
problems

problem	pizza	zgrade	reklame
source file	pizza.pas pizza.c pizza.cpp	zgrade.pas zgrade.c zgrade.cpp	reklame.pas reklame.c reklame.cpp
input data	stdin		
output data	stdout		
memory limit (heap)	16 MB	16 MB	128 MB
memory limit (stack)	2 MB		
time limit (pentium4 1.6 ghz)	1 second	0.4 seconds	1 second
points	40	50	60
	150		

pizza

Mirko is having a birthday party so he is inviting his friends to eat some pizza.

Unfortunately, Mirko's friends are very young and cannot eat an entire pizza, but all of them know **exactly** how much pizza they can eat and insist on getting **the exact amount of pizza in one slice**.

Their requests break down to three different pizza slices - either **one quarter** or **a half** or **three quarters** of pizza.

Write a program that will help Mirko find out what is the **minimal** number of pizzas he has to order so that everyone gets exact amount of pizza they want.

input data

First line contains an integer N , $1 \leq N \leq 10,000$, number of friends.

In each of next N lines there is amount of pizza that each of Mirko's friends wants to eat, that is the fraction $1/4$, $1/2$ or $3/4$.

output data

In the first and only line you should write the minimal number of pizzas Mirko has to order.

examples

input

3

1/2

3/4

3/4

output

3

input

5

1/2

1/4

3/4

1/4

1/2

output

3

input

6

3/4

1/2

3/4

1/2

1/4

1/2

output

4

zgrade

Lets observe sequences made only of **round** and **square** brackets, i.e. characters '()' []'.

A sequence of brackets is **regular** if it satisfies this inductive definition:

- 1.) '()' and '[]' are regular sequences
- 2.) If **A** is regular, then **(A)** and **[A]** are regular sequences
- 3.) If **A** and **B** are regular, then **AB** is regular sequence

For example '() () []', '([]) [()]' and '[(())] []' are regular, while '(,] [, '[' (]' and '([)]' are not regular.

The sequence of brackets is given.

In every step, one bracket is inserted **at the beginning** or **at the end** of the sequence (round or square, left or right).

Write a program that will, **after each step**, determine **the length of the shortest regular subsequence of consecutive characters** that contains the bracket added **in that step**.

input data

First line contains initial sequence of brackets, whose length is at most 100,000 characters.

Next line contains integer N , $1 \leq N \leq 100,000$, a number of steps.

In each of next N lines there are integer A and character C , separated by a single space. If A is **zero (0)**, than character C is inserted at the **beginning**, and if A is **one (1)** then C is inserted at the **end**.

output data

In each of N lines, you should write the length of subsequence after that step. If there is no such subsequence, write number **0**.

examples

input	input	input
((]	[])
1	3	3
1)	1)	0)
output	0)	0 (
2	0 (0 (
	output	output
	0	0
	0	2
	2	6

reklame

Slavko can display **at most K banners** each day on his web page.

Advertisers do not insist on Slavko telling them the exact time their banner appears the first time, but they request that after the first display the banner **starts appearing at some specific days**. For example, they can insist that after the banner is displayed first time on Slavko's web page, (that day is referenced as 1st day) it is displayed also on 3rd and 4th day. One request is applied to **at most 7 days** starting from the time of the first display to the last display of a banner.

Slavko receives requests for banners in a certain order, and if he gets request A **before** request B, he **cannot start** displaying banner B before he starts displaying banner A.

Furthermore, Slavko wants to minimize **the number of days** from the day he begins displaying first banner until the day the last banner is displayed on his web page.

Write a program that will find that minimal number of days.

input data

First line contains integers N and K, $1 \leq N \leq 100$, $1 \leq K \leq 4$, number of requests and maximum number of banners on Slavko's web page.

Each of next N lines contains a description of one request in the same order Slavko received them. First number in each of those lines is number R_i , a total number of times that banner has to be displayed. Then R_i-1 numbers follow in the same line. They represent additional days when banner must be displayed. That numbers will be **sorted in ascending order**.

output data

In the first and only line, you should write minimal number of days as described in text.

examples

input

1 1
3 3 6

output

6

input

2 1
2 3
2 3

output

4

input

3 2
3 3 6
3 3 7
3 5 6

output

7