DATE: DAY:

Dry - Run

Bubble Sort:

```
                    1  2  3  4  5  6
int arr[] = { 64, 4, 5, 12, 22, 11 }
int n = sizeof(arr) / sizeof(arr[0])
```

24 / 04 → n = 6

↳ 4 bytes

// Bubble sort function:

```
BubbleSort(arr, n) {
```

arr → 64, 4, 5, 12, 22, 11 ; n → 6

```
for (int i=0; i<n-1; i++) {
    for (int j=0; j<n-i-1; j++) {
        if (arr[j] > arr[j+1]) {
            swap(arr[j], arr[j+1]);
        }
```

i = 0

64	4	5	12	22	11
↑ j	↑ j+1				

64 > 4 → swap

4	64	5	12	22	11

j++

DATE: DAY:

4	64	5	12	22	11

j j+1

64 > 5 → swap

4	5	64	12	22	11

j++

j j+1

64 > 12 → swap

4	5	12	64	22	11

j++

j j+1

so on for whole and At last largest number is place at the right side.

4	5	12	22	11	64

Same process for.

i=1

4	5	12	11	22	64

i=2

4	5	11	12	22	64

i=3

4	5	11	12	22	64

i=4

4	5	11	12	22	64

i=5

4	5	11	12	22	64

DATE: _____ DAY: _____

Till end we get sorted list.