

30. BUBBLE SORT CODE

The screenshot shows a C++ IDE with the following code in `bubble sort.cpp`:

```
1 #include <stdio.h>
2 int main()
3 {
4     int array[100], n, c, d, swap;
5     printf("Enter number of elements\n");
6     scanf("%d", &n);
7     printf("Enter %d integers\n", n);
8     for (c = 0; c < n; c++)
9         scanf("%d", &array[c]);
10    for (c = 0; c < n - 1; c++)
11    {
12        for (d = 0; d < n - c - 1; d++)
13        {
14            if (array[d] > array[d+1])
15            {
16                swap = array[d];
17                array[d] = array[d+1];
18                array[d+1] = swap;
19            }
20        }
21    }
22    printf("Sorted list in ascending order:\n");
23    for (c = 0; c < n; c++)
24        printf("%d\n", array[c]);
25    return 0;
26 }
```

The execution output is as follows:

```
Enter number of elements
5
Enter 5 integers
4 2 6 1 9
Sorted list in ascending order:
1
2
4
6
9

-----
Process exited after 10.85 seconds with return value
0
Press any key to continue . . .
```

Compilation results:

```
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\VARMA\Documents\bubble sort.exe
- Output Size: 128.7705078125 KiB
- Compilation Time: 0.14s
```

31. C PROGRAM TO IMPLEMENT INSERTION SORT

The screenshot shows a C++ IDE with the following code in `INSERTION SORT.cpp`:

```
1 #include <stdio.h>
2 int main()
3 {
4     int arr[100], n, i, j, key;
5     printf("Enter the number of elements: ");
6     scanf("%d", &n);
7     printf("Enter the elements: ");
8     for (i = 0; i < n; i++) {
9         scanf("%d", &arr[i]);
10    }
11    printf("Original array: ");
12    for (i = 0; i < n; i++) {
13        printf("%d ", arr[i]);
14    }
15    for (i = 1; i < n; i++) {
16        key = arr[i];
17        j = i - 1;
18        while (j >= 0 && arr[j] > key) {
19            arr[j + 1] = arr[j];
20            j = j - 1;
21        }
22        arr[j + 1] = key;
23    }
24    printf("\nSorted array: ");
25    for (i = 0; i < n; i++) {
26        printf("%d ", arr[i]);
27    }
28    return 0;
29 }
```

The execution output is as follows:

```
Enter the number of elements: 5
Enter the elements: 55 46 77 42 68
Original array: 55 46 77 42 68
Sorted array: 42 46 55 68 77

-----
Process exited after 14.09 seconds with return value 0
Press any key to continue . . .
```

Compilation results:

```
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\VARMA\Documents\INSERTION SORT.exe
- Output Size: 128.6015625 KiB
- Compilation Time: 0.14s
```