

2016/11/19

Variation on a theme → different sorting logic using Chauncy's Bubble Sort

[8, 6, 4, 2, 1, 3, 5, 7, 9]

if odd & even → odd is "larger"

if even & odd → no change

if even & even → sort descending

if odd & odd → sort ascending

how to check odd/even

→ modulo ~~2~~ 2 = $\frac{1}{0}$

```
if (num % 2 == 0) {
    do stuff;
}
```

```
int num = 0;
```

```
if (num[i] % 2 == 0) {
    if (num[i+1] % 2 == 0) {
```

sort even to even desc;

```
} elseif (num[i+1] % 2 == 1) {
```

sort even to odd → odd larger;
(do nothing)

```
}
```

```
return 0;
```

```
if (num[i] % 2 == 1) {
```

```
    if (num[i+1] % 2 == 1) {
```

sort odd to odd asc;

```
} elseif (num[i+1] % 2 == 0) {
```

sort odd to even → odd larger;

```
}
```

Return 1;

```
if (num[i] > num[i+1]) {
```

```
    return 1;
```

```
} else {
```

```
    return 0;
```

```
}
```

* ask Jim how -1, 0, 1 works in qsort
(versus my 1=true, 0=false paradigm)

* next steps: Create void pointers?
→ sort alpha letters?
OR maybe move on????

exercise 3-1; other ch. 3 exercises; play w/ ch 4

10/1/19

Characteristics of a chain → differentiating factor

1. 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

if odd & even → odd & even
if even & odd → no change
if even & even → sort ascending
if odd & odd → sort descending

```
int main() {  
    int n;   
    cin >> n;  
    int arr[n];  
    for (int i = 0; i < n; i++)  
        cin >> arr[i];  
    sort(arr, arr + n);  
    for (int i = 0; i < n; i++)  
        cout << arr[i] << " ";  
    return 0;  
}
```

sort odd to odd and
sort even to even and
sort odd to even and
sort even to odd and

```
int main() {  
    int n;  
    cin >> n;  
    int arr[n];  
    for (int i = 0; i < n; i++)  
        cin >> arr[i];  
    sort(arr, arr + n);  
    for (int i = 0; i < n; i++)  
        cout << arr[i] << " ";  
    return 0;  
}
```

if you have 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
you will get 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

if you have 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
you will get 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

if you have 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
you will get 1, 2, 3, 4, 5, 6, 7, 8, 9, 10