

C 9\_1\_bubble\_sort.c > ...

```
1  /* Given an array, arr[]. Sort the array using bubble sort algorithm.
2
3  Examples :--
4  Input: arr[] = [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
5  Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int *bubblesort(int *arr, int n){
11     for(int i = 0; i < n; i++){
12         for(int j = 0; j < n-1; j++){
13             if(arr[j] > arr[j+1]){
14                 int temp = arr[j+1];
15                 arr[j+1] = arr[j];
16                 arr[j] = temp;
17             }
18         }
19     }
20     return arr;
21 }
22
23 int main(){
24     while(1){
25         printf("\nEnter your Choice :--\n1 - Sort input array by Bubble Sort\n2 - Exit\nChoice : ");
26         int choice;
27         scanf("%d",&choice);
28         switch(choice){
29             case 1:
30                 {
31                     int n;
32                     printf("Enter the no. of elements in the array : ");
33                     scanf("%d",&n);
34                     int arr[n];
35                     printf("Enter %d numbers : ",n);
36                     for(int _ = 0; _ < n; _++){
37                         scanf("%d",&arr[_]);
38                     }
39                     int *res = bubblesort(arr,n);
40                     printf("Sorted Array : ");
41                     for(int _ = 0; _ < n; _++){
42                         printf("%d ",res[_]);
43                     }

```

```
44         printf("\n");
45     }
46     break;
47     case 2: exit(0);
48         break;
49     default : printf("INVALID CHOICE - TRY AGAIN.");
50 }
51 }
52 }
```

TERMINAL COMMENTS

Enter your Choice :--

1 - Sort input array by Bubble Sort

2 - Exit

Choice : 1

Enter the no. of elements in the array : 10

Enter 10 numbers : 10 9 8 7 6 5 4 3 2 1

Sorted Array : 1 2 3 4 5 6 7 8 9 10

Enter your Choice :--

1 - Sort input array by Bubble Sort

2 - Exit

Choice : 1

Enter the no. of elements in the array : 10

Enter 10 numbers : 8 5 9 2 4 2 1 0 0 4

Sorted Array : 0 0 1 2 2 4 4 5 8 9

Enter your Choice :--

1 - Sort input array by Bubble Sort

2 - Exit

Choice : 2

PS C:\Users\shuvr\OneDrive\Documents\CODING\College C codes\DSA-ASS-9>

C 9\_2\_merge\_sort.c > merge(int [], int, int, int)

```
1  /* Given an array arr[], its starting position l and its ending position r.
2  Sort the array using the merge sort algorithm.
3  Examples :--
4  Input: arr[] = [4, 1, 3, 9, 7]
5  Output: [1, 3, 4, 7, 9] */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 void merge(int arr[], int l, int m, int r){
11     int i, j, k;
12     int n1 = m - l + 1;
13     int n2 = r - m;
14     int L[n1], R[n2];
15     for(i = 0; i < n1; i++){
16         L[i] = arr[l + i];
17     }
18     for(j = 0; j < n2; j++){
19         R[j] = arr[m + 1 + j];
20     }
21     i = 0;
22     j = 0;
23     k = l;
24     while(i < n1 && j < n2){
25         if(L[i] <= R[j]){
26             arr[k] = L[i];
27             i++;
28         }
29         else{
30             arr[k] = R[j];
31             j++;
32         }
33         k++;
34     }
```

```
35     while(i < n1){
36         arr[k] = L[i];
37         i++;
38         k++;
39     }
40     while(j < n2){
41         arr[k] = R[j];
42         j++;
43         k++;
44     }
45 }
46
47 void mergeSort(int arr[], int l, int r){
48     if(l < r){
49         int m = l + (r - l)/2;
50         mergeSort(arr, l, m);
51         mergeSort(arr, m + 1, r);
52         merge(arr, l, m, r);
53     }
54 }
55
56 void printArray(int arr[], int size){
57     for(int i = 0; i < size; i++){
58         printf("%d ", arr[i]);
59     }
60     printf("\n");
61 }
62
```

```
62
63 int main(){
64     while(1){
65         int ch;
66         printf("\nEnter your choice :--\n1 - Merge-Sort an array\n2- Exit\nChoice : ");
67         scanf("%d",&ch);
68         switch(ch){
69             case 1:
70                 {
71                     int n;
72                     printf("Enter length of array : ");
73                     scanf("%d",&n);
74                     int arr[n];
75                     printf("Enter an space separated array of %d numbers : ",n);
76                     for(int _ = 0; _ < n; _++){
77                         scanf("%d",&arr[_]);
78                     }
79                     mergeSort(arr, 0, n - 1);
80                     printf("Sorted array : ");
81                     printArray(arr, n);
82                     break;
83                 }
84             case 2: exit(0);
85                     break;
86             default: printf("INVALID INPUT - TRY AGAIN.\n");
87         }
88     }
89     return 0;
90 }
91
```

Enter your choice :--

1 - Merge-Sort an array

2- Exit

Choice : 1

Enter length of array : 10

Enter an space separated array of 10 numbers : 10 9 8 7 6 5 4 3 2 1

Sorted array : 1 2 3 4 5 6 7 8 9 10

Enter your choice :--

1 - Merge-Sort an array

2- Exit

Choice : 1

Enter length of array : 10

Enter an space separated array of 10 numbers : 7 9 3 1 7 0 2 4 1 0

Sorted array : 0 0 1 1 2 3 4 7 7 9

Enter your choice :--

1 - Merge-Sort an array

2- Exit

Choice : 2

PS C:\Users\shuvr\OneDrive\Documents\CODING\College C codes\DSA-ASS-9>