

BUBBLE SORT:

```
#include <stdio.h>
```

```
void swap(int* arr, int i, int j)
```

```
{  
    int temp = arr[i];  
    arr[i] = arr[j];  
    arr[j] = temp;  
}
```

```
void bubbleSort(int arr[], int n)
```

```
{  
    int i, j;  
    for (i = 0; i < n - 1; i++)  
  
        // Last i elements are already  
        // in place  
        for (j = 0; j < n - i - 1; j++)  
            if (arr[j] > arr[j + 1])  
                swap(arr, j, j + 1);  
}
```

```
// Function to print an array
```

```
void printArray(int arr[], int size)
```

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

```
// Driver code
int main()
{
    int arr[] = { 5, 1, 4, 2, 8 };
    int N = sizeof(arr) / sizeof(arr[0]);
    bubbleSort(arr, N);
    printf("Sorted array: ");
    printArray(arr, N);
    return 0;
}
```

SELECTION SORT:

```
#include <stdio.h>
```

```
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}
```

```
void selectionSort(int arr[], int n)
{
    int i, j, min_idx;

    // One by one move boundary of unsorted subarray
    for (i = 0; i < n-1; i++)
    {
        // Find the minimum element in unsorted array
        min_idx = i;
```

```
    for (j = i+1; j < n; j++)
        if (arr[j] < arr[min_idx])
            min_idx = j;

    // Swap the found minimum element with the first element
    if(min_idx != i)
        swap(&arr[min_idx], &arr[i]);
}
}
```

```
/* Function to print an array */
void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
```

```
// Driver program to test above functions
int main()
{
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr)/sizeof(arr[0]);
    selectionSort(arr, n);
    printf("Sorted array: \n");
    printArray(arr, n);
    return 0;
}
```