**C Questions**

**4\*25 = 100 Marks**

1. Given an array, of size n, reverse it.

Example: If array, arr = [1,2,3,4,5] , after reversing it, the array should be, arr = [5,4,3,2,1] .

**Input Format**

The first line contains an integer, n, denoting the size of the array. The next line contains n space-separated integers denoting the elements of the array.

**Constraints**

1≤ n≤ 1000   
1 ≤ arrᵢ ≤ 1000, where arrᵢ is the ᵢth element of the array.

**Output Format**

The output is handled by the code given in the editor, which would print the array.

**Sample Input 0**

6

16 13 7 2 1 12

**Sample Output 0**

12 1 2 7 13 16

**Explanation 0**

Given array, arr = [16, 13, 7, 2, 1, 12]. After reversing the array, arr = [12, 1, 2, 7, 13, 16]

**Sample Input 1**

7

1 13 15 20 12 13 2

**Sample Output 1**

2 13 12 20 15 13 1

**Sample Input 2**

8

15 5 16 15 17 11 5 11

**Sample Output 2**

11 5 11 17 15 16 5 15

1. An array is a container object that holds a fixed number of values of a single type. To create an array in C, we can do int arr[n];. Here, arr, is a variable array which holds up to 10 integers. The above array is a static array that has memory allocated at compile time. A dynamic array can be created in C, using the malloc function and the memory is allocated on the heap at runtime. To create an integer array, arr of size n, int \*arr = (int\*)malloc(n \* sizeof(int)), where  points to the base address of the array.

In this challenge, you have to create an array of size n dynamically, input the elements of the array, sum them and print the sum of the elements in a new line.

**Input Format**

The first line contains an integer, n.   
The next line contains n space-separated integers.

**Constraints**

 1 ≤ n ≤ 1000  
 1 ≤ aᵢ ≤ 1000

**Output Format**

Print in a single line the sum of the integers in the array.

**Sample Input 0**

6

16 13 7 2 1 12

**Sample Output 0**

51

**Sample Input 1**

7

1 13 15 20 12 13 2

**Sample Output 1**

76

1. Given a sentence, s, print each word of the sentence in a new line.

**Input Format**

The first and only line contains a sentence, s.

**Constraints**

**1**≤Len(s) ≤1000

**Output Format**

Print each word of the sentence in a new line.

**Sample Input 0**

This is C

**Sample Output 0**

This

is

C

**Explanation 0**

In the given string, there are three words ["This", "is", "C"]. We have to print each of these words in a new line.

**Sample Input 1**

Learning C is fun

**Sample Output 1**

Learning

C

is

fun

**Sample Input 2**

How is that

**Sample Output 2**

How

is

that

1. Given N integers, compute their average correct to three decimal places.

**Input Format**   
The first line contains an integer, N.   
 N lines follow, each containing a single integer.

**Output Format**   
Display the average of the N integers, rounded off to three decimal places.

**Input Constraints**   
 1≤N≤500

-10000≤x≤10000 (x refers to elements of the list of integers for which the average is to be computed)

**Sample Input**

4

1

2

9

8

**Sample Output**

5.000

**Explanation**   
The '4' in the first line indicates that there are four integers whose average is to be computed. The average = (1 + 2 + 9 + 8)/4 = 20/4 = 5.000 (correct to three decimal places) Please include the zeroes even if they are redundant (e.g. 0.000 instead of 0).