

# Assignment 3

## Introduction to programming in C

### Question 1

Complete the function `int find_factorial(int k)` to find the factorial of the positive number  $k$ .

The factorial of a positive integer  $k$ , denoted by  $k!$ , is the product of all positive integers less than or equal to  $k$ .

$$k! = k \times (k - 1) \times \dots \times 1.$$

#### 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

For each  $k$  given as input, print  $k!$ .

### Solution

```
1 #include <stdio.h>
2
3 int find_factorial(int k){
4     int p = 1;
5     for (int i=1;i<=k;i++){
6         p*=i;
7     }
8     return p;
9 }
10
11 int main(){
12     int n,k;
13     scanf("%d",&n);
14
15     for (int i=0;i<n;i++){
16         scanf("%d",&k);
17         printf("%d ", find_factorial(k));
18     }
19
20     return 0;
21 }
```

## Question 2

Write a C function to find the  $k^{th}$  occurrence of an odd 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\_odd(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 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

```
1 #include <stdio.h>
2
3 void find_odd(int k) {
4     int odd_count = 0;
5     int curr;
6
7     scanf("%d", & curr);
8     while (curr != -1) {
9         if (curr % 2 == 1) {
10             odd_count = odd_count + 1;
11             if (odd_count == k) {
12                 printf("%d", curr);
13                 return;
14             }
15         }
16         scanf("%d", & curr);
17     }
18     printf("-1");
19     return;
20 }
21
22 int main() {
23     int k;
24     scanf("%d", & k);
25     find_odd(k);
26     return 0;
27 }
```

### Question 3

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

```
1 #include <stdio.h>
2
3 int main() {
4     float first;
5     float second;
6     float curr;
7
8     scanf("%f", & first);
9     scanf("%f", & second);
10    printf("%.1f ", (first + second) / 2);
11    scanf("%f", & curr);
12
13    while (curr != -1) {
14        first = second;
15        second = curr;
16
17        printf("%.1f ", (first + second) / 2);
18        scanf("%f", & curr);
19    }
20    return 0;
21 }
```