

ICPC Central Russia Regional Contest - 2020

A. Motivation problems

2 seconds, 256 megabytes

It is difficult to participate in Olympiads when you do not understand how they can help you in the future. Thus, after another failure, the spirit in the team dropped and the wish to compose Olympiads disappeared completely. The coach noticed that in time and decided to do something about it – after all, he understood what participation in such events could lead to. After much thought, he realized – you just need to show the team its prospects! Then, for simplicity, he presented all possible career development options arising after participating in the Olympiads in the form of a root tree. Each of its node was some position in a prestigious IT company, and the edges were drawn between positions only if career growth was possible from one position to another. Naturally, participation in the Olympiads made it easier to move up that tree, and the coach hurried up to show the tree to the team!

However, not all was as simple as that. The fact was that the team had absolutely no knowledge of those difficult positions and transitions. Therefore, the coach needed to choose several positions with beautiful and loud names and tell the team that after participating in the Olympiads, they could take up those positions. However, that choice was also limited. The fact was that the team was quite friendly and didn't want to occupy too diverse positions within the company. Therefore, they were to be selected in such a way that the highest position, from which one could still get into each of the selected ones, was far from the root as possible.

Help the coach to motivate the guys — find the necessary set of positions.

Formally speaking, you are given a rooted tree with N vertices, and you need to choose K vertices so that their least common ancestor could be as far from the root as possible.

The smallest common ancestor of several vertices is such a vertex that its distance from the root is maximal and from it one could get to any of the selected vertices if moving in the direction from the root.

Input

In the first line of the input you are given two numbers — N and K ($1 \leq K \leq N \leq 10^5$) — the number of positions in general and the number of positions that the coach should find.

The next line gives you $N - 1$ number, each of which does not exceed N . The i -th of them contains the number of the position from which one can go up to the $i + 1$ -th position. Position 1 is considered to be the root one. It is guaranteed that this structure is a tree rooted at the first vertex.

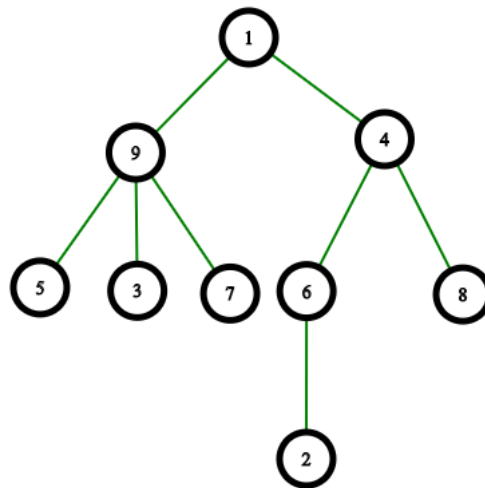
Output

For an answer, type a single line of K numbers randomly — the numbers of the positions that the coach should choose, or -1 if there is no answer. If there are several answers, it is allowed to output any of them.

input
5 2
5 1 1 1
output
5 2

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

In the second test, the graph looks like that:



The following triplets can also be output as an answer:

(5, 3, 7); (8, 4, 2); (9, 5, 7);

B. Time to reap the harvest

2.5 seconds, 256 megabytes

In addition to working as a system administrator, Dementy is very keen on growing valerian and adding it into his tea.

Dementy has a map of his one-dimensional field of valerian on his desk. Today he plans to collect his harvest. Dementy mows valerian stalks with a special mower, the blades of which hang above the ground at a certain height that can be adjusted (regulated). Dementy asks the question K times — how much valerian he will collect if he hangs the blades L centimeters above the ground?

Formally, the amount of valerian collected by Dementy is determined by the formula:

$$\sum_{i=1}^N \max(a_i - L, 0),$$

where L is the height of the blades, a_i is the height of the i -th bush of valerian.

Input

The first line of the input contains the number N — the number of valerian bushes in Dementy's garden ($1 \leq N \leq 10^5$).

The second line of the input contains N number of a_i — the height of the i -th valerian bush in centimeters ($0 \leq a_i \leq 10^9$).

The third line of the input contains the number K — the number of requests ($1 \leq K \leq 10^5$).

All the next K lines contain one number L_i ($0 \leq L_i \leq 10^9$) — the height to which Dementy wants to raise the blades of the lawn mower.

Output

Type on each of the K separate lines one number — the answers to the gardener's queries.

input
4
0 0 0 0
3
0
1
2
output
0
0
0

input
5 4 0 2 1 2 7 0 1 2 3 4 5 6
output
9 5 2 1 0 0 0

C. Lucky or not?

2 seconds, 256 megabytes

Not so long ago, Domin ordered a specific condition for one of the tasks in the online store, and is now expecting its delivery by mail. Domin lives in a country consisting of n cities. Between some of the cities u and v there is a two-way mail communication line that works only on the w -th day of the week (all in all there are 7 days in a week). Also, delivery can be made from any city to any other city (possibly through several other cities). The parcel can be sent in the morning from one city through an open communication channel and arrive in another in the evening, or be kept in the post office for several days until the communication starts working.

An important contest is coming up, so Domin wonders when he could add a new task for the participants. Help him and find the minimum number of days required to deliver the package from the warehouse of conditions to Domin. Domin starts expecting the package from the first day of the week, and the next morning after the package arrives in his city, he immediately runs to the post office to pick it up.

Input
The first line contains 4 numbers: n, m ($2 \leq n \leq 10^5$, $n - 1 \leq m \leq \min(\frac{n \cdot (n-1)}{2}, 2 \cdot 10^5)$) — the number of cities and posts, s and k ($1 \leq s, k \leq n$ and $s \neq k$) — a city with a warehouse and the city where Domin lives.

The next m lines describe mail messages, each with three numbers: u, v, w ($1 \leq u, v \leq n, u \neq v$ and $1 \leq w \leq 7$) — the numbers of the cities between which there is a communication and the day on which the mail communication works between the cities u and v .

Output
In a single line type the minimum number of days of expecting the parcel.

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

In the first example, the answer is 4 days, as the parcel, after travelling along the path 1 - 2 - 3 - 4 - 5, will arrive by the evening of the fourth day of the week and Domin will take it on the fifth day. And if it is sent directly, then it will arrive by the evening of the fifth day, that is, it will take 5 days to expect it.

D. Professor R's. Median

2 seconds, 256 megabytes

In mathematical statistics there is such a thing as the median of a series of numbers. The median of a series of numbers is a number in the middle of an ascending series of numbers if the number of figures is odd. If the number of figures in the row is even, then the median of the row is the half-sum of the two numbers in the middle of the ascending order.

Problems - Codeforces

But in Yaroslavl State University they are not lazy — scholars have invented a new concept — the median of Professor R. Let's define it. The median of Professor R. of a series of numbers is the number in this series closest to the half-sum of the minimum and maximum figures, and if there are several such numbers, the minimum in value is selected from them. Your task is to find Professor R's median in a given series of numbers.

Input
The first line contains a single integer n ($1 \leq n \leq 10^5$) — the number of elements.

The second line contains n space-separated integers a_i — elements of a series of numbers ($|a_i| \leq 2 \cdot 10^9$).

Output
In a single line type The median of Professor R. value.

input
5 1 1 1 1 1
output
1

input
3 1 2 3
output
2

E. The Highlanders' Tournament

2 seconds, 256 megabytes

In ancient times, which no one remembers, a Highlanders' tournament was held. This tournament was not a sport event rather it was a matter of life or death! Still, it was meant as a manifestation of life! The tournament was attended by n highlanders lined up in a row. Each fighter was endowed with strength! The strength of the i -th highlander was determined by the value of p_i . We should note that there were no highlanders with the same level of strength. The tournament itself was held in m stages. Before the start of each battle, the great leader Gyda chose several consecutive highlanders from the row. The selected group began the battle, where everyone fought for himself. The strongest fighters won the battle. After the battle, the winner stood back into his own place in the line, then the line closed and the fighters again stood shoulder to shoulder.

You need to state the strength of the fighters in the row after the end of the tournament.

Input
The first line contains two integers n and m . n is the number of highlanders fighting in the tournament ($1 \leq n \leq 2 \cdot 10^5$). m is the number of battles ($1 \leq m \leq 10^5$).

The next line contains the strength indicators p_i of each fighter ($1 \leq p_i \leq 10^9$).

The next m lines contain pairs of numbers l and r ($l \leq r$), which specify the range of numbers of the fighting highlanders in the row.

Output
State the strength of the fighters after the end of the tournament.

input
7 4 48 1 57 25 69 26 88 1 2 2 3 2 5 1 2
output
88

input
10 3 8 27 4 1 9 2 10 66 43 13 1 4 1 2 2 4

output
27 66 43 13

F. Square transit

5 seconds, 256 megabytes

The government of the square country decided to earn extra money on transit rail transportation. Money was allocated for the new railway network, and the construction began. Typically, the network bandwidth was not considered until after the completion of the project. Help the government determine the maximum train length that can transit a square country.

A square country has a width of x_{max} ($1 \leq x_{max} \leq 10^4$) and a length of y_{max} ($1 \leq y_{max} \leq 10^4$). The railway network consists of n ($1 \leq n \leq 10^4$) switches and two customs houses, which are connected by m ($1 \leq m \leq 30000$) straight railway segments. The coordinates of the switches (x_i, y_i) are integers, the coordinates of the customs points $(s, 0)$ and (t, y_{max}) are also integers, and lie on the opposite borders. Sections of the railway road are connected only in switch points (and customs points), and do not intersect at other points. The coordinates of all switches and customs points are different figures.

Through a segment of the railway road, between two switches, a train of length not exceeding the length of this same segment of road can easily pass. The length of the road segment connecting two switches (or customs) is the distance between points on the plane. The length of a train is the number of wagons in it. All cars are the same and have a length of 1. Thus, a train of no more than 4 cars can pass through a segment of length 4.5, and no more than 10 cars can pass through a segment of length 10.

The train can move along the railroad in any direction. Moreover, it can stop at any moment and change the direction of movement for the opposite. The train cannot occupy more than one switch at a time (but can touch them at its ends). When passing the switch point, the train cannot deviate for more than 60 degrees from the previous direction of movement (that is, the train cannot bend arbitrarily – the angle between adjacent cars cannot be sharper than 120 degrees).

Using the map of the railways of a square country, determine the maximum length of the train, which can go between two customs. It is assumed that from abroad, the train can leave on any track adjacent to the customs point.

Input

The first line contains 2 integers - the country sizes x_{max} ($1 \leq x_{max} \leq 10^4$) and y_{max} ($1 \leq y_{max} \leq 10^4$).

The second line contains the integer x -coordinates of customs s and t ($0 \leq s, t \leq x_{max}$).

The third line contains 2 integers – the number of switches n ($1 \leq n \leq 10^4$) and the number of railway segments m ($1 \leq m \leq 30000$).

Then come n lines with integer pairs x_i and y_i ($0 \leq x_i \leq x_{max}; 0 \leq y_i \leq y_{max}$) - coordinates of arrows.

They are followed by m lines describing the railway sections. Each segment is described by a pair of integers u_i and v_i ($0 \leq u_i, v_i \leq n + 1$) – the numbers of the connected switch points. The switches are numbered from 1 to n in the order they appear in the input. Customs office with coordinates $(s, 0)$ has number 0, and customs office with coordinates (t, y_{max}) has number $n + 1$.

Sections of the railroad do not intersect, and do not have a common point other than switches (or customs). The segment cannot connect the switch onto itself. Two switches cannot be connected more than once.

Output

If no train can pass between customs, output 0. Otherwise, type an integer – the maximum number of cars in a train.

input
6 16 0 0 4 5 0 4 3 8 0 12 6 8 0 1 1 2 2 3 3 5 2 4
output
3

G. Progress bar

1 second, 256 megabytes

Progress bars are often used to display the current state of a running process. One of the types of such strings is block line. A block line consists of n identical blocks. To display progress, the percentage of which is p ($0 \leq p \leq 1$), this number is multiplied by n , and the resulting number is rounded to the nearest integer. As a result we get the number $k = round(p \cdot n)$. Further, k blocks are displayed in the progress line.

If processes are fast, it is very seldom that we see a completely filled line, so n is not always possible to determine. Suppose we managed to count the number of blocks k_1 for the process that completed a third of the work ($p = 1/3$), and the number k_2 for the process that worked half ($p = 1/2$). Determine which numbers n match these conditions, or report that there are no such numbers.

Input

The only line contains two numbers k_1 and k_2 ($1 \leq k_1 \leq k_2 \leq 10^6$), separated by a space.

Output

If there is no solution, type 0 (zero). If there are suitable n , then type them in ascending order, on one line, separated by a space.

input
3 5
output
9 10

input
3 4
output
8

input
4 5
output
0

H. Chess knight on the curb stone

0.25 seconds, 256 megabytes

From a checkered field of n rows and m columns ($5 \leq n, m \leq 1000$), the inner part of size $(n - 4) \times (m - 4)$ was cut out so that on the sides of it there remained strips of 2 cells in width.

There is a chess knight on the top left square of the field. Is it possible, using only narrow strips of 2 cells width, to go around the cutout with the knight, and return to the original (initial) cell? And if it is possible, what is the minimum number of moves required for this (the knight follows the chess rules)?

For example, if there is a field 5×5 (the square $(3, 3)$ is cut out), then the knight can go around the cut in 4 moves:
 $(1, 1) \rightarrow (3, 2) \rightarrow (4, 4) \rightarrow (2, 3) \rightarrow (1, 1)$.

Note. At the moment of movement, the knight cannot jump over the cut cells.

Input

The size of the field is given as two integers n and m ($5 \leq n, m \leq 1000$), separated by a space.
Output
Output 0 if it is impossible to walk around the cutout, or the required number of the knight's moves for such a walk is minimal.

input
5 5
output
4

input
20 15
output
30

input
1000 1000
output
1996

input
5 6
output
6

I. Pharaoh hEx

0.5 seconds, 256 megabytes

Pharaoh hEx ruled a square country many thousands of years ago. Like all previous pharaohs, the main goal of his life was the construction of a magnificent pyramid. hEx was vain and very eager to out-perform its predecessors. How? By building a higher pyramid, of course!

The fact is that the pyramids of all previous pharaohs were of the same size, and there were serious reasons for this. For the construction of the pyramid it is necessary to lay a solid foundation of extremely hard granite. Granite of the required quality was mined at the only quarry. It was in the form of absolutely identical cubes. The transportation of these cubes was not a simple matter. Exactly n ($1 \leq n \leq 10^9$) of such granite blocks were delivered from the quarry in one supply.

Since the base of the pyramid must be a perfect square, all previous pharaohs simply performed n deliveries and built pyramids on the base of n^2 blocks. Pharaoh hEx wants to build his pyramid on a larger foundation. He has already planned n required deliveries of n cubic stones, but additional shipments will be needed to enlarge the foundation. Pharaoh wants to make the minimum possible number of over-scheduled transportations p . He also requires that the volume of single shipment does not decrease (all the same n blocks) and that after laying the square foundation there won't be extra blocks left.

Let's look at an example. Let $n = 4$, then for the 4 scheduled transportations 16 blocks will be received, of which a 4×4 foundation can be laid. If we add one over-planned transportation ($p = 1$), then there will be 20 stone cubes, from which it will not be possible to lay out a square, which means that p cannot be one. Similarly, for $p = 2, 3, 4$ we get 24, 28, and 32 cubes, which also does not allow us to lay them in the form of a square without excess. Finally, with the 5 over-planned deliveries, we get 36 cubic blocks, from which you can build a 6×6 foundation. So for $n = 4$ we get $p = 5$.

Input
The input data consists of a single integer number n ($1 \leq n \leq 10^9$).

Output
It is necessary to calculate the number of over-scheduled transportations p ($p > 0$).

input
1
output
3

input
2
output
6

input
3
output
9

input
4
output
5

J. Connect and Disconnect 1

1 second, 256 megabytes

While playing one of the popular games, programmer Vasya faced an interesting task.

The game contains producers and consumers of resources, which can be connected by conveyor belts. The conveyor works if it has a working input and output device. If it is not connected to an output device, the conveyor eventually overflows and stops.

For more flexible control, conveyor belts can be connected using a device called a «merger» or disconnected using a «splitter».

The «merger» has 3 inputs and one output. Looking through the inputs one by one, the device shifts items from the input conveyor belts to the output. For the correct operation of the «merger», it is necessary that the items are supplied by the conveyor to at least one input and removed from the output (if the output conveyor has nowhere to transport the items, or the output conveyor stops due to overflow, the device stops).

The «splitter» has 1 input and 3 outputs. Receiving items from the input conveyor, the «splitter» moves them evenly, one by one, onto the connected output conveyors. That is, if all three outputs are connected to running conveyors, then the input flow will be divided into 3 identical ($1/3$ each), and if only 2 outputs are connected, then the flow will be divided into 2 identical ($1/2$ each).

In the game, for production, it is often necessary to divide the flow into 2 sub-flows, in a certain ratio. On the Game Forum, Vasya discovered several standard schemes for dividing the flow in specified proportions. Looking at these diagrams, Vasya wondered if it was possible to come up with an algorithm for any ratio using a reasonable number of auxiliary devices?

Suppose we have a ratio $n : m$ ($1 \leq n, m; n + m \leq 10^6$). Suggest a scheme for connecting «splitters» and «mergers» that splits the input flow in the proportion n to m , using a total of no more than 48 «splitters» and «mergers». There should be no stopped conveyors and devices in your scheme. If there are several such schemes, any of them is considered correct.

Input
The only input line contains 2 integer numbers n and m ($1 \leq n, m; n + m \leq 10^6; gcd(n, m) = 1$), separated by a space.

Output
If there is no solution, then the output should contain a single number 0. If there is a solution, then the first line of the output contains an integer number k ($0 < k \leq 48$). Then k lines follow, describing the connection scheme of «splitters» and «mergers». Each line contains a description of one device. Devices are numbered from 1 to k in the order they appear.

The description of «splitter» begins with a capital Latin letter «S». This is followed by a space and 3 integer numbers separated by spaces – the numbers of the devices to which the outputs of the device are connected. If the output is not connected, then 0 (zero) should be stated as the number.

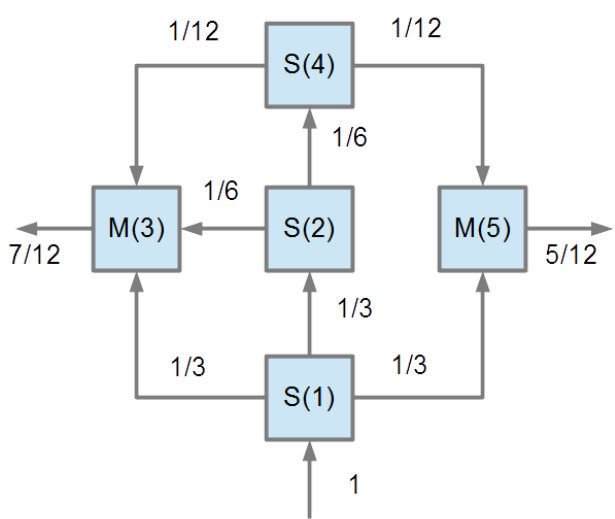
The description of «merger» begins with a capital Latin letter «M». This is followed by a space and an integer number – the device number to which the connected flow is sent.

The split input flow always goes to device number 1. The result of the split should go to special «devices» numbered -1 and -2 . These «devices» can only be used once.

input
7 5
output
5 S 2 3 5 S 3 4 0 M -1 S 3 5 0 M -2

input
1 4
output
5 M 2 S 3 4 5 S -1 4 5 M -2 M 1

Scheme 7 : 5 splitting are follow:



K. Divide and Connect 2
1 second, 256 megabytes

While playing one of the popular games, programmer Vasya faced an interesting task.

The game contains producers and consumers of resources, which can be connected by conveyor belts. The conveyor works if it has a working input and output device. If it is not connected to an output device, the conveyor eventually overflows and stops.

For more flexible control, conveyor belts can be connected using a device called a «merger» or disconnected using a «splitter».

The «merger» has 3 inputs and one output. Looking through the inputs one by one, the device shifts items from the input conveyor belts to the output. For the correct operation of the «merger», it is necessary that the items are supplied by the conveyor to at least one input and removed from the output (if the output conveyor has nowhere to transport the items, or the output conveyor stops due to overflow, the device stops).

The «splitter» has 1 input and 3 outputs. Receiving items from the input conveyor, the «splitter» moves them evenly, one by one, onto the connected output conveyors. That is, if all three outputs are connected to running conveyors, then the input flow will be divided into 3 identical ($1/3$ each), and if only 2 outputs are connected, then the flow will be divided into 2 identical ($1/2$ each).

In the game, for production, it is often necessary to divide the flow into 2 sub-flows, in a certain ratio. On the Game Forum, Vasya discovered several standard schemes for dividing the flow in specified proportions. Looking at these schemes, Vasya wondered do they work correctly?

Problems - Codeforces

Write a program that, according to a given scheme, will determine in what ratio the input flow is divided. All tested schemes are correct, that is, they do not contain idle devices, they accept one flow as input, and output two.

Input
The first line of input contains integer number k ($0 < k \leq 32$). Then k lines follow, describing the connection scheme of «splitters» and «mergers». Each line contains a description of one device. Devices are numbered from 1 to k in the order they appear.

The description of «splitter» begins with a capital Latin letter «S». This is followed by 3 integer numbers separated by spaces – the numbers of the devices to which the outputs of the device are connected. If the output is not connected, then 0 (zero) is stated as the number.

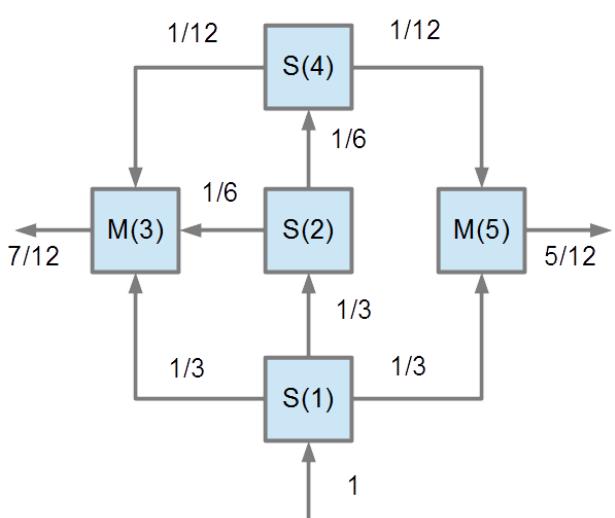
The description of «merger» begins with a capital Latin letter «M». This is followed by an integer number - the device number to which the combined flow is sent. The split input flow always goes to device number 1. The result of the split goes to special «devices» numbered -1 and -2 . These «devices» are used strictly one time.

Each device used has at least one connected input and output. From the first (input) device there is a path to any other device, and from it – to one of the outputs.

Output
The output consists of two numbers separated by a space: the ratio of flows going to outputs -1 and -2 .

input
5 S 2 3 5 S 3 4 0 M -1 S 3 5 0 M -2
output
7 5

Picture for example:



L. Cipher
2 seconds, 256 megabytes

The Prime Minister of Borland, fearing that his e-mail could be hacked and the correspondence would go public, turned to the computer security laboratory of Borland University, whose employees came up with a new encryption method, for help. They found out that the Prime Minister wrote texts consisting of numbers and lowercase Latin letters, the number of characters in which was equal exactly to the powers of two. They came up with the following method for encrypting a message – all the characters were glued together into one large string of length 2^n . Then they flipped the line, then divided it into two parts and flipped each half, then divided it into four parts and flipped every quarter, ... then they divided it by 2^{n-1} and flipped each part. However, a little difficulty arose – scientists had not yet figured out how to restore the originally specified string. Your task is to help them write a program to decrypt the message.

Input
A string of length 2^n ($0 \leq n \leq 13$) – an encrypted message – is given for input. It

Output

As an answer, output a string of length 2^n , consisting of lowercase Latin letters and numbers, which is the answer to the problem.

input
2301
output
0123

input
mikkmfz
output
fmznmkk

M. Beautiful hockey

1 second, 256 megabytes

Programmer Vasya loves hockey and is a fan of the best club in the world "200 OK". In hockey, Vasya likes two things: the victory of his favorite team and a beautiful score on the scoreboard. At the same time, Vasya understands the beauty of a score in a peculiar way. So, he calls a game *beautiful* if each period ends with a beautiful result.

It should be stated, that a hockey match consists of three periods, and the result of each period is recorded with a pair of non-negative integers. For example, the results of a match might look like this: (3 : 0), (1 : 3), (1 : 1). Such a record means that in the first period, the first team scored three goals, the second — 0; in the second period, the first scored 1 goal, the second — 3; finally, in the third period, both teams scored one goal each. In such game, the first team won with a score of 5 : 4.

Vasya, enchanted by the binary code, considers the game to be beautiful if each period ends with one of the following results: (0 : 0), (0 : 1), (1 : 0), (1 : 1). Vasya is an experienced fan. He realized long ago that there are exactly 22 different beautiful games in which the first team wins (two games are considered different if they differ in the result of at least one period). However, this is only valid for a 3-period match.

Now Vasya is going to the ice arena for the next match. He is trying to count - how many different, consisting of n periods, beautiful games, in which the first team wins, are there in total? Help Vasya find a solution before the match starts.

Input

The only line contains number n — the number of periods in the game ($1 \leq n \leq 50000$).

Output

In a single line, calculate the remainder of dividing the desired number of games by $10^9 + 7$.

input
1
output
1

input
3
output
22

N. Contest with bug

2 seconds, 256 megabytes

A team of Champions in sports programming participated in important competitions that lasted K minutes. They were held according to the standard rules, but the guys learned that the scoring system did not always correctly calculate the penalty time. Namely, when the penalty time became more than 23 hours and 59 minutes, 24 hours were subtracted from it.

Problems - Codeforces

Having received a set of N problems, the champions could instantly determine for each of them the time t_i (in minutes) that they would need to solve and debug the program for solving the i -th problem. As a matter of fact, the Champions had always submitted all the tasks on the first attempt. As true professionals, the guys did not waste a single minute during the competition. As soon as they passed one task, they immediately began to solve the next one. Knowing about the error in the calculation of penalty minutes, they wanted to determine the best result that they could show if they passed the necessary tasks in the right order. Create a program that will help them do this.

Input

The first line contains two positive integers separated by a space — the duration of the competition K in minutes ($1 \leq K \leq 1000$) and the number of tasks N ($1 \leq N \leq 10$).

In the second line, separated by a space, N natural numbers t_i are written, ($1 \leq t_i \leq 1000$) — the solution time in minutes of the i -th problem.

Output

Output two integers, that determine the best possible result of the team — the number of tasks M , that the team will have time to pass during the competition and the total penalty time T , which is calculated in the erroneous way described above.

input
75 5 5 25 15 10 20
output
5 175

input
480 8 3 150 160 2 165 200 2 300
output
5 0

In the first example, it is advantageous for the team to adhere to standard tactics and solve problems in the ascending order of their solution duration, because they will not be able to use an error (if they have it) in the time calculation.

In the second example, the team will have time to solve only 5 out of 8 tasks. If they pass the tasks in the order of 5, 2, 1, 7, 4, then the total penalty time will be $165 + (165 + 150) + (165 + 150 + 3) + (165 + 150 + 3 + 2) + (165 + 1$ minutes, which will be calculated in 0 minutes.

[Codeforces](#) (c) Copyright 2010-2024 Mike Mirzayanov
The only programming contests Web 2.0 platform