

## OOP Lab 3

### Question 1

- Write a function `allocateMemory` takes a pointer to an integer array and its size as argument to allocate the memory using a new operator.
- Write a function `initialize` takes a pointer and size as argument to initialize the array with user defined values.
- Write another function `countPrime()`, that takes pointer and size as argument, and counts the total number of prime numbers.
- Deallocate the memory at the end.

*Question for your thought: why are we passing the pointer by reference into the function?*

*Following are the test cases for `countPrime` function, you have to use `gTest` and make a separate `tests.cpp` file. Perform your test cases at the end of the lab task. And remember to run it via terminal*

Test Case	Input Array	Size (s)	Expected Output
1	{2, 5, 8, 10, 13}	5	3
2	{1, 3, 5, 7, 9}	5	0
3	{4, 6, 8, 12, 15, 18}	6	5

### Question 2

- Write a function **overloaded** `allocateMemory` for students data i.e name, age and gender (m for male, f for female). Function takes a pointer to an integer array, char array, string array and `numStudents` as arguments to allocate the memory using a new operator.
- Write a function `initialize` takes the same arguments to initialize the array with user defined values (name, age & gender).
- Write another function `display students()`, that takes pointer(s) and size as argument, and display data for all students.
- Deallocate the memory at the end.

### Question 3

Dynamically allocate with user defined size, then initialize char Array with user defined values, display the array, reverse it and then display it again. Similarly, make another function to check if the array is palindrome or not? Remember to deallocate the memory at the end. Make separate functions for your ease. Set the arguments of your choice.

**Subscript [] notation is not allowed in this question. However you can use pointer arithmetics.**

#### Question 4

Dynamically allocate integer array of **size = 5**, then initialize char Array with user defined values, display the array.

Now assume wants to enter two more integers in the array i.e along with these 5 values, user wants to append two more values. How can you handle it? Delete memory allocated in the end.

#### Question 5 part 1

Dynamically allocate integer ptr with size taken as input, initialize it by taking user inputs.

Write a function having arguments array of integers represented by a pointer `int *ptr` and its length `size`, `target` that represents the target sum. Your task is to i) print ii) number of pairs of numbers from the array whose sum equals the `target`. Following is the sample output for reference. Along with printing, the function will return 3 because there are three pairs.

**Also implement a following test case, but the expected output will be 3**

*Input:*

- `ptr = [2, 4, 3, 5, 6, -2, 8, 10]`
- `size = 8`
- `target = 8`

*Output:*

- The pairs that sum to 8 could be:
  - (2, 6)
  - (3, 5)
  - (-2, 10)

#### Question 5 part 2

Extend the functionality, write a function sort that take ptr and size as an argument and sort the array. Display the array elements after sorting.