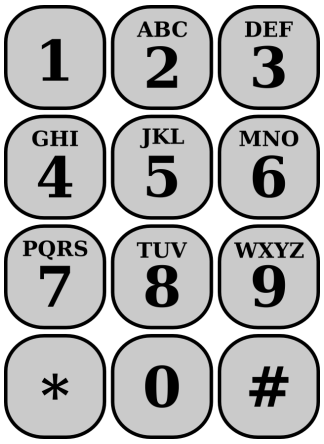
Problem 1

People sometimes give their phone number as a character string representing the 8-digit phone number. For example, if my phone number was 8368 8737, I could present it as VENTURES. That could be easier to remember. It can be seen that there are many strings representing the 8368 8737 number but most of them are not English words. The picture below presents the mapping between letters and digits used in standard phones:



Please write a program which reads from the standard input one 8-digit phone number and writes to the standard output a list of all strings representing the given number. That list should be in lexicographical order and consecutive strings should be separated by a newline character. To transform the input number you should use the mapping on the picture above. Because 1 and 0 keys do not have any letters assigned, only numbers from 2 to 9 should be transformed. In case when a badly formatted input is passed to the program, that is not a 8-digit phone number, the program should write word ERROR to the standard output.

Sample input

```
11001501
```

Sample output

```
11001J01
11001K01
11001L01
```

Problem 2

You have found an offer in a newspaper saying that someone has *n* square plots of land to sell. All plots have the same area, each of them adjoins its neighbors and their placement resembles a straight line. Each of the plots has the same selling price. The plots' placement and numbering are presented on the picture below:

The offer also gives the profitability of each of the plots. Some of them are profitable, some are losing money and some bring no profit.

The offer has interested you very much and you are going to buy a set of plots which maximizes your profit. The profitability of a set of plots is defined as the sum of the profitabilities of each plot in the set. There is one additional constraint. The seller does not want to split his estate excessively, so while he will sell as many plots as the buyer wants, the plots must form a contiguous set.

Please write a program which computes, for a given number of plots and their profitabilities, a list of plots you should buy to fulfill the conditions above. Ideally, your program should be very fast to place an order before others do.

Input

The first line of the input is the number of plots for sale - *n*. *n* is a natural number ranging from 1 to 2,000,000. The second line contains *n* integers ranging from -1000 to 1000, specifying the profitabilities of consecutive plots. You can assume that the input is always correct, it conforms to the conditions above and has at least one plot with a profitability greater than 0.

Output

Your program should output 3 numbers separated by a single space: index of the first plot in the set; index of the last plot in the set; and the profitability of the selected set. When there is more than one correct answer, your program should write the one that contains the smallest number of plots. If there is still more than one possible answer, your program can write any of them.

Sample Input

```
9
-5 0 7 -6 4 3 -5 0 2
```

Sample output

```
3 6 8
```