



Assignment 2 Report
Ehab Adel EL-Kady
1610505

How to build:-

for building any of the problems, the following commands have to be written in the terminal

```
g++ main.cpp
```

```
./a.out
```

Problems explanation and goal:-

problem 1:-

Problem 1 is like an attendance checking system in which there are 256 students and there are 8 classrooms each having 256 chairs. Each student sits on the chair of his id and no one else can sit on his chair.

The goal of this problem is to use memory wisely and choose the perfect data types that would use less memory.

Problem 2:-

Problem 2 is an inventory check system. This inventory has up to 256 products each product can have 8 colors. Each product can have up to 256 items of that color.

The goal of this problem is also to manage the use of memory and choose the perfect data types.

Problem 3:-

Problem 3 is a maximum heap data structure in which the operators of addition and assignment and indexing can be performed upon.

The goal of this problem is to use operators overloading and how to handle the increase of memory cases and allocating new memory in case of exceeding memory limits.

Design decisions:-

=>For problem 1 :-

The data types chosen to represent the data of the student or the seat number is `uint8_t` as this data type uses only 1 byte and not like the normal `int` data type that uses 4 bytes which will reduce the memory used as there is no need for more than 8 bits as the range of any of the values is from 0 to 255 which means that only 8 bits is sufficient for representing these data.

The problem was designed to be in two classes to avoid the usage of two dimensional array that will consume so much memory when creating a 256 instances of the classes.

=>For problem 2:-

Just like problem 1, the data type used is `uint8_t` to create a 2 dimensional array, the first dimension represents the index of the product while the second represents the color. The colors are indexed from 1 to 8 and to access any of the product colors enter one of the color codes to access its data. The problem was designed to be in one class with two dimensional array as it is the best design.

=>For problem 3:-

A method added that makes the heap order as maximum heap called buildmax() that takes no parameters. The operators required were implemented. In the += operator realloc function was used in order to increase the max_size of the size of the heap and avoid any restrictions for adding elements to the other heap. Also in the example used in the main.cpp file, the heap object was created as a pointer and to use any of the implemented operators, it has to be dereferenced by the asterisk sign (*) to access the operators. The heap was implemented in one class that contains the prototype functions in it and their implementation is right below the class. Using prototypes is important to organize the code and increase its readability.

Achievement:-

This assignment made me use operators overloading and learn how to use them well also I learned how to manage the memory well and the use the perfect data types that will use the memory needed without any usage of useless memory or not needed.

A link to GITHUB repository that has the code with the make file and its report:-

<https://github.com/ehabel-kady/-cs-Assignment2>