Introduction:

Welcome to assignment 5, in this assignment you are tasked with allocating and managing a dynamic array.

What is a dynamic array?

A dynamic array is an array that is allocated at runtime, which means that the compiler does not need to know the size of the array at compilation.
A dynamic array starts at size 0, and grows as it is being filled dynamically.
A well known data structure that is commonly used to represent a dynamically sized array is a Vector, which is what you will be implementing in this assignment.

Assignment Instructions:

Requirements

1. Fill in sthe missing code snippets labelled in the following sections of this document
2. Only modify code in the file `vector.c`
3. Ensure that there are no memory leaks in your program, you are responsible for managing the memory that you allocate

CreateVector Function

This function handles the creation of dynamically allocated vectors:

- Makes a memory allocation request
- Checks if the memory was allocated successfully, else throws an error
- Initializes the size and capacity of the vector
- Returns the vector

PushBack Function

This function creates a new element and pushes it to the back of the specified vector.
It resizes the array when the max capacity has been reached:

- Checks if the size has reached the capacity
- Resizes the vector if it is full
    - Checks if the memory was reallocated successfully, else throws an error
- Adds the new element to the end of the vector
- Increments the size of the vector

GetElement Function

This function accesses the vector to retrieve the value at a specified index:

- Checks if the index is within the range of the vector
    - Returns the element at index

- Else, throws index out of bounds error

DeleteVector Function

This function frees up the heap allocation for the vector:

- Frees the allocated memory

Note:

1. The `DEFAULT_CAPACITY` is set to be 10 in the `vector.h` file

Submission Requirements:

- You are required to modify a single C source code file named `vector.c` that contains the implementations of the functions declared in `vector.h`
- The `!run.bat` batch file contains test cases for this assignment, it will also check your output against the expected output file `sample_output.txt`
- Your program should minimally reproduce the output shown in the example below

```
Test Case 1: Adding elements to the vector
vector 1: 0 2 4 6 8 10 12 14 16 18

Test Case 2: Accessing elements in the vector
vector 2: 0 3 6 9 12
Element at index 2: 6

Test Case 3: Resizing of the vector
vector 3: 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70
```