Note : The input array can be fixed data in your file like that [1, 2, 4, 4] or takes input from user many numbers and save them in empty list then use it as input

1

Write a function **getMin** that takes an array as a parameter for the function and returns the minimum value of the array

For example:

input

```
[ 8, 0, 1, 5 ]
```

output

```
0
```

2

Write a function **CalculateEven** that takes an array as a parameter for the function and calculate the sum of even numbers in the array

For example:

input

```
[ 1, 4, 2, 5, -1, 8 ]
```

output

```
14
```

3

Write a function **count2** that takes an array as a parameter and counts occurrences of the number **2**

for example:

input

```
[ 1, 5, 2, 2, 5, 2 ]
```

output

```
3
```

4

Modify the above function in the previous problem to be **CountGeneral** which takes the array as a parameter and counts occurrences of every number in the array and prints it

Hint: as in the previous example but the input will be every number in the input array

for example

input

```
[ 1, 2, 3, 1, 3, 6 ]
```

output

```
1 - > 2
2 - > 1
3 - > 2
6 - > 1
```

5

Write function **bubbleSort** that takes an array as a parameter and sorts it

Hint: search about bubble sorting and implement it

for example

input

```
[ 1, 0, 3, 6, 2, 5 ]
```

output (using bubble sorting)

```
[ 0, 1, 2, 3, 5, 6 ]
```

6

Modify the above function in the previous problem to be **OptimizedBubbleSort**

Hint: search about how Optimized bubble sort works and implement it

for example

input

```
[ 1, 0, 3, 6, 2, 5 ]
```

output (using optimized bubble sorting)

```
[ 0, 1, 2, 3, 5, 6 ]
```

.

7

Write the function **SelectionSort** that takes an array as a parameter and sorts it

Hint: search about selection sort and implement it

for example

input

```
[ 1, 0, 3, 6, 2, 5]
```

output (using selection sort)

```
[ 0, 1, 2, 3, 5, 6 ]
```

8

Compare between **binary search** and **linear search** in terms of performance and best case and worst case and average case.

Hint: this comparison will be in a text file or comments in a dart file **No Code**

9

Worst case ?? best Case ?? average case ?? all the previous terms are used to analyze and compare between algorithms. Read about these terms and make a comparison between bubble sort and selection sort and insertion sort according to these terms.

Hint: read first about complexity analysis then try to get the answer to the question yourself. then search about it in Wikipedia or any other website, book, or even YouTube, all of them are not secret they are facts based on the analysis) you will not invent the wheel :D **(No Code)**