Task: ELE
Elections

Day 2. Source file `ele.*` *21-04-2008*

Available memory: 32 MB. Maximum running time: 1 s.

The citizens of Byteland have recently been voting in the parliamentary elections. Now, when the results have been published, the parties have to decide on a coalition to form the government.

Each party received a certain number of seats in the parliament. The coalition must be a subset of the parties such that together they have strictly more than half of all the seats in the parliament. It is desirable for the coalition to have as many seats as possible, to ensure they can still pass their proposed laws even if a few of their members are absent from a parliament session.

A coalition is called *redundant* if one of its parties can be removed with the remaining ones still having more than half of the seats in the parliament. Of course, such a removable party would effectively have no power — the other members of the coalition would be able to force the laws regardless of its opinion.

Task

Write a program that:

- reads the election results from the standard input,
- finds a non-redundant coalition that has the maximal possible number of seats in the parliament,
- writes the description of this coalition to the standard output.

Input

The first line of the standard input contains one integer n ($1 \le n \le 300$) — the number of parties that participated in the elections. The parties are numbered from 1 to n.

The second line contains n nonnegative integers $a_1, \ldots, a_n$, separated by single spaces, where a_i is the number of seats received by the i-th party. You may assume that the total number of seats in the parliament will be positive and lower or equal to 100 000.

Additionally, in test cases worth 40% of points, the number of parties will not exceed 20.

Output

The first line of the standard output should contain one integer k — the number of parties in a non-redundant coalition which has the maximal number of seats.

The second line should contain k distinct integers separated by single spaces — the numbers of parties that form the coalition.

If there are several non-redundant coalitions with the maximal number of seats, you may output any of them. The member parties can be listed in any order.

Example

For the input data:

```
4
1 3 2 4
```

the correct result is:

```
2
2 4
```

Task: GRI
Grid

Day 2. Source file `gri.*` *21–04–2008*

Available memory: 32 MB. Maximum running time: 8 s.

The map of Byteland is drawn on a grid of size $n \times m$ (n is the vertical dimension, m is the horizontal dimension). The horizontal lines marking the division are called *parallels*, and are numbered from 0 to n, while the vertical lines of the division are called *meridians*, and are numbered from 0 to m (see figure on the next page).

Weather forecasting is a serious issue in Byteland. For each unit square of the grid a certain amount of computation time is required to prepare the forecast. Due to terrain conditions and other factors this time may vary from square to square. Until very recently the forecasting system was processing the unit squares one after another, so it took as long as the sum of all the unit times to prepare the complete forecast.

You have been asked to design a new system, running on a multiprocessor computer. To share the computations among processors, the area of Byteland should be divided by r parallels and s meridians into $(r+1)(s+1)$ smaller rectangles. Each processor will cover one rectangle of this division and will process the squares of this rectangle one after another. This way the computation time for such rectangle will be the sum of all computation times of the unit squares contained in this rectangle. The computation time of the complete forecast will be the maximum among computation times of the individual processors.

Your task is to find the minimal possible computation time for some choice of r parallels and s meridians.

Task

Write a program, that:

- reads the dimensions of the map of Byteland, the required number of parallels and meridians and the unit computation times from the standard input,
- finds the minimal time required to compute the complete forecast,
- writes the obtained value to the standard output.

Input

The first line of the input contains four integers n, m, r and s, separated by single spaces ($1 \leq r < n \leq 18$, $1 \leq s < m \leq 18$). The following n lines contain the computation times of the unit squares. The j-th number in the $(i+1)$-st line is $c_{i,j}$ — the time required to prepare the weather forecast for the unit square located between the $(i-1)$-st and i-th parallel and between the $(j-1)$-st and j-th meridian ($1 \leq i \leq n$, $1 \leq j \leq m$, $0 \leq c_{i,j} \leq 2\,000\,000$).

Additionally, in test cases worth 40% of points, n and m will not exceed 10.

Output

Your program should write exactly one line. It should contain one integer — the optimal computation time.

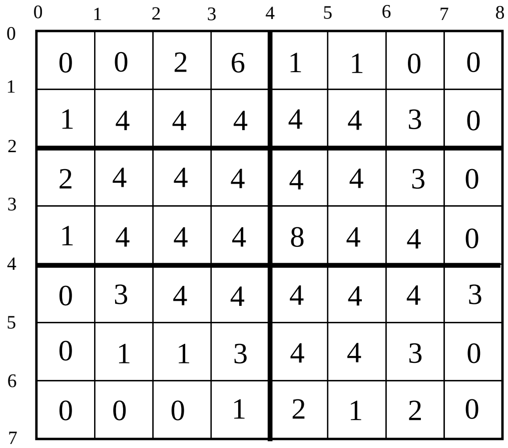
Example

For the input data:

```
7 8 2 1
0 0 2 6 1 1 0 0
1 4 4 4 4 4 3 0
2 4 4 4 4 4 3 0
1 4 4 4 8 4 4 0
0 3 4 4 4 4 4 3
0 1 1 3 4 4 3 0
0 0 0 1 2 1 2 0
```

the correct result is:

```
31
```



0	0	2	6	1	1	0	0
1	4	4	4	4	4	3	0
2	4	4	4	4	4	3	0
1	4	4	4	8	4	4	0
0	3	4	4	4	4	4	3
0	1	1	3	4	4	3	0
0	0	0	1	2	1	2	0

The 2-nd and 4-th parallel and the 4-th meridian divide the country into 6 rectangles with computation times 21, 13, 27, 27, 17, 31. The computation time of the complete forecast is 31.

Task: GLO
Gloves

Day 2. Source file `glo.*` *21–04–2008*

Available memory: 32 MB. Maximum running time: 4 s.

In the dark basement of chemistry professor Acidrain's house there are two drawers with gloves — one with left hand and other with right hand gloves. In each of them there are gloves of n different colours. Professor knows how many gloves of each colour there are in each drawer (the number of gloves of the same colour may differ in both drawers). He is also sure that it is possible to find a pair of gloves of the same colour.

Professor's experiment may be successful only if he uses gloves of the same colour (it does not matter which one), so before every experiment he goes to the basement and takes gloves from the drawers hoping that there will be at least one pair of the same colour. It is so dark in the basement that there is no possibility to recognize colour of any glove without going out of the basement. Professor hates going to the basement more than once (in case there was no pair of gloves of the same colour), as well as bringing unnecessarily large amounts of gloves to the laboratory.

Task

Write a program that:

- reads the number of colours and the number of gloves in each colour in each drawer from the standard input,
- calculates the smallest total number of gloves which must be taken to be sure that among them it is possible to find at least one pair of gloves of the same colour (it is necessary to specify the exact number of gloves to be taken from each drawer),
- writes the result to the standard output.

Input

The first line of the standard input contains one positive integer n ($1 \le n \le 20$) describing the number of distinct colours. The second line of input contains n non-negative integers $0 \le a_1, a_2, \ldots a_n \le 10^8$, where a_i corresponds to the number of gloves of colour number i in the drawer with left hand gloves. Finally, the third line of input contains n non-negative integers $0 \le b_1, b_2, \ldots, b_n \le 10^8$, where b_i corresponds to the number of gloves of colour number i in the drawer with right hand gloves.

Additionally, in test cases worth 40% of points, $n \le 4$ and $a_i, b_i \le 10$.

Output

The first line of the standard output should contain a single integer — the number of gloves which must be taken from the drawer with left hand gloves. The second line of output should contain a single integer — the number of gloves which must be taken from the drawer with right hand gloves. The sum of these two numbers should be as small as possible. If there are several correct results, your program should output any of them.

Example

For the input data:

```
4
0 7 1 6
1 5 0 6
```

the correct result is:

```
2
8
```