Gracias Sponsors!

Organizador

Diamond

Gold

Contest 6 - Inicial

A

Sonya and Exhibition

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Sonya decided to organize an exhibition of flowers. Since the girl likes only roses and lilies, she decided that only these two kinds of flowers should be in this exhibition.

There are n flowers in a row in the exhibition. Sonya can put either a rose or a lily in the i-th position. Thus each of n positions should contain exactly one flower: a rose or a lily.

She knows that exactly m people will visit this exhibition. The i-th visitor will visit all flowers from l_i to r_i inclusive. The girl knows that each segment has its own *beauty* that is equal to the product of the number of roses and the number of lilies.

Sonya wants her exhibition to be liked by a lot of people. That is why she wants to put the flowers in such way that the sum of *beauties* of all segments would be maximum possible.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 10^3$) — the number of flowers and visitors respectively.

Each of the next m lines contains two integers l_i and r_i ($1 \leq l_i \leq r_i \leq n$), meaning that i-th visitor will visit all flowers from l_i to r_i inclusive.

Output

Print the string of n characters. The i-th symbol should be «0» if you want to put a rose in the i-th position, otherwise «1» if you want to put a lily.

If there are multiple answers, print any.

Examples

input Copy

```
5 3
1 3
2 4
2 5
```

output Copy

```
01100
```

input Copy

```
6 3
5 6
1 4
4 6
```

output Copy

```
110010
```

Note

In the first example, Sonya can put roses in the first, fourth, and fifth positions, and lilies in the second and third positions;

- in the segment $[1 \ldots 3]$, there are one rose and two lilies, so the *beauty* is equal to $1 \cdot 2 = 2$;

- in the segment $[2\ldots4]$, there are one rose and two lilies, so the *beauty* is equal to $1 \cdot 2 = 2$;
- in the segment $[2\ldots5]$, there are two roses and two lilies, so the *beauty* is equal to $2 \cdot 2 = 4$.

The total *beauty* is equal to $2 + 2 + 4 = 8$.

In the second example, Sonya can put roses in the third, fourth, and sixth positions, and lilies in the first, second, and fifth positions;

- in the segment $[5\ldots6]$, there are one rose and one lily, so the *beauty* is equal to $1 \cdot 1 = 1$;
- in the segment $[1\ldots4]$, there are two roses and two lilies, so the *beauty* is equal to $2 \cdot 2 = 4$;
- in the segment $[4\ldots6]$, there are two roses and one lily, so the *beauty* is equal to $2 \cdot 1 = 2$.

The total *beauty* is equal to $1 + 4 + 2 = 7$.

B Cashback

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Since you are the best Wraith King, Nizhniy Magazin «Mir» at the centre of Vinnytsia is offering you a discount.

You are given an array a of length n and an integer c.

The value of some array b of length k is the sum of its elements except for the $\lfloor \frac{k}{c} \rfloor$ smallest. For example, the value of the array $[3, 1, 6, 5, 2]$ with $c = 2$ is $3 + 6 + 5 = 14$.

Among all possible partitions of a into contiguous subarrays output the smallest possible sum of the values of these subarrays.

Input

The first line contains integers n and c ($1 \le n, c \le 100\,000$).

The second line contains n integers a_i ($1 \le a_i \le 10^9$) — elements of a.

Output

Output a single integer — the smallest possible sum of values of these subarrays of some partition of a.

Examples

input

```
3 5
1 2 3
```

output

```
6
```

input

```
12 10
1 1 10 10 10 10 10 10 9 10 10 10
```

output

```
92
```

input

```
7 2
2 3 6 4 5 7 1
```

output

```
17
```

input

```
8 4
1 3 4 5 5 3 4 1
```

output

```
23
```

Note

In the first example any partition yields 6 as the sum.

In the second example one of the optimal partitions is $[1, 1], [10, 10, 10, 10, 10, 10, 9, 10, 10, 10]$ with the values 2 and 90 respectively.

In the third example one of the optimal partitions is $[2, 3], [6, 4, 5, 7], [1]$ with the values 3, 13 and 1 respectively.

In the fourth example one of the optimal partitions is $[1], [3, 4, 5, 5, 3, 4], [1]$ with the values 1, 21 and 1 respectively.

C Light It Up

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently, you bought a brand new smart lamp with programming features. At first, you set up a schedule to the lamp. Every day it will turn power on at moment 0 and turn power off at moment M. Moreover, the lamp allows you to set a program of switching its state (states are "lights on" and "lights off"). Unfortunately, some program is already installed into the lamp.

The lamp allows only *good* programs. Good program can be represented as a non-empty array a, where $0 < a_1 < a_2 < \dots < a_{|a|} < M$. All a_i must be integers. Of course, preinstalled program is a good program.

The lamp follows program a in next manner: at moment 0 turns power and light on. Then at moment a_i the lamp flips its state to opposite (if it was lit, it turns off, and vice versa). The state of the lamp flips instantly: for example, if you turn the light off at moment 1 and then do nothing, the total time when the lamp is lit will be 1. Finally, at moment M the lamp is turning its power off regardless of its state.

Since you are not among those people who read instructions, and you don't understand the language it's written in, you realize (after some testing) the only possible way to alter the preinstalled program. You can **insert at most one** element into the program a, so it still should be a *good* program after alteration. Insertion can be done between any pair of consecutive elements of a, or even at the begining or at the end of a.

Find such a way to alter the program that the total time when the lamp is lit is maximum possible. Maybe you should leave program untouched. If the lamp is lit from x till moment y, then its lit for $y - x$ units of time. Segments of time when the lamp is lit are summed up.

Input

First line contains two space separated integers n and M ($1 \le n \le 10^5$, $2 \le M \le 10^9$) — the length of program a and the moment when power turns off.

Second line contains n space separated integers $a_1, a_2, \dots, a_n$ ($0 < a_1 < a_2 < \dots < a_n < M$) — initially installed program a.

Output

Print the only integer — maximum possible total time when the lamp is lit.

Examples

input Copy

```
3 10
4 6 7
```

output Copy

```
8
```

input Copy

```
2 12
1 10
```

output Copy

```
9
```

input Copy

```
2 7
3 4
```

output Copy

```
6
```

Note

In the first example, one of possible optimal solutions is to insert value $x = 3$ before a_1, so program will be $[3, 4, 6, 7]$ and time of lamp being lit equals $(3 - 0) + (6 - 4) + (10 - 7) = 8$. Other possible solution is to insert $x = 5$ in appropriate place.

In the second example, there is only one optimal solution: to insert $x = 2$ between a_1 and a_2. Program will become $[1, 2, 10]$, and answer will be $(1 - 0) + (10 - 2) = 9$.

In the third example, optimal answer is to leave program untouched, so answer will be $(3 - 0) + (7 - 4) = 6$.

D

Sonya and Hotels

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Sonya decided that having her own hotel business is the best way of earning money because she can profit and rest wherever she wants.

The country where Sonya lives is an endless line. There is a city in each integer coordinate on this line. She has n hotels, where the i-th hotel is located in the city with coordinate x_i. Sonya is a smart girl, so she does not open two or more hotels in the same city.

Sonya understands that her business needs to be expanded by opening new hotels, so she decides to build one more. She wants to make the minimum distance from this hotel to all others to be equal to d. The girl understands that there are many possible locations to construct such a hotel. Thus she wants to know the number of possible coordinates of the cities where she can build a new hotel.

Because Sonya is lounging in a jacuzzi in one of her hotels, she is asking you to find the number of cities where she can build a new hotel so that the minimum distance from the original n hotels to the new one is equal to d.

Input

The first line contains two integers n and d ($1 \leq n \leq 100$, $1 \leq d \leq 10^9$) — the number of Sonya's hotels and the needed minimum distance from a new hotel to all others.

The second line contains n different integers in strictly increasing order $x_1, x_2, \ldots, x_n$ ($-10^9 \leq x_i \leq 10^9$) — coordinates of Sonya's hotels.

Output

Print the number of cities where Sonya can build a new hotel so that the minimum distance from this hotel to all others is equal to d.

Examples

input

```
4 3
-3 2 9 16
```

output

```
6
```

input

```
5 2
4 8 11 18 19
```

output

```
5
```

Note

In the first example, there are 6 possible cities where Sonya can build a hotel. These cities have coordinates -6, 5, 6, 12, 13, and 19.

In the second example, there are 5 possible cities where Sonya can build a hotel. These cities have coordinates 2, 6, 13, 16, and 21.

E

Points and Powers of Two

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n distinct points on a coordinate line, the coordinate of i-th point equals to x_i. Choose a subset of the given set of points such that the distance between each pair of points in a subset is an integral power of two. It is necessary to consider each pair of points, not only adjacent. Note that any subset containing one element satisfies the condition above. Among all these subsets, choose a subset with maximum possible size.

In other words, you have to choose the maximum possible number of points $x_{i_1}, x_{i_2}, \dots, x_{i_m}$ such that for each pair x_{i_j}, x_{i_k} it is true that $|x_{i_j} - x_{i_k}| = 2^d$ where d is some non-negative integer number (not necessarily the same for each pair of points).

Input

The first line contains one integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of points.

The second line contains n pairwise distinct integers $x_1, x_2, \dots, x_n$ $(-10^9 \le x_i \le 10^9)$ — the coordinates of points.

Output

In the first line print m — the maximum possible number of points in a subset that satisfies the conditions described above.

In the second line print m integers — the coordinates of points in the subset you have chosen.

If there are multiple answers, print any of them.

Examples

input

```
6
3 5 4 7 10 12
```

output

```
3
7 3 5
```

input

```
5
-1 2 5 8 11
```

output

```
1
8
```

Note

In the first example the answer is $[7, 3, 5]$. Note, that $|7-3| = 4 = 2^2$, $|7-5| = 2 = 2^1$ and $|3-5| = 2 = 2^1$. You can't find a subset having more points satisfying the required property.

F Frog Jumps

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There is a frog staying to the left of the string $s = s_1 s_2 \ldots s_n$ consisting of n characters (to be more precise, the frog initially stays at the cell 0). Each character of s is either 'L' or 'R'. It means that if the frog is staying at the i-th cell and the i-th character is 'L', the frog can jump only to the left. If the frog is staying at the i-th cell and the i-th character is 'R', the frog can jump only to the right. **The frog can jump only to the right from the cell 0**.

Note that the frog can jump into the same cell twice and can perform as many jumps as it needs.

The frog wants to reach the $n+1$-th cell. The frog chooses some **positive integer** value d **before the first jump** (and cannot change it later) and jumps by no more than d cells at once. I.e. if the i-th character is 'L' then the frog can jump to any cell in a range $[max(0, i-d); i-1]$, and if the i-th character is 'R' then the frog can jump to any cell in a range $[i+1; min(n+1; i+d)]$.

The frog doesn't want to jump far, so your task is to find the minimum possible value of d such that the frog can reach the cell $n+1$ from the cell 0 if it can jump by no more than d cells at once. **It is guaranteed that it is always possible to reach $n+1$ from 0**.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \le t \le 10^4$) — the number of test cases.

The next t lines describe test cases. The i-th test case is described as a string s consisting of at least 1 and at most $2 \cdot 10^5$ characters 'L' and 'R'.

It is guaranteed that the sum of lengths of strings over all test cases does not exceed $2 \cdot 10^5$ ($\sum |s| \le 2 \cdot 10^5$).

Output

For each test case, print the answer — the minimum possible value of d such that the frog can reach the cell $n+1$ from the cell 0 if it jumps by no more than d at once.

Example

input Copy

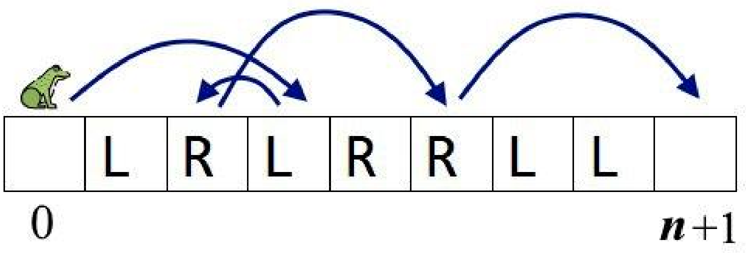
```
6
LRLRRLL
L
LLR
RRRR
LLLLLL
R
```

output Copy

```
3
2
3
1
7
1
```

Note

The picture describing the first test case of the example and one of the possible answers:



In the second test case of the example, the frog can only jump directly from 0 to $n + 1$.

In the third test case of the example, the frog can choose $d = 3$, jump to the cell 3 from the cell 0 and then to the cell 4 from the cell 3.

In the fourth test case of the example, the frog can choose $d = 1$ and jump 5 times to the right.

In the fifth test case of the example, the frog can only jump directly from 0 to $n + 1$.

In the sixth test case of the example, the frog can choose $d = 1$ and jump 2 times to the right.

G

Build String

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You desperately need to build some string t. For that you've got n more strings $s_1, s_2, ..., s_n$. To build string t, you are allowed to perform exactly $|t|$ ($|t|$ is the length of string t) operations on these strings. Each operation looks like that:

1. choose any non-empty string from strings $s_1, s_2, ..., s_n$;
2. choose an arbitrary character from the chosen string and write it on a piece of paper;
3. remove the chosen character from the chosen string.

Note that after you perform the described operation, the total number of characters in strings $s_1, s_2, ..., s_n$ decreases by 1. We are assumed to build string t, if the characters, written on the piece of paper, in the order of performed operations form string t.

There are other limitations, though. For each string s_i you know number a_i — the maximum number of characters you are allowed to delete from string s_i. You also know that each operation that results in deleting a character from string s_i, costs i rubles. That is, an operation on string s_1 is the cheapest (it costs 1 ruble), and the operation on string s_n is the most expensive one (it costs n rubles).

Your task is to count the minimum amount of money (in rubles) you will need to build string t by the given rules. Consider the cost of building string t to be the sum of prices of the operations you use.

Input

The first line of the input contains string t — the string that you need to build.

The second line contains a single integer n $(1 \le n \le 100)$ — the number of strings to which you are allowed to apply the described operation. Each of the next n lines contains a string and an integer. The i-th line contains space-separated string s_i and integer a_i $(0 \le a_i \le 100)$. Number a_i represents the maximum number of characters that can be deleted from string s_i.

All strings in the input only consist of lowercase English letters. All strings are non-empty. The lengths of all strings do not exceed 100 characters.

Output

Print a single number — the minimum money (in rubles) you need in order to build string t. If there is no solution, print -1.

Examples

input Copy

```
bbaze
3
bzb 2
aeb 3
ba 10
```

output Copy

```
8
```

input Copy

```
abacaba
4
aba 2
bcc 1
caa 2
bbb 5
```

output

```
18
```

input

```
xyz
4
axx 8
za 1
efg 4
t 1
```

output

```
-1
```

Note

Notes to the samples:

In the first sample from the first string you should take characters "b" and "z" with price 1 ruble, from the second string characters "a", "e" и "b" with price 2 rubles. The price of the string t in this case is $2·1 + 3·2 = 8$.

In the second sample from the first string you should take two characters "a" with price 1 ruble, from the second string character "c" with price 2 rubles, from the third string two characters "a" with price 3 rubles, from the fourth string two characters "b" with price 4 rubles. The price of the string t in this case is $2·1 + 1·2 + 2·3 + 2·4 = 18$.

In the third sample the solution doesn't exist because there is no character "y" in given strings.

H. Mahmoud and Ehab and the even-odd game

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mahmoud and Ehab play a game called the even-odd game. Ehab chooses his favorite integer n and then they take turns, starting from Mahmoud. In each player's turn, he has to choose an integer a and subtract it from n such that:

- $1 \le a \le n$.
- If it's Mahmoud's turn, a has to be even, but if it's Ehab's turn, a has to be odd.

If the current player can't choose any number satisfying the conditions, he loses. Can you determine the winner if they both play optimally?

Input

The only line contains an integer n $(1 \le n \le 10^9)$, the number at the beginning of the game.

Output

Output "`Mahmoud`" (without quotes) if Mahmoud wins and "`Ehab`" (without quotes) otherwise.

Examples

input

```
1
```

output

```
Ehab
```

input

```
2
```

output

```
Mahmoud
```

Note

In the first sample, Mahmoud can't choose any integer a initially because there is no positive even integer less than or equal to 1 so Ehab wins.

In the second sample, Mahmoud has to choose $a = 2$ and subtract it from n. It's Ehab's turn and $n = 0$. There is no positive odd integer less than or equal to 0 so Mahmoud wins.

I

Mahmoud and Ehab and the message

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mahmoud wants to send a message to his friend Ehab. Their language consists of n words numbered from 1 to n. Some words have the same meaning so there are k groups of words such that all the words in some group have the same meaning.

Mahmoud knows that the i-th word can be sent with cost a_i. For each word in his message, Mahmoud can either replace it with another word of the same meaning or leave it as it is. Can you help Mahmoud determine the minimum cost of sending the message?

The cost of sending the message is the sum of the costs of sending every word in it.

Input

The first line of input contains integers n, k and m $(1 \le k \le n \le 10^5, 1 \le m \le 10^5)$ — the number of words in their language, the number of groups of words, and the number of words in Mahmoud's message respectively.

The second line contains n strings consisting of lowercase English letters of length not exceeding 20 which represent the words. It's guaranteed that the words are **distinct**.

The third line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ where a_i is the cost of sending the i-th word.

The next k lines describe the groups of words of same meaning. The next k lines each start with an integer x $(1 \le x \le n)$ which means that there are x words in this group, followed by x integers which represent the indices of words in this group. It's guaranteed that each word appears in exactly one group.

The next line contains m space-separated words which represent Mahmoud's message. Each of these words appears in the list of language's words.

Output

The only line should contain the minimum cost to send the message after replacing some words (maybe none) with some words of the same meaning.

Examples

input Copy

```
5 4 4
i loser am the second
100 1 1 5 10
1 1
1 3
2 2 5
1 4
i am the second
```

output Copy

```
107
```

input Copy

```
5 4 4
i loser am the second
100 20 1 5 10
```

```
1 1
1 3
2 2 5
1 4
i am the second
```

output Copy

```
116
```

Note

In the first sample, Mahmoud should replace the word "`second`" with the word "`loser`" because it has less cost so the cost will be 100+1+5+1=107.

In the second sample, Mahmoud shouldn't do any replacement so the cost will be 100+1+5+10=116.

J

Diverse Team

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n students in a school class, the rating of the i-th student on Codehorses is a_i. You have to form a team consisting of k students ($1 \le k \le n$) such that the ratings of all team members **are distinct**.

If it is impossible to form a suitable team, print "NO" (without quotes). Otherwise print "YES", and then print k distinct numbers which should be the indices of students in the team you form. If there are multiple answers, print any of them.

Input

The first line contains two integers n and k ($1 \le k \le n \le 100$) — the number of students and the size of the team you have to form.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 100$), where a_i is the rating of i-th student.

Output

If it is impossible to form a suitable team, print "NO" (without quotes). Otherwise print "YES", and then print k distinct integers from 1 to n which should be the indices of students in the team you form. All the ratings of the students in the team should be distinct. You may print the indices in any order. If there are multiple answers, print any of them.

Assume that the students are numbered from 1 to n.

Examples

input Copy
```
5 3
15 13 15 15 12
```
output Copy
```
YES
1 2 5
```

input Copy
```
5 4
15 13 15 15 12
```
output Copy
```
NO
```

input Copy
```
4 4
20 10 40 30
```
output Copy
```
YES
1 2 3 4
```

Note

All possible answers for the first example:

- {1 2 5}
- {2 3 5}
- {2 4 5}

Note that the order does not matter.

K

Madoka and The Best University

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Madoka wants to enter to "*Novosibirsk State University*", but in the entrance exam she came across a very difficult task:

Given an integer n, it is required to calculate $\sum \operatorname{lcm}(c, \gcd(a, b))$, for all triples of positive integers (a, b, c), where $a + b + c = n$.

In this problem $\gcd(x, y)$ denotes the greatest common divisor of x and y, and $\operatorname{lcm}(x, y)$ denotes the least common multiple of x and y.

Solve this problem for Madoka and help her to enter to the best university!

Input

The first and the only line contains a single integer n ($3 \le n \le 10^5$).

Output

Print exactly one interger — $\sum \operatorname{lcm}(c, \gcd(a, b))$. Since the answer can be very large, then output it modulo $10^9 + 7$.

Examples

input

```
3
```

output

```
1
```

input

```
5
```

output

```
11
```

input

```
69228
```

output

```
778304278
```

Note

In the first example, there is only one suitable triple $(1, 1, 1)$. So the answer is $\operatorname{lcm}(1, \gcd(1, 1)) = \operatorname{lcm}(1, 1) = 1$.

In the second example,
$\operatorname{lcm}(1, \gcd(3, 1)) + \operatorname{lcm}(1, \gcd(2, 2)) + \operatorname{lcm}(1, \gcd(1, 3)) + \operatorname{lcm}(2, \gcd(2, 1)) + \operatorname{lcm}(2, \gcd(1, 2)) + \operatorname{lcm}(3, \gcd(1, 1$

L King Escape

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob are playing chess on a huge chessboard with dimensions $n \times n$. Alice has a single piece left — a queen, located at (a_x, a_y), while Bob has only the king standing at (b_x, b_y). Alice thinks that as her queen is dominating the chessboard, victory is hers.

But Bob has made a devious plan to seize the victory for himself — he needs to march his king to (c_x, c_y) in order to claim the victory for himself. As Alice is distracted by her sense of superiority, **she no longer moves any pieces around, and it is only Bob who makes any turns.**

Bob will win if he can move his king from (b_x, b_y) to (c_x, c_y) **without ever getting in check**. Remember that a king can move to any of the 8 adjacent squares. A king is in check if it is on the same rank (i.e. row), file (i.e. column), or diagonal as the enemy queen.

Find whether Bob can win or not.

Input

The first line contains a single integer n ($3 \le n \le 1000$) — the dimensions of the chessboard.

The second line contains two integers a_x and a_y ($1 \le a_x, a_y \le n$) — the coordinates of Alice's queen.

The third line contains two integers b_x and b_y ($1 \le b_x, b_y \le n$) — the coordinates of Bob's king.

The fourth line contains two integers c_x and c_y ($1 \le c_x, c_y \le n$) — the coordinates of the location that Bob wants to get to.

It is guaranteed that Bob's king is currently not in check and the target location is not in check either.

Furthermore, the king is not located on the same square as the queen (i.e. $a_x \neq b_x$ or $a_y \neq b_y$), and the target does coincide neither with the queen's position (i.e. $c_x \neq a_x$ or $c_y \neq a_y$) nor with the king's position (i.e. $c_x \neq b_x$ or $c_y \neq b_y$).

Output

Print "YES" (without quotes) if Bob can get from (b_x, b_y) to (c_x, c_y) without ever getting in check, otherwise print "NO".

You can print each letter in any case (upper or lower).

Examples

input

```
8
4 4
1 3
3 1
```

output

```
YES
```

input

```
8
4 4
```

```
2 3
1 6
```

output

```
NO
```

input

```
8
3 5
1 2
6 1
```

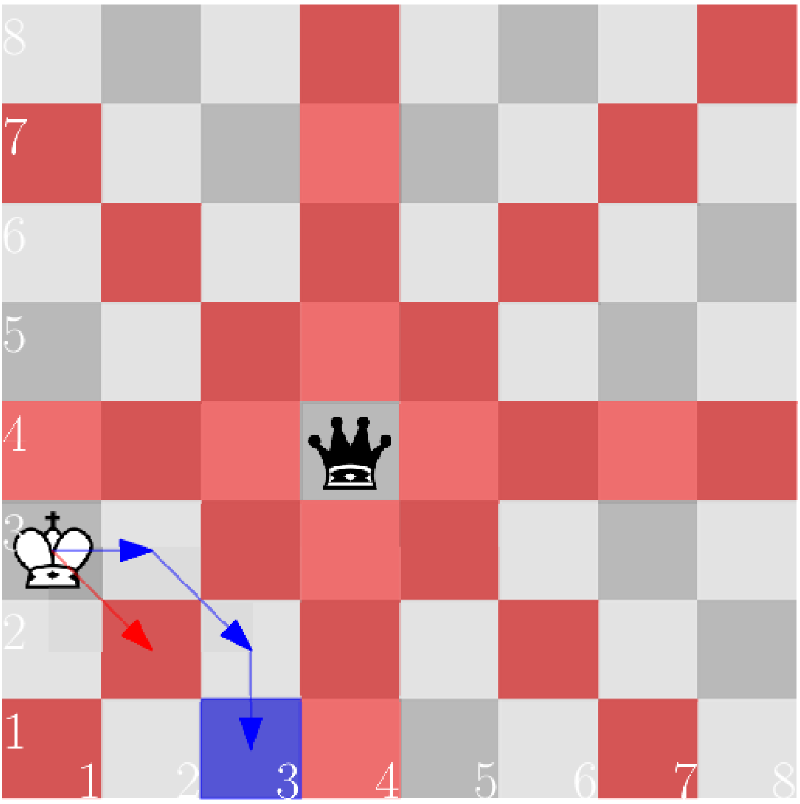
output

```
NO
```

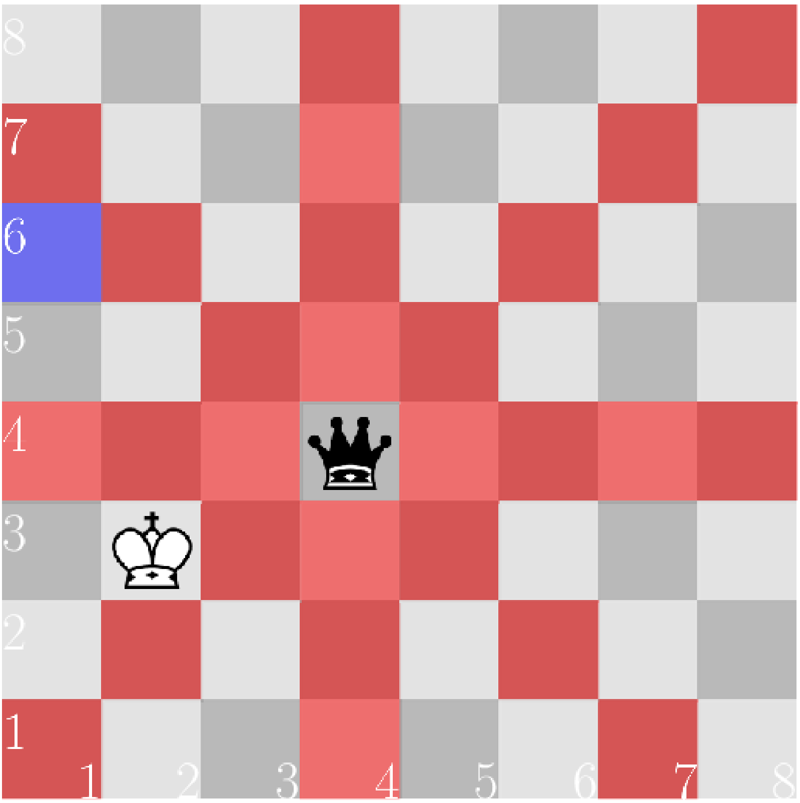
Note

In the diagrams below, the squares controlled by the black queen are marked red, and the target square is marked blue.

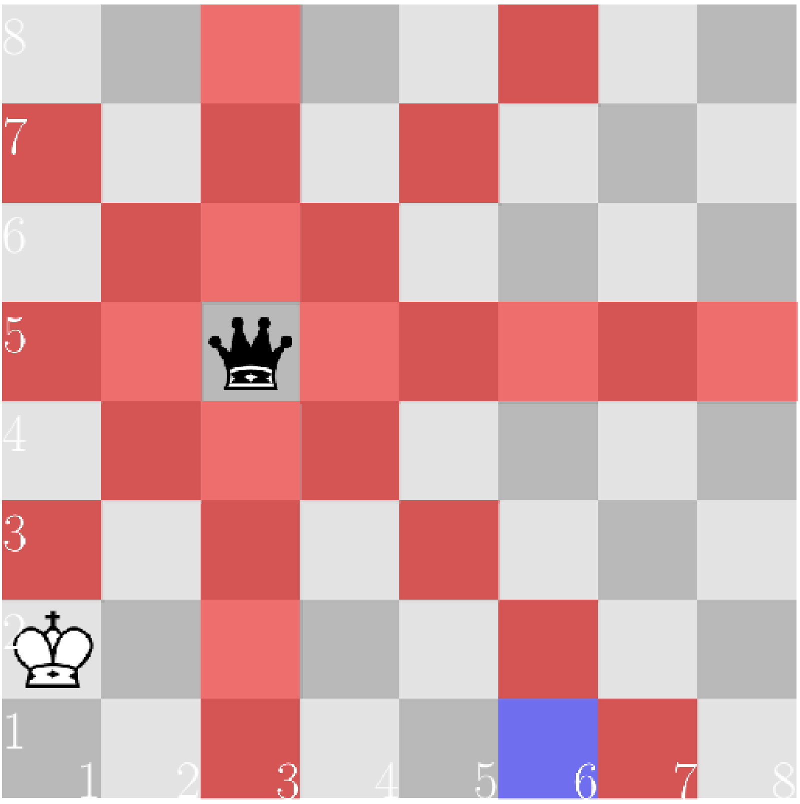
In the first case, the king can move, for instance, via the squares $(2, 3)$ and $(3, 2)$. Note that the direct route through $(2, 2)$ goes through check.



In the second case, the queen watches the fourth rank, and the king has no means of crossing it.



In the third case, the queen watches the third file.



M. Anton and Digits

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently Anton found a box with digits in his room. There are k_2 digits 2, k_3 digits 3, k_5 digits 5 and k_6 digits 6.

Anton's favorite integers are 32 and 256. He decided to compose this integers from digits he has. He wants to make the sum of these integers as large as possible. Help him solve this task!

Each digit can be used no more than once, i.e. the composed integers should contain no more than k_2 digits 2, k_3 digits 3 and so on. Of course, unused digits are not counted in the sum.

Input

The only line of the input contains four integers k_2, k_3, k_5 and k_6 — the number of digits 2, 3, 5 and 6 respectively ($0 \le k_2, k_3, k_5, k_6 \le 5 \cdot 10^6$).

Output

Print one integer — maximum possible sum of Anton's favorite integers that can be composed using digits from the box.

Examples

input

```
5 1 3 4
```

output

```
800
```

input

```
1 1 1 1
```

output

```
256
```

Note

In the first sample, there are five digits 2, one digit 3, three digits 5 and four digits 6. Anton can compose three integers 256 and one integer 32 to achieve the value $256 + 256 + 256 + 32 = 800$. Note, that there is one unused integer 2 and one unused integer 6. They are not counted in the answer.

In the second sample, the optimal answer is to create on integer 256, thus the answer is 256.