit, you can print all of the possible assignments that results in the minimum highest workload. Code the bonus portion in a separate program other than your mandatory portion, so that we can judge that separately.

Md. Tanvir Al Amin
tanviralamin@gmail.com