VK Cup 2012 Qualification Round 1

A. Next Round

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

"Contestant who earns a score equal to or greater than the k-th place finisher's score will advance to the next round, as long as the contestant earns a positive score..." — an excerpt from contest rules.

A total of n participants took part in the contest ($n \geq k$), and you already know their scores. Calculate how many participants will advance to the next round.

Input

The first line of the input contains two integers n and k ($1 \leq k \leq n \leq 50$) separated by a single space.

The second line contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 100$), where a_i is the score earned by the participant who got the i-th place. The given sequence is non-increasing (that is, for all i from 1 to $n - 1$ the following condition is fulfilled: $a_i \geq a_{i+1}$).

Output

Output the number of participants who advance to the next round.

Examples

input
8 5 10 9 8 7 7 7 5 5
output
6

input
4 2 0 0 0 0
output
0

Note

In the first example the participant on the 5th place earned 7 points. As the participant on the 6th place also earned 7 points, there are 6 advancers.

In the second example nobody got a positive score.

B. Taxi

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After the lessons n groups of schoolchildren went outside and decided to visit Polycarpus to celebrate his birthday. We know that the i-th group consists of s_i friends ($1 \le s_i \le 4$), and they want to go to Polycarpus together. They decided to get there by taxi. Each car can carry at most four passengers. What minimum number of cars will the children need if all members of each group should ride in the same taxi (but one taxi can take more than one group)?

Input

The first line contains integer n ($1 \le n \le 10^5$) — the number of groups of schoolchildren. The second line contains a sequence of integers $s_1, s_2, \ldots, s_n$ ($1 \le s_i \le 4$). The integers are separated by a space, s_i is the number of children in the i-th group.

Output

Print the single number — the minimum number of taxis necessary to drive all children to Polycarpus.

Examples

input
5 1 2 4 3 3
output
4

input
8 2 3 4 4 2 1 3 1
output
5

Note

In the first test we can sort the children into four cars like this:

- the third group (consisting of four children),
- the fourth group (consisting of three children),
- the fifth group (consisting of three children),
- the first and the second group (consisting of one and two children, correspondingly).

There are other ways to sort the groups into four cars.

C. Cd and pwd commands

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya is writing an operating system shell, and it should have commands for working with directories. To begin with, he decided to go with just two commands: cd (change the current directory) and pwd (display the current directory).

Directories in Vasya's operating system form a traditional hierarchical tree structure. There is a single root directory, denoted by the slash character "/". Every other directory has a name — a non-empty string consisting of lowercase Latin letters. Each directory (except for the root) has a parent directory — the one that contains the given directory. It is denoted as "..".

The command cd takes a single parameter, which is a path in the file system. The command changes the current directory to the directory specified by the path. The path consists of the names of directories separated by slashes. The name of the directory can be "..", which means a step up to the parent directory. «..» can be used in any place of the path, maybe several times. If the path begins with a slash, it is considered to be an absolute path, that is, the directory changes to the specified one, starting from the root. If the parameter begins with a directory name (or ".."), it is considered to be a relative path, that is, the directory changes to the specified directory, starting from the current one.

The command pwd should display the absolute path to the current directory. This path must not contain "..".

Initially, the current directory is the root. All directories mentioned explicitly or passed indirectly within any command cd are considered to exist. It is guaranteed that there is no attempt of transition to the parent directory of the root directory.

Input

The first line of the input data contains the single integer n ($1 \le n \le 50$) — the number of commands.

Then follow n lines, each contains one command. Each of these lines contains either command pwd, or command cd, followed by a space-separated non-empty parameter.

The command parameter cd only contains lower case Latin letters, slashes and dots, two slashes cannot go consecutively, dots occur only as the name of a parent pseudo-directory. The command parameter cd does not end with a slash, except when it is the only symbol that points to the root directory. The command parameter has a length from 1 to 200 characters, inclusive.

Directories in the file system can have the same names.

Output

For each command pwd you should print the full absolute path of the given directory, ending with a slash. It should start with a slash and contain the list of slash-separated directories in the order of being nested from the root to the current folder. It should contain no dots.

Examples

input
7 pwd cd /home/vasya pwd cd .. pwd cd vasya/../petya pwd
output
/ /home/vasya/ /home/ /home/petya/

input
4 cd /a/b pwd cd ../a/b pwd
output
/a/b/ /a/a/b/

D. Ice Sculptures

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Berland University is preparing to celebrate the 256-th anniversary of its founding! A specially appointed Vice Rector for the celebration prepares to decorate the campus. In the center of the campus n ice sculptures were erected. The sculptures are arranged in a circle at equal distances from each other, so they form a regular n-gon. They are numbered in clockwise order with numbers from 1 to n.

The site of the University has already conducted a voting that estimated each sculpture's characteristic of t_i — the degree of the sculpture's attractiveness. The values of t_i can be positive, negative or zero.

When the university rector came to evaluate the work, he said that this might be not the perfect arrangement. He suggested to melt some of the sculptures so that:

- the remaining sculptures form a regular polygon (the number of vertices should be between 3 and n),
- the sum of the t_i values of the remaining sculptures is maximized.

Help the Vice Rector to analyze the criticism — find the maximum value of t_i sum which can be obtained in this way. It is allowed not to melt any sculptures at all. The sculptures can not be moved.

Input

The first input line contains an integer n ($3 \le n \le 20000$) — the initial number of sculptures. The second line contains a sequence of integers $t_1, t_2, \ldots, t_n$, t_i — the degree of the i-th sculpture's attractiveness ($-1000 \le t_i \le 1000$). The numbers on the line are separated by spaces.

Output

Print the required maximum sum of the sculptures' attractiveness.

Examples

input
8 1 2 -3 4 -5 5 2 3
output
14

input
6 1 -2 3 -4 5 -6
output
9

input
6 1 2 3 4 5 6
output
21

Note

In the first sample it is best to leave every second sculpture, that is, leave sculptures with attractivenesses: 2, 4, 5 и 3.

E. Phone Talks

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Cool J has recently become a businessman Mr. Jackson, and he has to make a lot of phone calls now. Today he has n calls planned. For each call we know the moment t_i (in seconds since the start of the day) when it is scheduled to start and its duration d_i (in seconds). All t_i are different. Mr. Jackson is a very important person, so he never dials anybody himself, all calls will be incoming.

Mr. Jackson isn't Caesar and he can't do several things at once. If somebody calls him while he hasn't finished the previous conversation, Mr. Jackson puts the new call on hold in the queue. In this case immediately after the end of the current call Mr. Jackson takes the earliest incoming call from the queue and starts the conversation. If Mr. Jackson started the call at the second t, and the call continues for d seconds, then Mr. Jackson is busy at seconds $t, t + 1, ..., t + d - 1$, and he can start a new call at second $t + d$. Note that if Mr. Jackson is not busy talking when somebody calls, he can't put this call on hold.

Mr. Jackson isn't Napoleon either, he likes to sleep. So sometimes he allows himself the luxury of ignoring a call, as if it never was scheduled. He can ignore at most k calls. Note that a call which comes while he is busy talking can be ignored as well.

What is the maximum number of seconds Mr. Jackson can sleep today, assuming that he can choose an arbitrary continuous time segment from the current day (that is, with seconds from the 1-st to the 86400-th, inclusive) when he is not busy talking?

Note that some calls can be continued or postponed to the next day or even later. However, the interval for sleep should be completely within the current day.

Input

The first input line contains a pair of integers n, k ($0 \le k \le n \le 4000$) separated by a space. Following n lines contain the description of calls for today. The description of each call is located on the single line and consists of two space-separated integers t_i and d_i, ($1 \le t_i, d_i \le 86400$). All t_i are distinct, the calls are given in the order of strict increasing t_i.

Scheduled times of calls [t_i, $t_i + d_i - 1$] can arbitrarily intersect.

Output

Print a number from 0 to 86400, inclusive — the maximally possible number of seconds for Mr. Jackson to sleep today.

Examples

input
3 2 30000 15000 40000 15000 50000 15000
output
49999

input
5 1 1 20000 10000 10000 20000 20000 25000 10000 80000 60000
output
39999

Note

In the first sample the most convenient way is to ignore the first two calls.

In the second sample it is best to ignore the third call. In this case Mr. Jackson will have been speaking:

- first call: from 1-st to 20000-th second,
- second call: from 20001-st to 30000-th second,
- fourth call: from 30001-st to 40000-th second (the third call is ignored),
- fifth call: from 80000-th to 139999-th second.

Thus, the longest period of free time is from the 40001-th to the 79999-th second.

Codeforces (c) Copyright 2010-2016 Mike Mirzayanov

The only programming contests Web 2.0 platform