Arrays - DS | HackerRank

hackerrank.com/challenges/arrays-ds/problem?isFullScreen=true

Verify it's you | New Chrome available

HackerRank | Prepare | Data Structures | Arrays | Arrays - DS | Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions

An array is a data structure that stores elements of the same type in a contiguous block of memory. In an array, A, of size N, each memory location has some unique index, i (where $0 \leq i < N$), that can be referenced as $A[i]$ or A_i.

Your task is to reverse an array of integers.

Note: If you've already solved our C++ domain's Arrays Introduction challenge, you may want to skip this.

Example

$A = [1, 2, 3]$

Return $[3, 2, 1]$.

Function Description

Complete the function *reverseArray* with the following parameter(s):

- *int* $A[n]$: the array to reverse

Returns

- *int*$[n]$: the reversed array

Input Format

The first line contains an integer, N, the number of integers in A.

The second line contains N space-separated integers that make up A.

Constraints

- $1 \leq N \leq 10^3$
- $1 \leq A[i] \leq 10^4$, where $A[i]$ is the i^{th} integer in A

Copy Download

Change Theme | Language C++11

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n;
    cin >> n;

    vector<int> arr(n);

    // Input elements
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    // Reverse using vector (manual swap)
    for (int i = 0, j = n - 1; i < j; i++, j--) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }

    // Output reversed array
    for (int x : arr) {
        cout << x << " ";
```

Line: 30 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Search | NZ - WI In 7 hours | ENG IN | 23:29 21-11-2025

Arrays - DS | HackerRank

hackerrank.com/challenges/arrays-ds/problem?isFullScreen=true

Verify it's you | New Chrome available

HackerRank | Prepare | Data Structures | Arrays | Arrays - DS | Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions

An array is a data structure that stores elements of the same type in a contiguous block of memory. In an array, A, of size N, each memory location has some unique index, i (where $0 \leq i < N$), that can be referenced as $A[i]$ or A_i.

Your task is to reverse an array of integers.

Note: If you've already solved our C++ domain's Arrays Introduction challenge, you may want to skip this.

Example

$A = [1, 2, 3]$

Return $[3, 2, 1]$.

Function Description

Complete the function *reverseArray* with the following parameter(s):

- $int\ A[n]$: the array to reverse

Returns

- $int[n]$: the reversed array

Input Format

The first line contains an integer, N, the number of integers in A.

The second line contains N space-separated integers that make up A.

Constraints

- $1 \leq N \leq 10^3$
- $1 \leq A[i] \leq 10^4$, where $A[i]$ is the i^{th} integer in A

Copy Download

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin) Download

```
4
1 4 3 2
```

Your Output (stdout)

```
2 3 4 1
```

Expected Output Download

```
2 3 4 1
```

Search | NZ - WI In 7 hours | ENG IN | 23:29 21-11-2025

2D Array - DS | HackerRank

hackerrank.com/challenges/2d-array/problem?isFullScreen=true

Verify it's you | New Chrome available

HackerRank | Prepare | Data Structures | Arrays | 2D Array - DS | Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions

Given a 6×6 2D array, arr, an hourglass is a subset of values with indices falling in the following pattern:

```
a b c
  d
e f g
```

There are 16 hourglasses in a 6×6 array. The $hourglass\ sum$ is the sum of the values in an hourglass. Calculate the hourglass sum for every hourglass in arr, then print the $maximum$ hourglass sum.

Example

$arr =$

```
-9 -9 -9  1 1 1
 0 -9  0  4 3 2
-9 -9 -9  1 2 3
 0  0  8  6 6 0
 0  0  0 -2 0 0
 0  0  1  2 4 0
```

The 16 hourglass sums are:

```
-63, -34, -9, 12,
-10,   0, 28, 23,
-27, -11, -2, 10,
  9,  17, 25, 18
```

Change Theme | Language C++11

```cpp
#include <iostream>
#include <vector>
#include <limits>
using namespace std;

int main() {
    vector<vector<int>> arr(6, vector<int>(6));

    // Input
    for(int i = 0; i < 6; i++)
        for(int j = 0; j < 6; j++)
            cin >> arr[i][j];

    int maxSum = numeric_limits<int>::min();

    // Compute hourglass sum
    for(int i = 0; i < 4; i++) {
        for(int j = 0; j < 4; j++) {

            int sum = arr[i][j] + arr[i][j+1] + arr[i][j+2]
                      + arr[i+1][j+1]
                      + arr[i+2][j] + arr[i+2][j+1] + arr[i+2][j+2];

            maxSum = max(maxSum, sum);
        }
```

Line: 31 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Search | Finance headline India reported 6... | ENG IN | 23:29 21-11-2025

2D Array - DS | HackerRank

hackerrank.com/challenges/2d-array/problem?isFullScreen=true

Verify it's you | New Chrome available

HackerRank | Prepare | Data Structures | Arrays | 2D Array - DS | Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions

Given a 6×6 2D array, arr, an hourglass is a subset of values with indices falling in the following pattern:

```
a b c
  d
e f g
```

There are 16 hourglasses in a 6×6 array. The $hourglass\ sum$ is the sum of the values in an hourglass. Calculate the hourglass sum for every hourglass in arr, then print the $maximum$ hourglass sum.

Example

$arr =$

```
-9 -9 -9  1 1 1
 0 -9  0  4 3 2
-9 -9 -9  1 2 3
 0  0  8  6 6 0
 0  0  0 -2 0 0
 0  0  1  2 4 0
```

The 16 hourglass sums are:

```
-63, -34, -9, 12,
-10,   0, 28, 23,
-27, -11, -2, 10,
  9,  17, 25, 18
```

```
14    int maxSum = numeric_limits<int>::min();
15
16    // Compute hourglass sum
17    for(int i = 0; i < 4; i++) {
18        for(int j = 0; j < 4; j++) {
19
20            int sum = arr[i][j] + arr[i][j+1] + arr[i][j+2]
21                      + arr[i+1][j+1]
22                      + arr[i+2][j] + arr[i+2][j+1] + arr[i+2][j+2];
23
24            maxSum = max(maxSum, sum);
25        }
```

Line: 31 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

- Sample Test case 0
- Sample Test case 1
- Sample Test case 2

Input (stdin) | Download

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 2 4 4 0
```

Search | 25°C Mostly cloudy | ENG IN | 23:24 21-11-2025