Problem 1

To solve this problem, both two methods require an intermediate parameter to store value. The difference is, for pass-by-pointer, the intermediate parameter should be a pointer too, while for pass-by-reference the intermediate parameter is just a type of integer variable.

Sample run.

```
wyn@wyn-MSI:/mnt/e/wyndwd/ms/22Spring/EECE7205/HW1$ g++ -std=c++11 HW1Q1.cpp -o HW1Q1
wyn@wyn-MSI:/mnt/e/wyndwd/ms/22Spring/EECE7205/HW1$ ./HW1Q1
Before Swap, x=7, y=205
After SwapP, x=205, y=7
After SwapR, x=7, y=205
```

Problem 2

To mirror the array, we just need to swap the numbers around the center. It could be solved by assigning two variables and start a for loop. One variable starts from 0 and add 1 for one loop, the other variable equals to length minus (i+1) because the for loop is starting from 0. For each loop swap the position with two variables as the index. The end condition is the first variable equals to length divided by 2, which means reaching the center of the array.

Sample run.

```
wyn@wyn-MSI:/mnt/e/wyndwd/ms/22Spring/EECE7205/HW1$ g++ -std=c++11 HW1Q2.cpp -o HW1Q2
wyn@wyn-MSI:/mnt/e/wyndwd/ms/22Spring/EECE7205/HW1$ ./HW1Q2
2
8
7
6
5
wyn@wyn-MSI:/mnt/e/wyndwd/ms/22Spring/EECE7205/HW1$ g++ -std=c++11 HW1Q2.cpp -o HW1Q2
wyn@wyn-MSI:/mnt/e/wyndwd/ms/22Spring/EECE7205/HW1$ ./HW1Q2
1
2
8
7
6
5
```

Problem 3

1.

Get all inputs from the user and dynamically save them to a struct array. Then apply a bubble sort algorithm to sort the array with descending order. Finally display the list.

2.

The average could be calculated by total grades divided by student number. Use a for loop to add all grades in the array and calculate the average after loop.
Based on the sorted array, if array length is odd, the median is the center of the array. If array length is even, the median is the average of the center two grades. Therefore, it could be solved by assigning another variable in the average calculating for loop. loop variable starts from 0 and add 1 for one loop, and the added variable starts from the length of the array and minus 1 for one loop. The end condition is the two variables are equal or first one equals to second one minus 1. That means the median one or two grades in the array. Then calculate the median.

3.

Based on the descending sorted array, the first element is with the largest grade.

4.

Based on the descending sorted array, the last element is with the smallest grade.

Sample run.