Assignment 3

Introduction to programming in C

Question 1

In this question, you have to output the "two moving average" of a sequence of non-negative numbers.

The two moving average is the sequence of averages of the last 2 entries. For the first number, no average is output.

For example, if the sequence of numbers is a_1, a_2, a_3, a_4, a_5, the 2-moving average is

$$\frac{(a_1 + a_2)}{2}, \frac{(a_2 + a_3)}{2}, \frac{(a_3 + a_4)}{2}, \frac{(a_4 + a_5)}{2}$$

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.

Solution

```c
#include <stdio.h>

int main() {
  float first;
  float second;
  float curr;

  scanf("%f", &first);
  scanf("%f", &second);
  printf("%.1f ", (first + second) / 2);
  scanf("%f", &curr);

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

```
16
17      printf("%.1f ", (first + second) / 2);
18      scanf("%f", &curr);
19    }
20    return 0;
21  }
```

Question 2

Prime Checking

Complete the function int *is_prime*(int n) to check if a positive number n is prime or not.

The function returns 1 if n is prime, and 0 otherwise.

The function will be used in a program (code given) that prints the prime numbers in a given sequence.

Input

The first line of input is a positive integer N. The next line contains N positive integers k_i for i=1 to N.

Output

The elements in the input list which are primes, in the original order.

Solution

```c
#include <stdio.h>

int is_prime(int n){
    if (n == 0 || n == 1)
    return 0;

    for (int i = 2; i < n ; i++) {
        if (n % i == 0) {
            return 0;
        }
    }

  return 1;
}

int main() {

  int n,num;
  scanf("%d", &n);

  for(int i=0;i<n;i++){
      scanf("%d",&num);
      if(is_prime(num)){
          printf("%d ",num);
      }
  }

  return 0;
}
```

Question 3

Write a C function to find the kth occurrence of an odd integer in a sequence of non-negative integers.

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 terminated with a -1. You have to find the kth occurrence of an odd integer in the sequence.

Note: The -1 is not part of the sequence.

Output

If there are k odd numbers in the sequence, then output the k^{th} occurrence of an odd number in the sequence, if present. If there are less than k odd numbers in the sequence, output -1.

Solution

```c
#include <stdio.h>

int main() {
  int k;
  scanf("%d", &k);

  int odd_count = 0;
  int curr, flag = 0;

  scanf("%d", &curr);
  while (curr != -1) {
    if (curr % 2 == 1) {
      odd_count = odd_count + 1;
      if (odd_count == k) {
        printf("%d", curr);
        flag = 1;
      }
    }
    scanf("%d", &curr);
  }

  if(flag == 0)
    printf("-1");

  return 0;
}
```