REPEATS

A string *s* is called an *(k,l)-repeat* if s is obtained by concatenating $k \geq 1$ times some seed string *t* with length $l \geq 1$. For example, the string

```
s = abaabaabaaba
```

is a (4,3)-repeat with

```
t = aba
```

as its seed string. That is, the seed string *t* is 3 characters long, and the whole string *s* is obtained by repeating *t* 4 times.

Write a program for the following task: Your program is given a long string *u* consisting of characters 'a' and/or 'b' as input. Your program must find some *(k,l)*-repeat that occurs as substring within *u* with *k* as large as possible. For example, the input string

```
u = babbabaabaabaabab
```

contains the underlined (4,3)-repeat s starting at position 5. Since *u* contains no other contiguous substring with more than 4 repeats, your program must output this underlined substring.

Input

In the first line of the input file `repeats.in` one integer - length of the input string *n* ($1 \leq n \leq 50000$) is given.

The next *n* file lines contain the input string, one character (either 'a' or 'b') per line, in order.

Output

The output file `repeats.out` must consist of three integers, each on its own line. They report the *(k, l)*-repeat your program found as follows:

1. The first line consists of the repeat count *k* that is maximized.
2. The second line consists of the length *l* of the seed string that is repeated *k* times.
3. The third and final line consists of the position *p* ($1 \leq p \leq n$) at which the *(k, l)*-repeat starts.

If for given test data there are different solutions with the same *k*, your program must report any one of them.

Example *(corresponds to string u given in task description)*

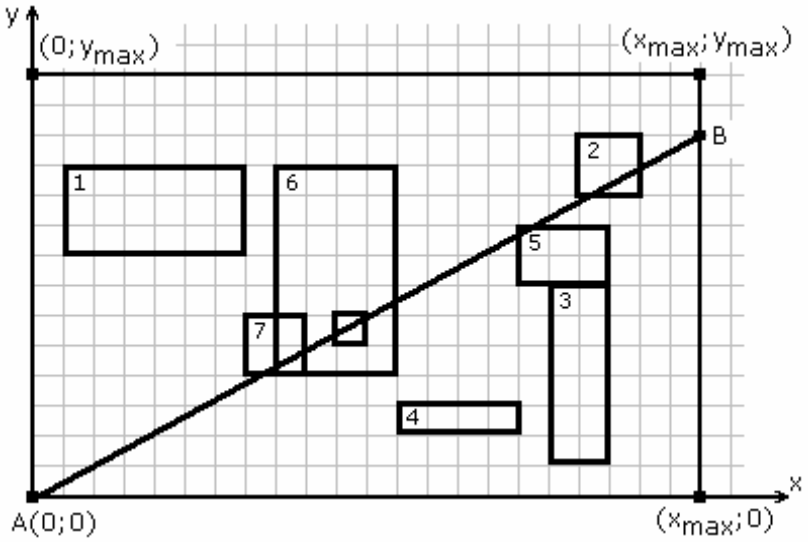
repeats.in	repeats.out
17 b a b b a b a a b a a b a a b a b	4 3 5 since a (4, 3)-repeat is found starting at the 5th character of the input string (which is line 6 of the input file).

There are given N rectangles on the plane. Rectangle sides are parallel to coordinate axis. These rectangles may overlap, coincide or be drawn inside one another. Their vertices have non-negative integer coordinates and *x* coordinates do not exceed x_{max} and *y* coordinates do not exceed y_{max}.

A segment is started in the point A(0, 0) and ended in point B. The coordinates of the point B (the other end of the segment) satisfy the following conditions:

- The coordinates of B are integer numbers;
- The point B belongs either to the segment $[(0, y_{max}), (x_{max}, y_{max})]$ or to the segment $[(x_{max}, 0), (x_{max}, y_{max})]$;

The segment AB might cross rectangles (we assume that crossing takes place even if only one rectangle vertex is crossed).



Example with 8 rectangles. The segment AB crosses 5 of them.

Task

Write a program to find a point B for which the segment AB crosses as many rectangles as possible.

Input

The first line of the input file `rect.in` contains three integers: x_{max}, y_{max} ($0 < x_{max}, y_{max} \le 10^9$) and N ($1 \le N \le 10000$). Each of the following N lines contains four integers: coordinates of the bottom left corner x_{bl} and y_{bl} and coordinates of the top right corner x_{tr} and y_{tr}. Neighbouring numbers are separated by single space character.

Output

On the first and only line of the output file `rect.out` three integer numbers should be written. First – the maximum number of crossed rectangles followed by x and y coordinates of point B. Neighbouring numbers must be separated by single space character.

If there are several solutions, find any one of them.

Example (corresponds to the drawing)

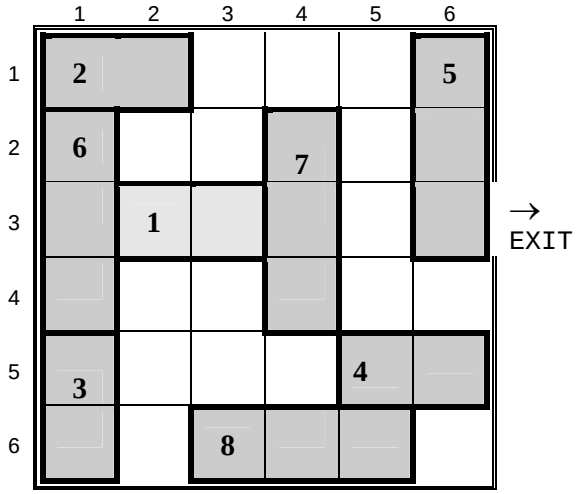
rect.in	rect.out
22 14 8 1 8 7 11 18 10 20 12 17 1 19 7 12 2 16 3 16 7 19 9 8 4 12 11 7 4 9 6 10 5 11 6	5 22 12 *Remark:* Another possible solution is 5 22 11

CAR PARK

DAY 2 (23-APR-2004)
ENGLISH VERSION

The youth hostel at which the BOI2004 is being held has a parking lot consisting of a grid of 6 by 6 squares. Lot's rows are numbered starting from 1 to 6 consecutively from top to down, columns are numbered in the same way from left to right. There is only one exit from the lot at the right side of third row.

On that lot, there are N parked cars. Your car is among those cars, but unfortunately, there is no easy way out because your car is blocked by the other cars.You and your friends can move the cars forwards and backwards however, since the gearbox of all cars is in Neutral. You may not steer or turn, neither your own car nor any of the other cars.



It is your task to determine the minimum number of steps necessary to get your car of size 2x1 squares off the parking lot. One step means moving one car one square. None of the other cars may be moved off the parking lot.

There are only two types of cars. One type is 2x1 squares in size, where as the other type occupies 3x1 squares. Cars may only be moved along the longer one of their two axes.

In the given example N =8 and your car is labeled with the number 1.

Below is the minimum sequence of length 18 to exit the parking lot with your car:
4←←←, 2→,6↑,3↑,8←←,5↓↓↓,7↓↓,1→→→→→.

Input

The first line of the input file `carpark.in` contains the number of cars N($1 \le N \le 16$).

Each of the subsequent N lines contains the description of the car that is labeled with the number i. Each line consists of four integers specifying the length l_i, the orientation o_i, and the start (upper left) coordinates x_i (number of column) and y_i (number of row). Neighbouring numbers are separated by single space character. o_i=1means that the car is parked horizontally. Otherwise, it is parked vertically.

The following limitations apply: $l_i \in \{2,3\}$, $o_i \in \{0,1\}$, $1 \le x_i, y_i \le 6$.

Your car is described in the first line following the single line consisting of the number N (i.e. the second line of the input file). Your car has to exit the parking lot using the only possible exit.

Output

The output file `carpark.out` should be comprised of only one single integer, representing the minimum number of steps necessary to exit the parking lot in your car.

If it is impossible to exit the parking lot, print `-1`.

Example (corresponds to the given example)

carpark.in	carpark.out
8 2 1 2 3 2 1 1 1 2 0 1 5 2 1 5 5 3 0 6 1 3 0 1 2 3 0 4 2 3 1 3 6	18