

## Experiment 1(a)

**Student Name:**DEEPANSHU

**UID:**22BCS16443

**Branch:** CSE

**Section/Group:**602-A

**Semester:** 5<sup>th</sup>

**Date of Performance:**18-07-24

**Subject Name:** Advanced Programming Lab-1

**Subject Code:** 22CSP-314

**1. Title:** Array Reversal

**2. Aim:** Given an array, of size, reverse it. Example: If array, arr= { 1,2,3,4,5}, after reversing it, the array should be, arr= {5,4,3,2,1}.

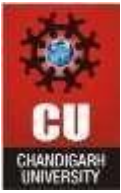
**3. Objective:** The objective of the problem "Given an array of size  $n$ , reverse it" is to write a function that takes an array of a given size as input and returns a new array where the order of the elements is reversed.

**4. Implementation/Code:**

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int num, *arr, i;
    scanf("%d", &num);
    arr = (int*) malloc(num * sizeof(int));
    for(i = 0; i < num; i++) {
        scanf("%d", arr + i);
    }

    int temp;
    for(i = 0; i < num / 2; i++) {
        temp = arr[i];
        arr[i] = arr[num - i - 1];
```



```
        arr[num - i - 1] = temp;
    }
    for(i = 0; i < num; i++)
        printf("%d ", *(arr + i));
    return 0;
}
```

## 5. Output:

The screenshot displays a coding platform interface. On the left, a list of test cases is shown, all marked as passed (green checkmarks). The main area on the right shows the 'Compiler Message' as 'Success'. Below this, the 'Input (stdin)' is displayed as two lines: '6' and '16 13 7 2 1 12'. The 'Expected Output' is shown as a single line: '12 1 2 7 13 16'. There are 'Download' links for both the input and output sections.

Test Case	Status
Test case 0	Passed
Test case 1	Passed
Test case 2	Passed
Test case 3	Passed
Test case 4	Passed
Test case 5	Passed
Test case 6	Passed

Compiler Message: Success

Input (stdin):

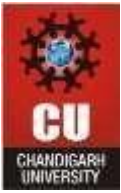
```
6
16 13 7 2 1 12
```

Expected Output:

```
12 1 2 7 13 16
```

6. Time Complexity:  $O(n)$

7. Space Complexity:  $O(n)$



## Experiment 1(B)

**Student Name:** DEEPANSHU

**UID:**22BCS16443

**Branch:** CSE

**Section/Group:**602-A

**Semester:** 5<sup>th</sup>

**Date of Performance:**18-07-24

**Subject Name:** Advanced Programming Lab-1

**Subject Code:** 22CSP-314

**1. Title:** Sum of Array

**2. Aim:** Given an array of integers, find the sum of its elements. For example, if the array, so return.

**3. Objective:**

- Read Input: Take an integer input representing the number of elements, followed by a line of space-separated integers.
- Calculate Sum: Compute the sum of these integers as they are read.

Output Result: Print the calculated sum.

**4. Implementation/Code:**

```
Change Theme Language C++11
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n, sum = 0, value;
6
7      cin >> n;
8      for (int i = 0; i < n; i++) {
9          cin >> value;
10         sum += value;
11     }
12
13     cout << sum << endl;
14     return 0;
15 }
16
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## 5. Output:

The screenshot displays a dark-themed interface for a coding problem. On the left, a sidebar lists three test cases, all marked as passed with green checkmarks. The first test case is highlighted. The main area shows the compiler message 'Success', the input (stdin) for the first test case, and the expected output.

Test case 0

Compiler Message

Success

Input (stdin)

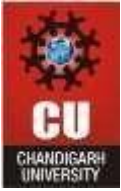
1	6
2	1 2 3 4 10 11

Expected Output

1	31
---	----

**6. Time Complexity:**  $O(n)$

**7. Space Complexity:**  $O(1)$



## Experiment 1(D)

**Student Name:**DEEPANSHU

**UID:**22BCS16443

**Branch:** CSE

**Section/Group:**602-A

**Semester:** 5<sup>th</sup>

**Date of Performance:**18-07-24

**Subject Name:** Advanced Programming Lab-1

**Subject Code:** 22CSP-314

### 1. Title: Diagonal Difference

### 2. Aim:

Given a square matrix, calculate the absolute difference between the sums of its diagonals.

For example, the square matrix is shown below:

```
1 2 3
4 5 6
9 8 9
```

The left-to-right diagonal =. The right to left diagonal =. Their absolute difference is.

### 3. Objective:

The objective of this program is to compare the ratings of two challenges created by Alice and The objective of this code is to compute the absolute difference between the sums of the diagonals of a square matrix. Given a square matrix of size  $n \times n$  the program performs the following tasks:

**Input Handling:** It reads the size of the matrix  $n \times n$  and the matrix elements from the standard input.

**Diagonal Sum Calculation:** It calculates the sum of the elements in the left-to-right diagonal and the sum of the elements in the right-to-left diagonal.

**Absolute Difference:** It computes the absolute difference between these two diagonal sums.

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## 4.Implementation/Code:

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      int n;
7      cin >> n;
8      int matrix[n][n];
9
10     for (int i = 0; i < n; i++) {
11         for (int j = 0; j < n; j++) {
12             cin >> matrix[i][j];
13         }
14     }
15     int LRdiagSum = 0;
16     int RLdiagSum = 0;
17     for (int i = 0; i < n; i++) {
18         LRdiagSum += matrix[i][i];
19         RLdiagSum += matrix[i][n - 1 - i];
20     }
21     int difference = abs(LRdiagSum - RLdiagSum);
22     cout << difference;
23
24     return 0;
25 }
26
```

## 5. Output:

The screenshot displays a coding platform interface with a sidebar on the left containing a list of test cases, each marked with a green checkmark and a lock icon. The main area on the right is divided into three sections: 'Compiler Message' showing 'Success', 'Input (stdin)' showing a 4x3 matrix, and 'Expected Output' showing the value 15. A 'Download' link is present next to the input and output sections.

Test case 0 ✓

Test case 1 ✓

Test case 2 ✓

Test case 3 ✓

Test case 4 ✓

Test case 5 ✓

Test case 6 ✓

Compiler Message

Success

Input (stdin) [Download](#)

1	3
2	11 2 4
3	4 5 6
4	10 8 -12

Expected Output [Download](#)

1	15
---	----

# **DEPARTMENT OF**

# **COMPUTER SCIENCE & ENGINEERING**

6. **Time Complexity:**  $O(n^2)$

7. **Space Complexity:**  $O(n^2)$