TASK	NASLJEDSTVO	GMO	SUMO	GUMA	ČOKOLADE	UTRKA

source code	`nasljedstvo.pas`
	`nasljedstvo.c`
	`nasljedstvo.cpp`
		`gmo.pas`
		`gmo.c`
		`gmo.cpp`
			`sumo.pas`
			`sumo.c`
			`sumo.cpp`
				`guma.pas`
				`guma.c`
				`guma.cpp`
					`cokolade.pas`
					`cokolade.c`
					`cokolade.cpp`
						`utrka.pas`
						`utrka.c`
						`utrka.cpp`

input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	1 second	1 second	1 second	1 second	2 seconds	2 seconds
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
point value	**50**	**80**	**100**	**120**	**140**	**160**
	650					

Problems translated from Croatian by: **Paula Gombar**

A wealthy estate owner is so old that she's at that point in her life when she can't help talking funny. That is, naturally, the reason why her loving **N** daughters have started discussing their mother's heritage.

The youngest is sick and tired of just talking, so she conveniently decided to grab a hold of her share of the heritage. She knew exactly where her mother keeps her golden medallions – inside a thick sock in the third drawer next to the mirror in the hallway! The cunning daughter found this pile of medallions, split it into **N** equal parts, claimed her part and put the rest back into the sock. If the medallions couldn't have been split into **N** identical parts, then the parts were nearly identical: each differed from another by one medallion at most. In that case, the daughter claimed one of the smaller parts for herself.

The rest of the daughters found out about this (mis)deed so they counted the remaining medallions and now they want to know the initial number of medallions inside the sock, before the youngest one took her share. It is your task to answer this question. Given that there could be more than one possible answer, output both the smallest and the largest of them.

INPUT

The first line of input contains the integer **N** (2 ≤ **N** ≤ 15), the number of daughters.

The second line of input contains the integer **O** (**N** ≤ **O** ≤ 100), the number of remaining medallions.

OUTPUT

The first and only line of output must contain two integers: the minimal and the maximal possible total number of medallions.

SAMPLE TESTS

input 2 5 **output** 9 10	**input** 3 5 **output** 7 7

Clarification of the first example: There are two daughters, so that means the youngest daughter took half of the medallions. If there were initially 9 medallions, the youngest took 4, so 5 are remaining. If, by any chance, there were initially 10 medallions, she took 5 so there are also 5 remaining. Hence, the possible answers are both 9 and 10.

A multinational company is asking you to help them genetically modify an apple. In order for the apples to grow faster, to get more of them, to make them bigger and make them look nicer and more simmetrical, the apple's DNA requires an insertion of a certain swine gene.

The apple's DNA is represented by a series of characters from the set {A, C, G, T}. The required swine gene is also comprised of charaters from this set. The apple's DNA should be injected with **some characters into some places**, so that the resulting sequence **contains a swine gene** somewhere (in successive locations). To make things a bit more complicated, inserting each of the characters A, C, G, T has its own **cost**.

Help this multinational company in achieving their goal with **the lowest possible total cost**. As a reward, you get a ton of their apples.

INPUT

The first line of input contains a sequence of **N** (1 ≤ **N** ≤ 10 000) characters which represent the apple's DNA.

The second line of input contains a sequence of **M** (1 ≤ **M** ≤ 5 000) characters which represent the swine gene that we want to insert into the apple's DNA.

Both the sequences are comprised only of characters from the set {A, C, G, T}.

The third line of input contains four integers from the interval [0, 1000]: the cost of inserting one character A, C, G, T, in that order.

OUTPUT

The first and only line of output must contains the minimal total cost.

SCORING

In test cases worth 80% of total points, **N** and **M** will not exceed 2000.

SAMPLE TESTS

input GTA CAT 5 7 1 3 **output** 10	**input** TATA CACA 3 0 3 0 **output** 3	**input** TCGCGAG TGCAG 10 10 15 15 **output** 25

Clarification of the first example: Some of the possible solutions are G**CA**TA and GT**C**A**T** (the inserted characters are bolded), the first solution costs 7 + 5, the second 7 + 3.

In a Japanese monastery, otherwise known for serious fasting and ascetic life, the Head of the sumo wrestling section has decided to organise training-competitions for his **N** fighters. He determined the exact sequence of **M** fights and its participants (two fighters face each other per fight).

Just moments before the competition, the Head realised he could easily stir things up a bit! He could divide his fighters into two teams so that only fighters of different teams face each other in each fight. Since the fighting schedule has already been made and it doesn't meet this condition, and we mustn't change it for whatever zen reason there is, the Head is left with only one option. That is to divide the fighters into two teams so that the fighters from the same team face each other in a fight as late as possible.

Help the Head! For a given fighting schedule, determine the ordinal number of the first fight where two fighters from the same team have to face each other, under the condition that we divide them in the best possible way, so that the required fight takes place as late as possible. In all test data, such fight will definitely occur.

INPUT

The first line of input contains the integer **N** (1 ≤ **N** ≤ 100 000), the number of fighters. The fighters are marked with numbers from 1 to **N**.

The second line of input contains the integer **M** (1 ≤ **M** ≤ 300 000), the number of fights.

Each of the following **M** lines contains fights in the order which they must take place. Each line contains two different integers from the interval [1, **N**]: the labels of fighters who are going to face each other.

OUTPUT

The first and only line of output must contain the ordinal number (from 1 to **M**) of the required fight.

SAMPLE TESTS

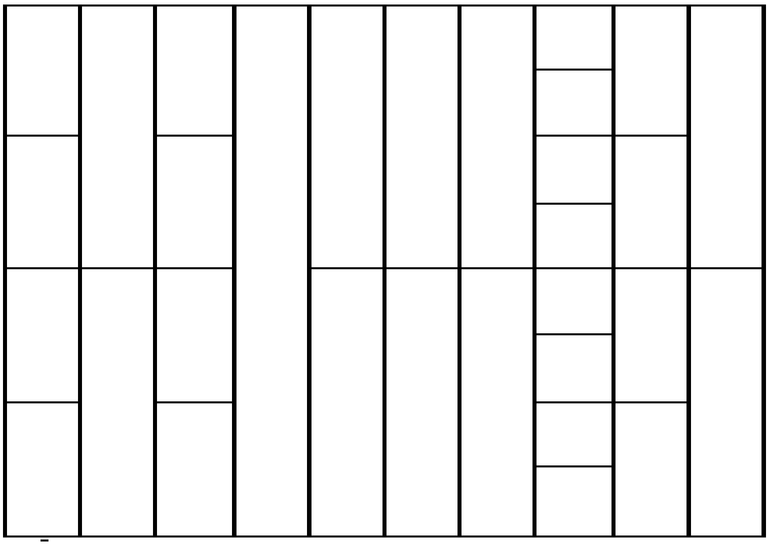
input	input

5
5
1 2
2 3
3 4
4 5
5 1
	6
	8
	1 2
	3 4
	5 6
	1 3
	1 6
	4 5
	2 4
	2 6

output	**output**
5	6

A factory called Gumi-Gumi is dedicated to making tires. Their carving machine is responsible for carving fillisters into the tire. The tire has **N** vertical fillisters which divide the rubber into **N**+1 vertical parts. Horizontal cuts are made on each vertical part so that all parts comprising the vertical part are of **equal size**. The machine can make fillisters on **one or more not necessarily continuous** vertical sections in one cut, but it can only cut in a **straight line**.

An example of a tire cutting strategy, corresponding to the third sample test.



The topmost and the lowest lines represent a full horizontal cut, whereas the first and the last vertical lines are the ends of the tire.

You are given the shape of the tire. Your task is to calculate the minimal possible number of cuts necessary in order to obtain such shape.

INPUT

The first line of input contains the integer **N** (1 ≤ **N** ≤ 100 000).

Each of the following **N**+1 lines contains an integer $\mathbf{a_i}$ (1 ≤ $\mathbf{a_i}$ ≤ 100 000), representing the number of parts which the ith vertical section should consist of.

OUTPUT

The first and only line of output must consist of the minimal number of cuts required.

SCORING

In test cases worth 20% of total points, **N** will not exceed 100.

SAMPLE TESTS

input	**input**	**input**
1 2 5	2 3 7 14	9 4 2 4 1 2 2 2 8 4 2
output	**output**	**output**
5	15	7

Mirko is a party animal, so he has decided to organise an **endless amount** of parties for his friends. To satisfy the party's needs, he has decided to set up **N** tables with **candy** on them. We know the number of candies $\mathbf{b_i}$ on each table. On **the first day** of the rest of eternity, Mirko is going to invite **one** friend per table, on **the second day** he will invite **two** friends per table, on **the third day three** friends... In general, obviously, on **the kth day** he is going to invite **k friends** per each table.

When his friends enter the room, **k** people will sit down at each table and they will divide the candies on their table in **k as large as possible equal pieces**, and **get rid of the possible remains**. After the candy division, because of jealousy and various other reasons, **only tables with the same amount of candy per capita will socialise together**. Mirko has all eternity to study the social dynamics of his parties. Firstly, he wants to know the answer to the following question: given an **s** between 1 and **N**, what is **the earliest day** when there is a group of **exactly s** tables socialising together?

As usual, Mirko is incapable of solving his own problems, so every few days he comes to you and asks you what the required number is, given an **s**. Alas, he has all eternity to ask questions, but you don't. Therefore, you are going to write a programme which outputs Mirko's required answers for each **s** from **1** to **N**.

Please note: Before each party, Mirko renews the candy supply on each table, meaning the supplies are equal to those before the first party. Additionally, all people leave from the current party before the next one starts.

INPUT

The first line of input contains the integer **N** ($1 \le \mathbf{N} \le 100$).

The second line of input contains **N** integers, the **ith** number marking the number of candy on the **ith** table.

The numbers are from the interval $[1, 10^8]$.

OUTPUT

Output **N** lines, each line containing a single integer.

The **sth** line should contain the required number for a group sized **s** or -1 if there will never be a group of that size.

SCORING

In test cases worth 30% of total points, the number of candy on all tables will not exceed 10^3.

In test cases worth additional 30% of total points, the number of candy on all tables will not exceed 10^6.

SAMPLE TESTS

input 5 11 10 9 6 4 **output** 1 2 3 6 12	**input** 3 5 5 5 **output** -1 -1 1	**input** 8 12 16 95 96 138 56 205 84 **output** 1 5 14 49 96 97 139 206

Clarification of the first example: On the first day, each table will socialise only with itself so the answer for groups sized 1 is 1. Already on the second day, people sitting at tables 1 and 2 are going to get 5 candies per capita and socialise together, so the answer for a group sized 2 is 2.

On the third day, tables 1, 2 and 3 will socialise (because they all have 3 candies per capita).

On the sixth day, tables 1, 2, 3 and 4 will socialise (because they now have 1 candy per capita).

Finally, on the twelfth day, all tables will socialise together because they will all get zero candy per capita.

Clarification of the second example: All tables have the same amount of candy per capita, so a group sized less than 3 will never exist.

Mirko and Slavko are the only two contestants at the Grand Prix of Dabrovina Donja which is driven through nearby villages. The villages are connected via **one-way** roads, and for each road **i** we know $\mathbf{M}_i$ and $\mathbf{S}_i$, the time necessary for Mirko and Slavko to cross that road. The race itself is circular (meaning it starts and begins in the same village), but the route itself hasn't been determined yet.

Mirko has bribed the organisers of the race so they'd pick a route in his favour. Specifically, the organisers are going to pick **the shortest route** (containing the minimal number of roads) such that Mirko is **strictly faster** than Slavko on that route. If, by any chance, there are several such routes, the organisers choose the one where Mirko gains **maximal** advantage.

INPUT

The first line of input contains two integers **N**, **M** ($2 \le \mathbf{N} \le 300$, $2 \le \mathbf{M} \le \mathbf{N}(\mathbf{N}-1)$), the number of villages and the number of connecting roads.

Each of the following **M** lines contains 4 integers $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{M}_i$, $\mathbf{S}_i$ ($1 \le \mathbf{A}_i, \mathbf{B}_i \le \mathbf{N}$, $\mathbf{A}_i \ne \mathbf{B}_i$, $0 \le \mathbf{S}_i, \mathbf{M}_i \le 10^6$). Respectively, the initial and ending village of the **i**th road, the time necessary for Mirko and the time necessary for Slavko to cross that road. There won't exist two different roads that connect the same pair of villages in the same direction.

OUTPUT

The first and only line of output must contain two integers: the shortest possible route (with the minimal number of roads) such that Mirko wins, and the maximal advantage Mirko can gain on a route of the shortest length.

Please note: The input data will be such that a route which meets the conditions from the text will always exist.

SAMPLE TESTS

input	**input**

3 4
1 2 3 0
2 3 3 0
3 1 0 100
2 1 0 4
	5 7
	1 2 4 1
	2 3 5 1
	3 1 1 6
	1 3 15 5
	2 4 7 5
	4 5 1 4
	5 3 1 0

output	**output**
2 1	5 2