Write the user defined functions to implement the following sorting algorithm.

i. Bubble sort
ii. Selection sort
iii. Insertion sort
iv. Merge sort
v. Quick sort

Programs:

(i) Bubble Sort:

```
#include <stdio.h>

void bubbleSort(int arr[], int n)
{
    int swap = 1, i = 0, j, temp;
    while (swap == 1 && i < n)
    {
        swap = 0;
        j = 0;
        while (j < n - i - 1)
        {
            if (arr[j] > arr[j + 1])
            {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
                swap = 1;
            }
            j += 1;
        }
        i += 1;
    }
}

int main()
{
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter array element (%d): ", i + 1);
        scanf("%d", &arr[i]);
    }
```

```
    printf("Unsorted array: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");

    bubbleSort(arr, n);

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

    return 0;
}
```

(ii) Selection Sort:

```
#include <stdio.h>

void selectionSort(int arr[], int n)
{
    int i, j, temp, loc;
    for (i = 0; i < n - 1; i++)
    {
        loc = i;
        for (j = i + 1; j < n; j++)
        {
            if (arr[j] < arr[loc])
            {
                loc = j;
            }
        }
        if (loc != i)
        {
            temp = arr[i];
            arr[i] = arr[loc];
            arr[loc] = temp;
        }
    }
}

int main()
{
```

```
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter array element (%d): ", i + 1);
        scanf("%d", &arr[i]);
    }

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

    selectionSort(arr, n);

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

    return 0;
}
```

(iii) Insertion Sort:

```
#include <stdio.h>

void insertionSort(int arr[], int n)
{
    int i, j, temp;
    for (i = 1; i < n; i++)
    {
        temp = arr[i];
        j=i-1;
        while (j>=0 && arr[j]>temp)
        {
            arr[j+1] = arr[j];
            j--;
        }
        arr[j+1] = temp;
    }
```

```
}

int main()
{
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter array element (%d): ", i + 1);
        scanf("%d", &arr[i]);
    }

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

    insertionSort(arr, n);

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

    return 0;
}
```

(iv) Merge Sort:

```
#include <stdio.h>

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

void merge(int A[], int mid, int low, int high)
```

```
{
    int i, j, k, B[100];
    i = low;
    j = mid + 1;
    k = low;

    while (i <= mid && j <= high)
    {
        if (A[i] < A[j])
        {
            B[k] = A[i];
            i++;
        }
        else
        {
            B[k] = A[j];
            j++;
        }
        k++;
    }
    while (i <= mid)
    {
        B[k] = A[i];
        k++;
        i++;
    }
    while (j <= high)
    {
        B[k] = A[j];
        k++;
        j++;
    }
    for (int i = low; i <= high; i++)
    {
        A[i] = B[i];
    }
}

void mergeSort(int A[], int low, int high)
{
    int mid;
    if (low < high)
    {
        mid = (low + high) / 2;
        mergeSort(A, low, mid);
        mergeSort(A, mid + 1, high);
```

```
            merge(A, mid, low, high);
        }
    }

    int main()
    {
        int n;
        printf("Enter the size of the array: ");
        scanf("%d", &n);
        int A[n];
        for (int i = 0; i < n; i++)
        {
            printf("Enter array element (%d): ", i + 1);
            scanf("%d", &A[i]);
        }

        printf("Unsorted array: ");
        printArray(A, n);
        mergeSort(A, 0, n - 1);
        printf("Sorted array: ");
        printArray(A, n);

        return 0;
    }
```

(v) Quick Sort:

```
    #include <stdio.h>

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

    int partition(int A[], int low, int high)
    {
        int i = low + 1, j = high, pivot = A[low], temp;

        do
        {
            while (A[i] <= pivot)
            {
                i++;
```

```
        }
        while (A[j] > pivot)
        {
            j--;
        }
        if (i < j)
        {
            temp = A[i];
            A[i] = A[j];
            A[j] = temp;
        }
    } while (i < j);

    // Swap A[low] and A[j]
    temp = A[low];
    A[low] = A[j];
    A[j] = temp;

    return j;
}

void quickSort(int A[], int low, int high)
{
    int partitionIndex;
// index of pivot after partition

    if (low < high)
    {
        partitionIndex = partition(A, low, high);
        quickSort(A, low, partitionIndex -
 1);  // sort left subarray
        quickSort(A, partitionIndex + 1, high);
     // sort right subarray
    }
}

int main()
{
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int A[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter array element (%d): ", i + 1);
        scanf("%d", &A[i]);
```

```
    }

    printf("Unsorted array: ");
    printArray(A, n);
    quickSort(A, 0, n - 1);
    printf("Sorted array: ");
    printArray(A, n);

    return 0;
}
```