Q. 1 Write a C function to find the kth occurrence of an even integer in a sequence of non-negative integers, and then call your function from main.

Your function should be according to the following declaration:

int find_even(int k);

Input
You are given the input in two lines:

The first line contains a positive integer k.
In the second line, you will be given a sequence of numbers.

You have to find the kth occurrence of n in the sequence below.

The second line consists of a sequence of non-negative integers, terminated with a -1. The -1 is not part of the sequence.

Output
If there are k even numbers in the sequence, then output the kth occurrence of even in the sequence. Otherwise, output a -1.

Sample Input
2
1 1 3 2 3 4 1 -1

Sample Output
4
Sol.
#include <stdio.h>

void find_even ( int k )
{
    int even_count=0;
    int curr;

    scanf("%d",&curr);
    while ( curr != -1 ){
        if ( curr % 2 == 0 ){
            even_count = even_count+1;
            if ( even_count == k ){
                printf ( "%d\n", curr );
                return ;
            }
        }
        scanf ( "%d", &curr );
    }
    printf ( "-1\n" );
    return ;
}

int main()
{
    int k;
    int k;
    scanf("%d",&k);
    find_even(k);
    return 0;
}
Q.2 When you keep track of stock prices, or your weight during a diet programme, the daily prices or weights fluctuate a lot. One way to identify the general trend is to keep track of the average over the last 3 days, for example. This technique often smooths out the fluctuations, and makes the trend clearer. This technique is called "moving average".

In this question, you have to output the "moving average" of a sequence of non-negative numbers. The moving average is the sequence of averages of the last 3 entries. For the first 2 numbers, no average is output.

For example, if the sequence of numbers is

\[ a_1, a_2, a_3, a_4, a_5 \]

and \( k=3 \), then the 3-moving average is

\[ \frac{a_3+a_2+a_1}{3}, \frac{a_4+a_3+a_2}{3}, \frac{a_5+a_4+a_3}{3}. \]

Input
-----

The input is a sequence of non-negative floating point numbers, terminated by a -1. The -1 is not part of the sequence. There will be at least 3 numbers in the sequence.

Output
-----

You have to output the moving average of the sequence. The output should be printed correct to one digit after the decimal.

Sample Input 1
---------------
70.8 70.9 71.2 70.7 70.2 -1

Sample Output 1
---------------
71.0 70.9 70.7
Sol.

#include <stdio.h>
int main()
{
    float first;
    float second;
    float third;
    float curr;

    scanf ( "%f", &first );
    scanf ( "%f", &second );
    scanf ( "%f", &third );
    printf ( "%.1f ", (first+second+third)/3);

    scanf ( "%f", &curr );

    while ( curr != -1 ){
        first = second;
        second = third;
        third = curr;

        printf ( "%.1f ", (first+second+third)/3);

        scanf ( "%f", &curr );
    }

    return 0;
}
Q.3 A line of English text will be given, where words are separated by one of the following symbols:

', ' \t', ',', ' and ';'

Each word may be separated from the next and the previous by one or more of the following symbols. You have to count the number of words in the sentence.

Note that to read the input, you have to read until the EOF symbol is read, as in the following example.

```c
int main()
{
    int c;
    
    c = getchar();
    while ( c != EOF ) {
        c = getchar();
    }
    return 0;
}
```

Note: it is possible to solve this question without arrays, so the maximum length of the line is not important.

Input
-----
A line of English text with words separated from one another by one or more occurrences of the symbols

', ' \t', ',', ' and ';'

Output
------
The number of words in the line.

Sample Input
-------------
This is a sentence, it has words separated by spaces and fullstops.

Sample Output
-------------
12
Sol.
#include <stdio.h>

void consume_space()
{
    int c;
    c = getchar();
    while(1){
        switch(c){
            case ' ':
            case '	':
            case '.':
            case ',':
            case ':':
            case ';':
                c = getchar();
                continue;
            default:
                ungetc(c,stdin);
                return;
        }
    }
}

int is_space(int c)
{
    switch(c){
        case ' ':
        case '	':
        case '.':
        case ',':
        case ':':
        case ';':
            return 1;
        default:
            return 0;
    }
}

int main()
{
    int c;
    int count=0;

    /* consume leading spaces */
    c = getchar();
    if ( is_space ( c ) ){
        consume_space();
    }
}
c = getchar();
while ( c != EOF ){
    if ( is_space(c) ){
        count++;
        consume_space();
    }
    c = getchar();
}
printf("%d\n", count);
return 0;