Name: - Atish Kumar

Roll No: - 120CS0173

Lab Sheet: - 08

Q1. Write a program to implement the Merge sort algorithm with the space optimization for auxiliary space O(n/2)

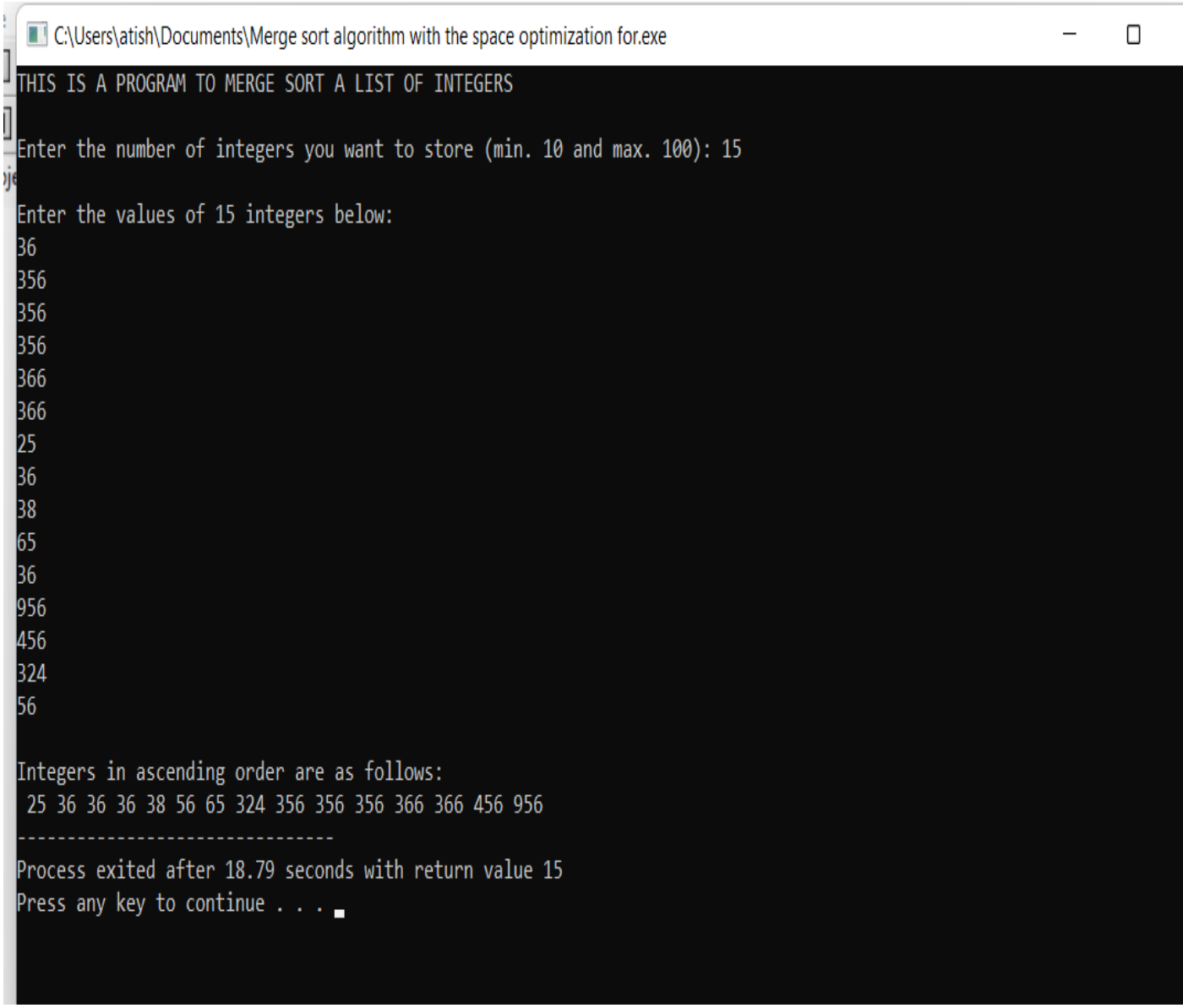
Program:-

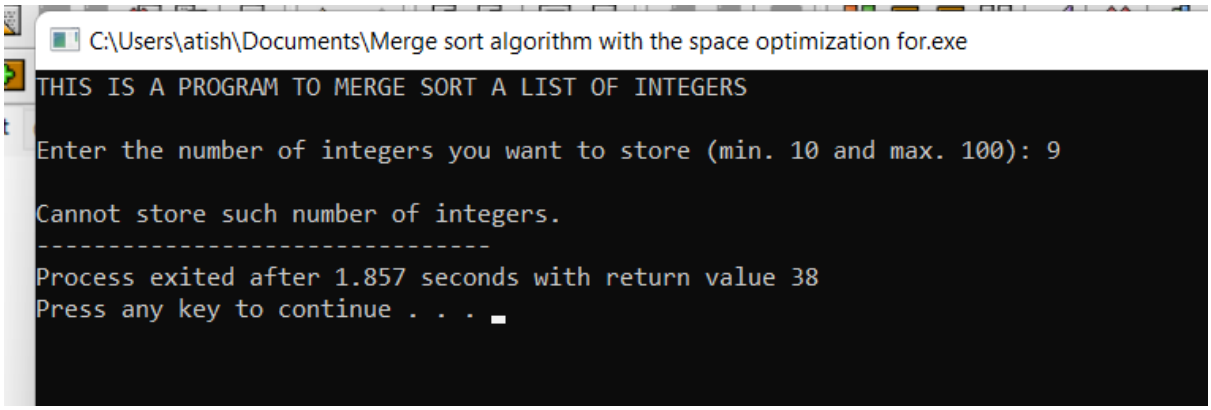
```
#include <stdio.h>

void merge_process(int [], unsigned short, unsigned short, unsigned short);
void merge_sort(int [], unsigned short, unsigned short);

void main()
{
	unsigned short num, i, low=0, high;
	printf("THIS IS A PROGRAM TO MERGE SORT A LIST OF INTEGERS\n\n");
	printf("Enter the number of integers you want to store (min. 10 and max. 100): ");
	scanf("%d", &num);
	if(num<10 || num>100)
	printf("\nCannot store such number of integers.");
	else
	{
		int A[num];
		printf("\nEnter the values of %d integers below:\n", num);
		for(i=0; i<num; i++)
		{
			scanf("%d", &A[i]);
		}
```

```
                high = num-1;
                merge_sort(A, low, high);
                printf("\nIntegers in ascending order are as follows:\n");
                for(i=0; i<num; i++)
                {
                        printf(" %d", A[i]);
                }
        }
}

void merge_process(int a[], unsigned short l, unsigned short m, unsigned short h)
{
        unsigned short i, j, k=l, n = m-l+1;

        //Only one auxilliary array has been taken to reduce the space complexity of
Merge sort from O(n) to O(n/2)
        int aux[n];

        for(i=0; i<n; i++)
        {
                aux[i] = a[l+i];
        }
        i=0;
        j=m+1;
        while(i<n && j<=h)
        {
                if(aux[i]<=a[j])//For descending order, we write '>=' in place of '<='
                a[k++] = aux[i++];
                else
                a[k++] = a[j++];
```

```
        }
        while(i<n)
        {
                a[k++] = aux[i++];
        }
        while(j<=h)
        {
                a[k++] = a[j++];
        }
}

void merge_sort(int arr[], unsigned short low, unsigned short high)
{
        unsigned short mid;
        if(low<high)
        {
                mid = (low+high)/2;
                merge_sort(arr, low, mid);
                merge_sort(arr, mid+1, high);
                merge_process(arr, low, mid, high);
        }
}
```

Output:-


C:\Users\atish\Documents\Merge sort algorithm with the space optimization for.exe
THIS IS A PROGRAM TO MERGE SORT A LIST OF INTEGERS
Enter the number of integers you want to store (min. 10 and max. 100): 15
Enter the values of 15 integers below:
36
356
356
356
366
366
25
36
38
65
36
956
456
324
56
Integers in ascending order are as follows:
25 36 36 36 38 56 65 324 356 356 356 366 366 456 956
Process exited after 18.79 seconds with return value 15
Press any key to continue . . .


C:\Users\atish\Documents\Merge sort algorithm with the space optimization for.exe
THIS IS A PROGRAM TO MERGE SORT A LIST OF INTEGERS
Enter the number of integers you want to store (min. 10 and max. 100): 9
Cannot store such number of integers.
Process exited after 1.857 seconds with return value 38
Press any key to continue . . .