■ **Prepare** › C › Arrays and Strings › Array Reversal   Exit Full Screen View ⤢

Change Theme   Language: C   ↺   ⋮

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

$1 \leq n \leq 1000$

$1 \leq arr_i \leq 1000$, where arr_i 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

```c
1    #include <stdio.h>
2    #include <stdlib.h>
3
4    void arrayReverse(int *arr, int n) {
5        int i, temp;
6        for(i = 0; i < n / 2; i++){
7            temp = arr[i];
8            arr[i] = arr[n - i - 1];
9            arr[n - i - 1] = temp;
10       }
11
12   }
13
14   int main()
15   {
16       int num, *arr, i;
17       scanf("%d", &num);
18       arr = (int*) malloc(num * sizeof
     (int));
19       for(i = 0; i < num; i++) {
20           scanf("%d", arr + i);
21       }
22
23
24       /* Write the logic to reverse the
     array. */
25       arrayReverse(arr, num);
26
27       for(i = 0; i < num; i++)
28           printf("%d ", *(arr + i));
29       return 0;
30   }
```

--INSERT--   Line: 6 Col: 19