Experiment No.1
Implement Reverse Array
Date of Performance:
Date of Submission:
Name:Gaurav Kishor Patil
Div / Roll No.:2/54 Batch:C

Experiment No. 1: Implementation of Reverse Array

Aim: Implementation of Reverse Array and finding average of students marks using array

Objective:

1) Implement a function to reverse the order of elements in an array.
2) Develop a program to calculate the average of student marks stored in an array.
3) Ensure code readability and modularity for both array reversal and average calculation implementations.

Theory:

Reversing an array involves changing the order of elements from their original sequence to the opposite. This can be achieved by swapping the elements at the beginning and end of the array, then progressively moving towards the centre until the entire array is reversed..

Calculating the average of student marks stored in an array involves summing up all the individual marks and dividing the total by the number of students.

Writing readable and modular code is essential for maintainability and understanding. This can be achieved by following coding conventions, using meaningful variable and function names, and breaking down the implementation into smaller functions with specific purposes. Additionally, adding comments at critical points can provide insights into the logic and purpose of each part of the code. Proper indentation and formatting also contribute to code readability.

Algorithm:

1. Initialize two pointers, 'start' and 'end', pointing to the beginning and end of the array, respectively.
2. While 'start' is less than 'end':
   a. Swap the elements at indices 'start' and 'end'.
   b. Increment 'start' and decrement 'end'.

3. Array is now reversed.

4. Initialize variables: 'sum' to store the sum of marks, and 'count' to store the number of students.

5. Iterate through each mark in the array:

   a. Add the mark to the 'sum'.

   b. Increment 'count' by 1.

6. Calculate the average:

   a. Average = sum / count.

7. Average of student marks is calculated.

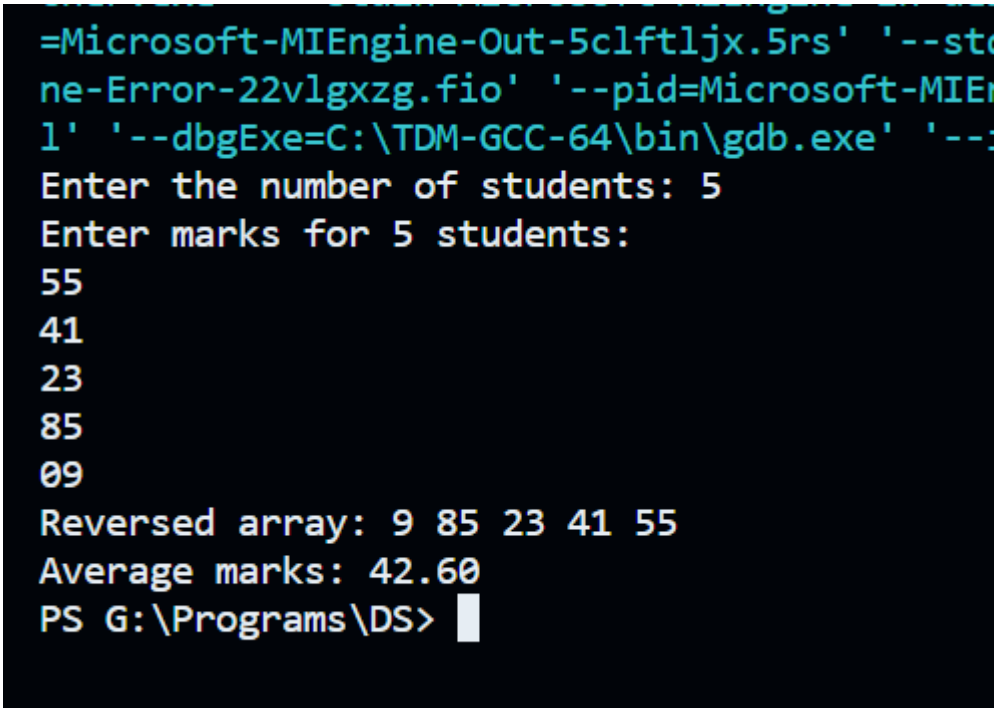
Code:

```
#include <stdio.h>

#include<stdlib.h>

int i=0 ;

void reverseArray(int arr[], int size) {

int temp;


for (i = 0; i < size / 2; i++) {

temp = arr[i];

arr[i] = arr[size - 1 - i];

arr[size - 1 - i] = temp;

}


}
```

```
void main() {

int numStudents;

printf("Enter the number of students: ");

scanf("%d", &numStudents);

int  marks[numStudents];

printf("Enter marks for %d students:\n", numStudents);
for ( i = 0; i < numStudents; i++) {

scanf("%d", &marks[i]);
}
int total = 0;
for (i = 0; i < numStudents; i++) {
total += marks[i];
}
double average = (double)total / numStudents;
reverseArray(marks, numStudents);
printf("Reversed array: ");
for (i = 0; i < numStudents; i++) {
```

```
printf("%d ", marks[i]);
}
printf("\n");
printf("Average marks: %.2lf\n", average);
}
```

Output:



Conclusion:

Time Complexity: O(n)

Space Complexity: O(n)

The implementation of a reverse array in C is a fundamental exercise that helps in understanding the concept of array manipulation in C programming. It involves traversing the array from both ends, i.e., from the start and end simultaneously, and swapping the elements at these positions. This process continues until the middle of the array is reached, effectively reversing the array. This operation has a time complexity of O(n), making it efficient for large datasets. The simplicity and efficiency of this algorithm make it a valuable tool in a programmer's arsenal for data manipulation tasks.