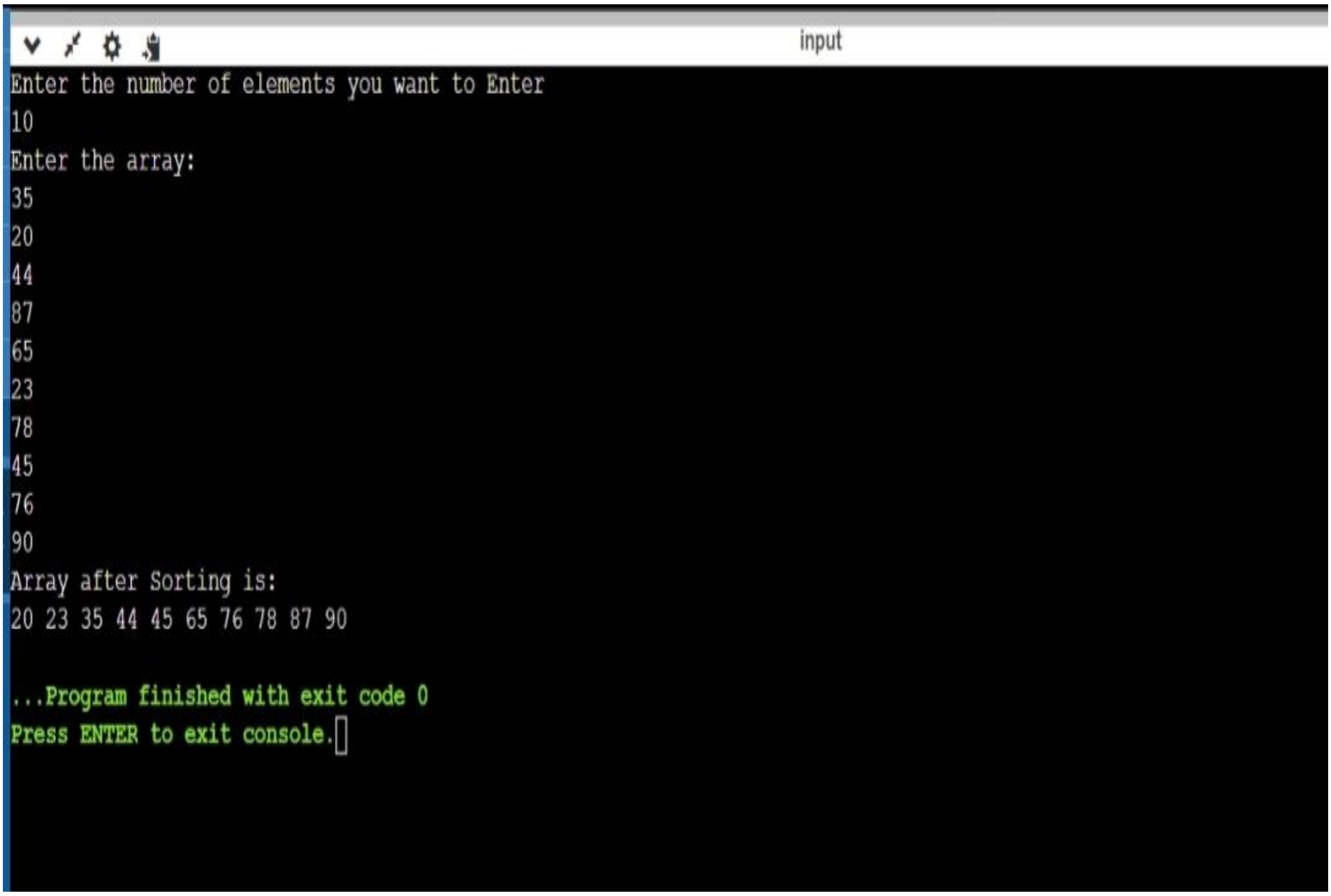
Experiment No. 5: Mini Project: Evaluate performance enhancement of parallel Quicksort Algorithm using Mi// operation is being performed.

```
// C++ program to implement the Quick Sort
// using OMI
#include <bits/stdc++.h>
#include <omp.h>
using namespace std;

// Function to swap two numbers a and b
void swap(int* a, int* b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

// Function to perform the partitioning
// of array arr[]
int partition(int arr[], int start, int end)
{
    // Declaration
    int pivot = arr[end];
    int i = (start - 1);

    // Rearranging the array
    for (int j = start; j <= end - 1; j++) {
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[end]);

    // Returning the respective index
    return (i + 1);
}

// Function to perform QuickSort Algorithm
// using openmp
void quicksort(int arr[], int start, int end)
{
    // Declaration
    int index;

    if (start < end) {
        // Getting the index of pivot
        // by partitioning
        index = partition(arr, start, end);

// Parallel sections
#pragma omp parallel sections
        {
#pragma omp section
            {
                // Evaluating the left half
                quicksort(arr, start, index - 1);
            }
#pragma omp section
            {
                // Evaluating the right half
                quicksort(arr, index + 1, end);
            }
        }
    }
}

// Driver Code
int main()
{
    // Declaration
    int N;

    // Taking input the number of
    // elements we wants
    cout << "Enter the number of elements"
         << " you want to Enter\n";
    cin >> N;

    // Declaration of array
    int arr[N];

    cout << "Enter the array: \n";

    // Taking input that array
    for (int i = 0; i < N; i++) {
        cin >> arr[i];
    }

    // Calling quicksort having parallel
    // code implementation
    quicksort(arr, 0, N - 1);

    // Printing the sorted array
    cout << "Array after Sorting is: \n";

    for (int i = 0; i < N; i++) {
        cout << arr[i] << " ";
    }

    return 0;
}
```

Output:



```
Enter the number of elements you want to Enter
10
Enter the array:
35
20
44
87
65
23
78
45
76
90
Array after Sorting is:
20 23 35 44 45 65 76 78 87 90

...Program finished with exit code 0
Press ENTER to exit console.
```