Expectation Melchor A. BSIT - 1 - 1B

1. What is 2d array?

- A two dimensional array is a data structure that contains a collection of cells laid out in a dimensional grid, similar to a table with rows and columns although the values are still stored linearly in memory.

2. How to declare 2D array?

- To declare a 2D array, specify the type of elements that will be stored in the array, then ([][]) to show that it is a 2D array of that type, then at least one space, and then a name for array.

3. How to initialize a 2D array and give example.

- data type array name [][] = { };
- int disp [2][4] = {
{ 10, 11, 12, 13 },
{ 14, 15, 16, 17 }
};

4. When and why use 2d?

- The two-dimensional array can be define as an array of arrays. The 2D array is organized as matrices which can be presented as the collection of rows and columns. However, 2D arrays are created to implement a relational database lookalike data structure.

5. How to access the 2D array element

- To access an element at position (i,j), we use array_name[i-1][j-1]. Thus the element at the 4th row and 3th column will be accessed by A[3][4]

6. Why use nested loops in storing and accessing the data in array?

- Nested Loops have an advantage over other join methods that they can quickly retrieved the first few rows of the result set without having to wait for the entire result set to be determined.

7. Write a code to read the exam score of students. Compute the weighted average score for each student.

Student 1	85	78	89	Average 84.00
Student 2	80	83	87	83.33
Student 3	90	91	86	89.00

```
int main()

    float average[3][3][3];
    int sum[3][3][3];
    int grade[3][3][3];

    for(int i=0; i<3; ++i)
    { printf("Student %d\n", i+1);
      for(int j=0; j<3; ++j)
      { printf("Enter your grade: ");
        scanf("%d", &grade[j]);
        sum[j][j][j] += grade[j];
        average = sum / 3.0;
      }
    }
    for(int m=0; m<3; ++m)
      { for(int n=0; n<3; ++n)
        { printf("%d\t %d\t %d\t %d\t %.2f\n", grade[n], sum[n],
                                                 average[n]);
        }
```