Institute of Computer Technology
B. Tech Computer Science and Engineering
Subject: DS (2CSE302)

PRACTICAL-21

AIM: - Implement selection sort and merge sort.

(This practical is in continuation of previous practical scenario, so append code of below scenario along with previous practical)

1. Swati is working on different sorting methods to sort the data. She wants to prepare sorting calculator which provides the facilities to sort all kind of sorting methods for same data. Kindly refer given scenario for calculator and implement it in C:

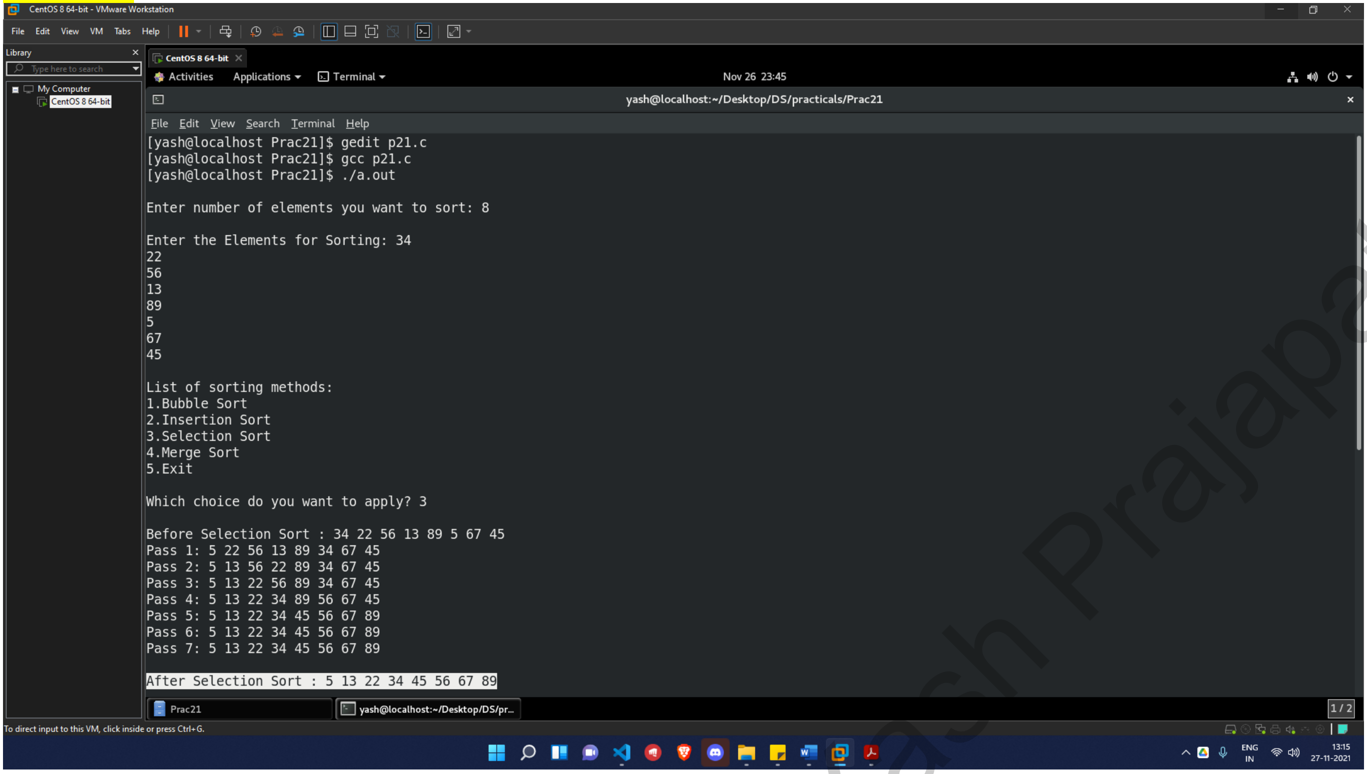
```
How many numbers you want to sort?
8
Enter the Elements for Sorting:
34
22
56
13
89
5
67
45
List of sorting methods:
1. Bubble Sort
2. Insertion Sort
3. Selection Sort
4. Merge Sort
5. Exit
Which choice do you want apply?
3
Pass-1: 5 22 56 13 89 34 67 45
Pass-2: 5 13 56 22 89 34 67 45
Pass-3: 5 13 22 56 89 34 67 45
Pass-4: 5 13 22 34 89 56 67 45
Pass-5: 5 13 22 34 45 56 67 89
Pass-6: 5 13 22 34 45 56 67 89
Pass-7: 5 13 22 34 45 56 67 89
```

SOLUTION

```
#include <stdio.h>
#include <stdlib.h>
int p = 0, q = 0, rm = 0;

void PrintArr(int arr[], int n)
{
	for (int i = 0; i < n; i++)
	{
		printf("%d ", arr[i]);
	}
	printf("\n");
}

void BubbleSort(int arr[], int m)
{
	int i, j, k, temp;
	for (i = 0; i < m - 1; i++)
	{
		for (j = 0; j < m - i - 1; j++)
		{
			if (arr[j] > arr[j + 1])
			{
				temp = arr[j];
				arr[j] = arr[j + 1];
				arr[j + 1] = temp;
			}
		}
		printf("Pass %d: ", i + 1);
		for (k = 0; k < m; k++)
		{
			printf("%d ", arr[k]);
		}
		printf("\n");
	}
}
void InsertionSort(int arr[], int m)
{
	int i, j, k, temp;
	for (i = 1; i < m; i++)
	{
		temp = arr[i];
		j = i - 1;
		while (j >= 0 && arr[j] > temp)
```

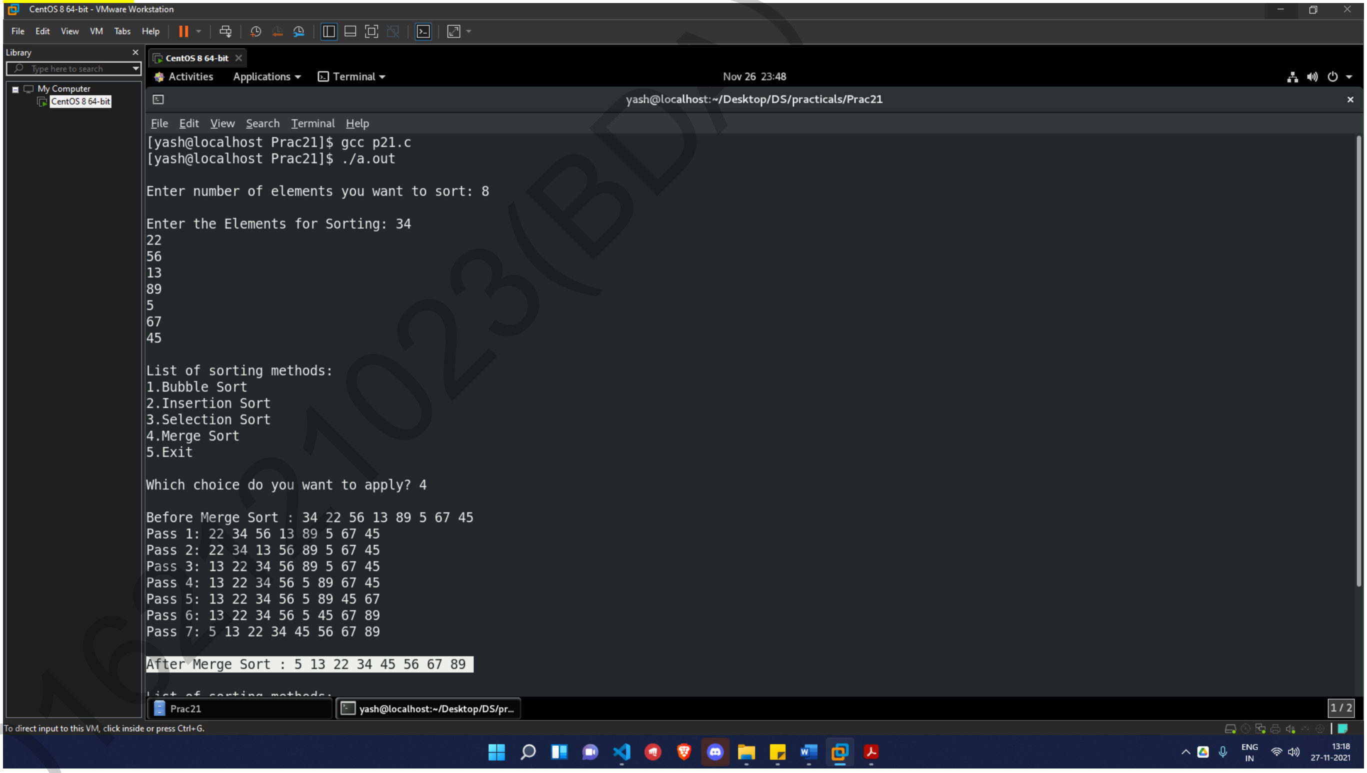
```
        {
            arr[j + 1] = arr[j];

            j -= 1;
        }
        arr[j + 1] = temp;
        printf("Pass %d: ", i);
        for (k = 0; k < m; k++)
        {
            printf("%d ", arr[k]);
        }
        printf("\n");
    }
}
void SelectionSort(int arr[], int m)
{
    int i, j, min, k, temp;
    for (i = 0; i < m - 1; i++)
    {
        min = i;
        for (j = i + 1; j < m; j++)
        {
            if (arr[j] < arr[min])
            {
                min = j;
            }
        }
        temp = arr[i];
        arr[i] = arr[min];
        arr[min] = temp;
        printf("Pass %d: ", i + 1);
        for (k = 0; k < m; k++)
        {
            printf("%d ", arr[k]);
        }
        printf("\n");
    }
}
void merge(int arr[], int l, int mid, int r)
{

    int i, j, k;
    int n1 = mid - l + 1;
    int n2 = r - mid;
    int left[n1], right[n2];
    for (i = 0; i < n1; i++)
    {
        left[i] = arr[l + i];
    }
    for (j = 0; j < n2; j++)
    {
        right[j] = arr[mid + 1 + j];
    }
    i = 0;
    j = 0;
    k = l;
    while (i < n1 && j < n2)
    {
        if (left[i] < right[j])
        {
            arr[k] = left[i];
            i++;
        }
        else
        {
            arr[k] = right[j];
            j++;
        }
        k++;
    }
    while (i < n1)
    {
        arr[k] = left[i];
        i++;
        k++;
    }
    while (j < n2)
    {

        arr[k] = right[j];
        j++;
        k++;
    }
}
void MergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        int mid = (l + r) / 2;
        MergeSort(arr, l, mid);
        MergeSort(arr, mid + 1, r);
        merge(arr, l, mid, r);
        printf("Pass %d: ", ++q);
        for (rm = 0; rm < p; rm++)
        {
            printf("%d ", arr[rm]);
        }
        printf("\n");
    }
}
int main()
{
    int n, i, choice;
    printf("\nEnter number of elements you want to sort: ");
    scanf("%d", &n);
    p = n;
    int yash[n], prajapati[n], y[n], p[n];
    printf("\nEnter the Elements for Sorting: ");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &yash[i]);
        prajapati[i] = yash[i];
        y[i] = yash[i];
        p[i] = yash[i];
    }
there:
    printf("\nList of sorting methods: \n1.Bubble Sort \n2.Insertion Sort \n3.Selection
Sort \n4.Merge Sort \n5.Exit");
    printf("\n\nWhich choice do you want to apply? ");
    scanf("%d", &choice);
    switch (choice)
    {
    case 1:
        printf("\nBefore Bubble Sort : ");
        PrintArr(yash, n);
        BubbleSort(yash, n);
        printf("\nAfter Bubble Sort : ");
        PrintArr(yash, n);
        goto there;
        break;
    case 2:
        printf("\nBefore Insertion Sort : ");
        PrintArr(prajapati, n);
        InsertionSort(prajapati, n);
        printf("\nAfter Insertion Sort : ");
        PrintArr(prajapati, n);
        goto there;
        break;
    case 3:
        printf("\nBefore Selection Sort : ");
        PrintArr(y, n);
        SelectionSort(y, n);
        printf("\nAfter Selection Sort : ");
        PrintArr(y, n);
        goto there;
        break;
    case 4:
        printf("\nBefore Merge Sort : ");
        PrintArr(p, n);
        MergeSort(p, 0, n - 1);
        printf("\nAfter Merge Sort : ");
        PrintArr(p, n);
        goto there;
        break;
    case 5:
        exit(0);
        break;
    default:
        break;
    }
    printf("\n");
    return 0;
}
```

OUTPUT (Selection Sort)


CentOS 8 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
CentOS 8 64-bit
Activities
Applications
Terminal
Nov 26 23:45
yash@localhost:~/Desktop/DS/practicals/Prac21
File Edit View Search Terminal Help
Prac21
yash@localhost:~/Desktop/DS/pr...
1 / 2
To direct input to this VM, click inside or press Ctrl+G.
ENG IN
13:15
27-11-2021

```
[yash@localhost Prac21]$ gedit p21.c
[yash@localhost Prac21]$ gcc p21.c
[yash@localhost Prac21]$ ./a.out

Enter number of elements you want to sort: 8

Enter the Elements for Sorting: 34
22
56
13
89
5
67
45

List of sorting methods:
1.Bubble Sort
2.Insertion Sort
3.Selection Sort
4.Merge Sort
5.Exit

Which choice do you want to apply? 3

Before Selection Sort : 34 22 56 13 89 5 67 45
Pass 1: 5 22 56 13 89 34 67 45
Pass 2: 5 13 56 22 89 34 67 45
Pass 3: 5 13 22 56 89 34 67 45
Pass 4: 5 13 22 34 89 56 67 45
Pass 5: 5 13 22 34 45 56 67 89
Pass 6: 5 13 22 34 45 56 67 89
Pass 7: 5 13 22 34 45 56 67 89

After Selection Sort : 5 13 22 34 45 56 67 89
```

OUTPUT (Merge Sort)


CentOS 8 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
CentOS 8 64-bit
Activities
Applications
Terminal
Nov 26 23:48
yash@localhost:~/Desktop/DS/practicals/Prac21
File Edit View Search Terminal Help
Prac21
yash@localhost:~/Desktop/DS/pr...
1 / 2
To direct input to this VM, click inside or press Ctrl+G.
ENG IN
13:18
27-11-2021

```
[yash@localhost Prac21]$ gcc p21.c
[yash@localhost Prac21]$ ./a.out

Enter number of elements you want to sort: 8

Enter the Elements for Sorting: 34
22
56
13
89
5
67
45

List of sorting methods:
1.Bubble Sort
2.Insertion Sort
3.Selection Sort
4.Merge Sort
5.Exit

Which choice do you want to apply? 4

Before Merge Sort : 34 22 56 13 89 5 67 45
Pass 1: 22 34 56 13 89 5 67 45
Pass 2: 22 34 13 56 89 5 67 45
Pass 3: 13 22 34 56 89 5 67 45
Pass 4: 13 22 34 56 5 89 67 45
Pass 5: 13 22 34 56 5 89 45 67
Pass 6: 13 22 34 56 5 45 67 89
Pass 7: 5 13 22 34 45 56 67 89

After Merge Sort : 5 13 22 34 45 56 67 89

List of sorting methods:
```