Dynamic and Jagged Arrays

Lesson 2.2

Learning Objectives

LO 2.2.1 **Create** and **initialize** dynamic/jagged arrays

LO 2.2.2 **Access** data in licit locations of the dynamic/jagged array

Dynamic Array Declaration

- Just like static arrays, dynamic arrays are also allocated in contiguous memory addresses.
- The only difference is that static arrays exist in the stack memory while dynamic arrays exist in the heap memory.
- When we allocate dynamic arrays, we just multiply the number of bytes we have for a single value.
- Example:

```
float* grades = (float*)
                malloc(sizeof(float)*5);
```

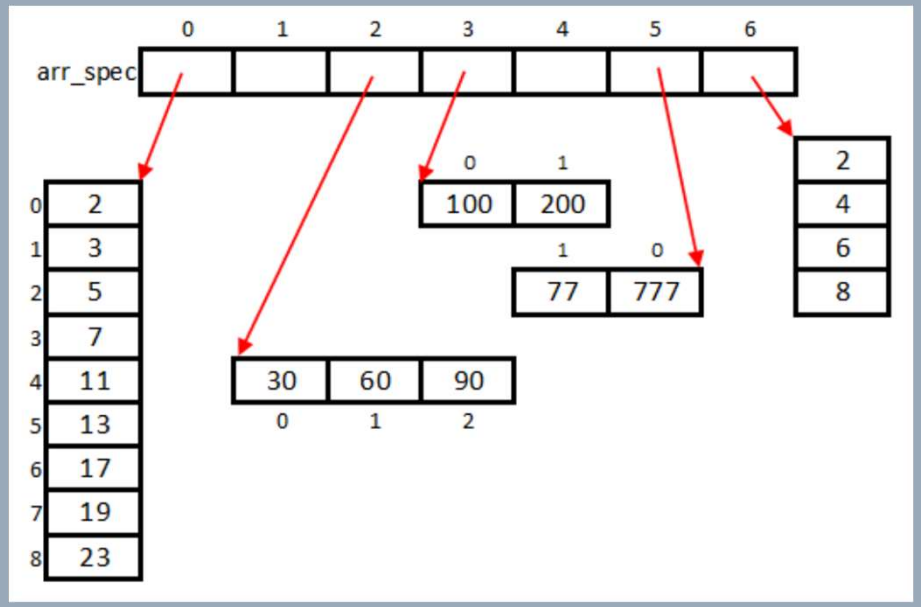
Accessing Elements in the Dynamic Array

- Accessing elements in the dynamic array is the same as accessing elements in static arrays.
- Example:

```
float ave = (grades[0]+grades[1]+grades[2]+
             grades[3]+grades[4])/5;
```

Jagged Array Declaration

- Jagged arrays are multi-dimensional dynamic array of pointers.
- The crucial part in declaring jagged arrays is in deciding what data type to declare/use.



Strengthening the Learning Objectives

LO 2.2.1 Create and initialize dynamic/jagged arrays

LO 2.2.2 Access data in licit locations of the dynamic/jagged array

Example:

1. Ask a user for a number n.
2. Create a dynamic array of characters with n number of elements.
3. Ask user to input n characters and store them sequentially on the dynamic array you have just created.
4. Display all elements of the dynamic array as a string (*in one line with no delimiters*).

LO 2.2.1 Create and initialize dynamic/jagged arrays

LO 2.2.2 Access data in licit locations of the dynamic/jagged array

Example:

1. Ask a user for a number n.
2. Create a dynamic array of integers with n number of elements.
3. Ask user to input n integers and store them sequentially on the dynamic array you have just created.
4. Display all elements of the dynamic array divisible by 3.

LO 2.2.1 Create and initialize dynamic/jagged arrays

LO 2.2.2 Access data in licit locations of the dynamic/jagged array

Example:

1. Create a jagged array that is conceptually equivalent below ([ ] *are indices*):

[0]	0				
[1]	0	1			
[2]	0	1	2		
[3]	0	1	2	3	
[4]	0	1	2	3	4

LO 2.2.1 **Create and initialize dynamic/jagged arrays**

LO 2.2.2 **Access data in licit locations of the dynamic/jagged array**

Example:

2. Display the elements of the jagged array.
3. Swap the index 1 and index 4 of the jagged array.
4. Display the elements of the jagged array.