International Olympiad in Informatics
National Olympiad in Informatics
Hong Kong Delegation Team Formation Test 2010

Task Overview Sheet

	peanutsii	bridge	tiles	stamp
Input file	stdin			
Output file	stdout			
Time limit per test case	1 second	1 second	1 second	1 second
Memory limit	32 MB	32 MB	32 MB	32 MB
Compiler options in C	`-lm -w -O2 -static -static-libgcc`			
Compiler options in C++	`-lm -w -O2 -static -static-libgcc`			
Compiler options in Pascal	`-So -XS -v0 -O2`			
Number of test cases	10	10	10	10
Maximum points per test case	10	10	10	10
Maximum total points	100	100	100	100

WARNING: **`iostream` is sometimes too slow for input reading/output writing. C++ coders are advised to use `cstdio` instead.**
Example: **"`cin >> n`" can be written as "`scanf("%d", &n)`" for `int` variable `n`.**
"`cout << n << endl`" can be replaced by "`printf("%d\n", n)`" for `int` variable `n`.

Peanuts (II)

PROBLEM

Mr. Peanuts loves reading threads in a forum. Besides, he loves planting peanuts in his own farmland. He seeds the field with peanuts. After the harvest, his mother will ask him to share the peanuts with his friends.

Mr. Peanuts crops P different kinds of peanuts. He knows that his friends are so greedy that, each of them is satisfied if he or she is shared at least N different kinds of peanuts, otherwise he or she will no longer be friend with Mr. Peanuts.

This year Mr. Peanuts has P_i peanuts of the i^{th} kind. Peanuts are limited but his friends aren't. Please help him to find out at most how many of his friends will still be friend with him after sharing the peanuts?

INPUT

The first line of the input contains two integers P and N separated by single space. P is the number of different kinds of peanuts Mr. Peanuts has and each of his friend will be satisfied if he or she is shared at least N different kinds of peanuts.

Each of the i^{th} of the following P lines contains an integer P_i, representing the amount of i^{th} kind peanuts.

OUTPUT

Output should always consist of one line, which is the maximum number of Mr. Peanuts's friends being satisfied by sharing the peanuts optimally.

SAMPLE INPUT 1

```
2 2
5
6
```

SAMPLE OUTPUT 1

```
5
```

SAMPLE INPUT 2

```
3 2
3
4
5
```

SAMPLE OUTPUT 2

```
6
```

CONSTRAINTS

In all of the test cases,

- $1 \leq N \leq P \leq 50,000$
- $1 \leq P_i \leq 2^{31} - 1$

In 70% of the test cases,

- $1 \leq P \leq 1,000$

In 50% of the test cases,

- $1 \leq P \leq 50$
- $1 \leq P_i \leq 100$

Crossing Bridges at Night

PROBLEM

There are N islands, connected by M bidirectional bridges. The time needed to cross the i^{th} bridge is T_i. At one night, Little Gary wanted to travel from island 1 to island N to buy peanuts.

Since it was dark and scary, every time after Little Gary crossed a bridge, he was frightened and only wanted to cross a bridge that uses time less than or equal to the time for the one he had just crossed.

Please find the minimum time needed for Little Gary to travel from island 1 to island N.

INPUT

The first line of input contains two integers N and M. Each of the following M lines contains three integers A_i, B_i and T_i, indicating that there is a bidirectional bridge between islands A_i and B_i with traveling time T_i. It is guaranteed that it is always possible to travel from island 1 to island N, and $A_i \neq B_i$.

OUTPUT

The first and only line of the output consists of one integer which is the shortest possible time needed to travel from island 1 to island N.

SAMPLE INPUT

```
5 7
1 2 8
1 3 4
2 3 7
3 5 6
2 5 10
2 4 4
4 5 5
```

SAMPLE OUTPUT

```
21
```

CONSTRAINTS

In all of the test cases,

- $1 \leq A_i, B_i \leq N$
- $1 \leq T_i \leq 10,000,000$
- $2 \leq N \leq 100$
- $1 \leq M \leq 2,000$

In 50% of the test cases,

- $1 \leq T_i \leq 50$
- $2 \leq N \leq 10$
- $1 \leq M \leq 50$

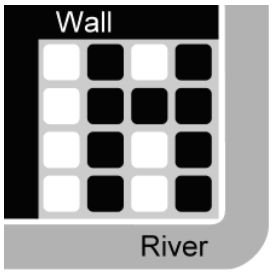
Tiles

PROBLEM

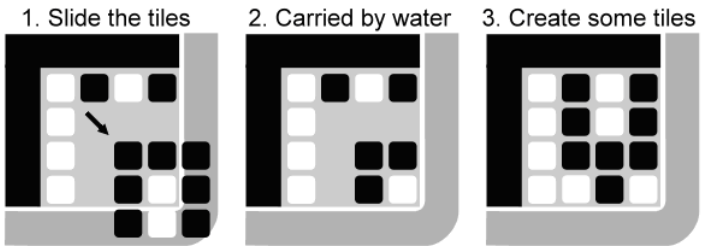
Harpy's Kept On Igloo!!! In order to rescue his friend, the brave penguin Tux is on a mission to rescue Harpy from the igloo full of traps.

The first difficulty Tux encounters is the tiles on the entrance of the igloo. The tiles are made of ice, some of them are white and the others are black (caused by oil spill). They are arranged in an $N \times N$ square. Two adjacent sides of the $N \times N$ square are the walls of the igloo, while the other two sides are parts of a river.

The tiles originally form a pattern which allows the gate to be opened. However, some evil creatures have changed the colour of some tiles. Fortunately, Tux knows what the correct pattern is. In order to open the gate, Tux has to slide and create some ice tiles so as to form the correct pattern.



In the figure, the upper and left sides are the wall, while the lower and right sides is the river. In each shift, Tux can select an integer $M (1 \leq M \leq N)$, then slide an $M \times M$ square of tiles by one unit downward and one unit rightward. Tux cannot push against the wall, so he can only push the tiles to the river. Therefore the $M \times M$ square has to be touching the two sides of the $N \times N$ square which are parts of a river. Tiles beyond the border of the $N \times N$ square are carried away by water. Then Tux creates some "fake" tiles (can be black or white) to put into the spaces on the floor. The following figure shows the case when $M = 3$.



In order to form the correct pattern, Tux can create any number of white and black tiles, and can perform any number of shifts. However, Tux is worried about the pollution problem of the ocean. Pure ice makes the ocean cleaner, while impure ice makes the ocean more polluted. In order to measure the impact of sliding the ice tiles, Tux introduces a value called the "cleanliness value", which is 0 before any shifting is done. When a real tile (tiles not created by Tux) is carried away by water, if it is white, the cleanliness value is increased by 1; if it is black, the cleanliness

value is decreased by 1. Note that fake tiles created by Tux will not change the cleanliness value, as they are not made of real ice. Apart from rescuing his unfortunate friend, Tux would like to maximize the cleanliness value. Given the current pattern and the correct pattern, write a program to find the maximal cleanliness value after the correct pattern is formed.

INPUT

The first line of input contains an integer N. The next N lines describe the pattern on the entrance of the igloo before shifting, in the same orientation as the figures in the problem description. Each of the N lines contains N characters, indicating the tiles on a row. A "`0`" indicates a white tile, while a "`1`" indicates a black tile. Then the next N lines describe the correct pattern, in the same format.

OUTPUT

The first and only line of the output consists of one integer which is the maximal cleanliness value after the correct pattern is formed.

SAMPLE INPUT

```
4
1011
1100
1110
1101
1010
1101
1110
1110
```

SAMPLE OUTPUT

```
-2
```

EXPLANATIONS

The steps are shown below: (*Italic* characters represent fake tiles)

Step 0	Step 1 ($M = 2$)	Step 2 ($M = 4$)
1011 1100 1110 1101	1011 1011 1100 1100 1110 *1101* 1101 *1111*	1011 *1010* 1100 *1101* *1101* *1110* *1111* *1110*
Cleanliness value $= 0$	Cleanliness value $= 1$	Cleanliness value $= -2$

CONSTRAINTS

In all of the test cases,

- $1 \leq N \leq 1,000$

In 50% of the test cases,

- $1 \leq N \leq 50$

Stamp

PROBLEM

In a special stamp store, stamps may have different prices. Customers do not need to pay for every stamp they choose. The amount that a customer has to pay is computed as follows:

First the customer can choose a rectangular container of arbitrary width and height so that the customer can bring the stamps back home easily. Let the width and height of the container chosen be W and H. The container is divided into H rows and W columns. For each space only one stamp can be placed. The value of a space is equal to the price of the stamp placed in the space. If no stamp is in the space, the value of the space is 0. The *maxvalue* of a row or a column is the greatest value among the spaces in the row or column. The amount that the customer needs to pay is the sum of the *maxvalue*s of all rows and columns.

Now John has to buy N stamps. Given the price of the stamps, help John to minimize the amount that he needs to pay.

INPUT

The first line contains a positive integer N, the number of stamps that John has to buy. The following line contains N positive integers, a_i, representing the price of the i^{th} stamp. The N positive integers are sorted in non-decreasing order, i.e. $a_1 \leq a_2 \leq \cdots \leq a_N$.

OUTPUT

The first and the only line of the output contains an integer, the minimum amount that John needs to pay.

SAMPLE INPUT

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

SAMPLE OUTPUT

```
18
```

EXPLANATIONS

One of the best ways to place the stamps is shown below (The numbers in the table are the prices of the stamps. A zero indicates an empty space):

1	0
5	4
2	3

The amount John needs to pay is $1 + 5 + 3 + 5 + 4 = 18$.

CONSTRAINTS

In all of the test cases,

- $1 \leq N \leq 100,000$
- $1 \leq a_i \leq 10,000$

In 50% of the test cases,

- $1 \leq N \leq 10$