Practical 5 (Week 7)

Task 5.1: Download the file "`task5_1.cpp`" from vUWS and complete the following sub-tasks:

1. Run the code and fully understand what the code does.
2. Rewrite the program in OO-style. Your new program should:
   a. have exactly the same functionalities as the original code;
   b. contain a class, named `ProductionGraph`, with
      i. a private data member, which is a pointer of integers;
      ii. a constructor that takes a parameter as the size of a dynamic array and allocate the size of memories to the dynamic array
      iii. a destructor that releases the dynamically allocated memories.
   c. contain a main function that takes an input from the user as the number of plants and pass it via a `ProductionGraph` object to the class.

Task 5.2 Download the file "`task5_2.cpp`" from vUWS. Explain to your tutor the meaning of the following statements:

```
1. int **grid = new int*[row];

2. for (int i = 0; i < row; i++)
       grid[i] = new int[col];

3. for (int i = 0; i < row; i++)
       delete[] grid[i];

4. delete[] grid;
```

Task 5.3 Download the file "`task5_3_OOP.cpp`" from vUWS, which is an OOP version of the program `task5_2.cpp`. Fill in the missing code in functions `fillArray()` and `printArray()`by referring to the original code of `task4_2.cpp`.

Task 5.4 Create a class named `Student` that has three member variables (all private):

name - A string that stores the name of the student numUnits - An integer that records how many units the student is currently enrolled in

unitList - A string pointer that will be used to point to a dynamic allocated array to store the names of the units that the student is enrolled in

Write appropriate functions including:

- A constructor that takes two parameters – student name and number of units enrolled meanwhile creates an array of strings using dynamic memory allocation.
- A function that takes user's input of the list of units.
- A function that displays the student name and the list of units.
- A destructor that releases dynamically allocated memory.

Write a main function that tests all of your functions.

Hints: The way of declaring an array using dynamic memory allocation is:

```
unitList = new string[numUnits];
```

Task 5.5 (Assignment 1 pre-check) Complete **Tasks 1** of **Assignment 1** and show your code to your tutor. Note that your tutor is not in the position to mark your work at this stage. They could tell you if you are in the right direction or not but may not tell you exactly what you have done is correct or not.