LOTTERY NUMBERS GENERATOR REPORT

As an occasional lottery player, I developed an program to help me generate random lottery numbers using linklist and vectors data structures.

A data structure is a particular way of organizing data in a computer so that it can be used effectively.
For example, we can store a list of items having the same data-type using the *array* data structure.
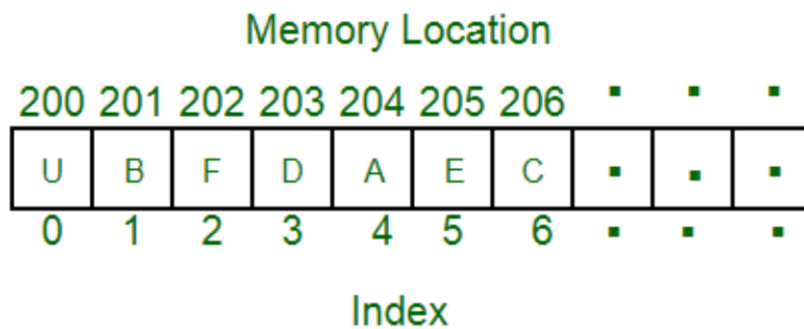
It is about rendering data elements in terms of some relationship, for better organization and storage. These can either be a string data or an integer data. Data structure represent the knowledge of data to be organize in memory. It is designed in such a way that it increases the efficiency and reduce its complexity.

This lottery program is designed to randomly generate numbers for users. The program has four types of lottery games. This includes;

- EuroMillions
- Lotto
- Set for life
- Thunder Ball

Using linked list and vectors as data structures, this program is designed in three operational stages.
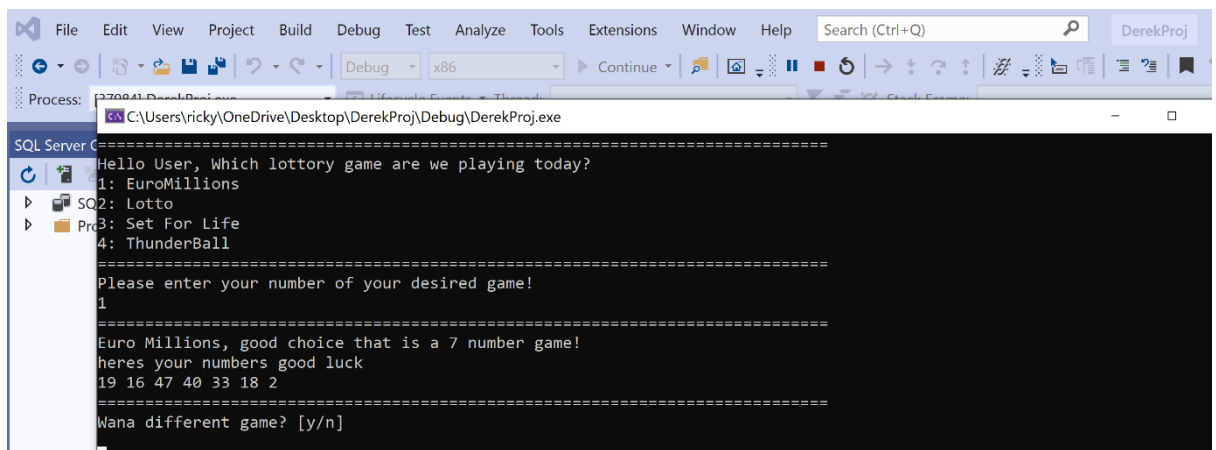
User Interaction (Interface)

This section contains all user instructions for all the games. This are instructions that the user will have to follow to get his/ her lottery numbers generated using vectors.
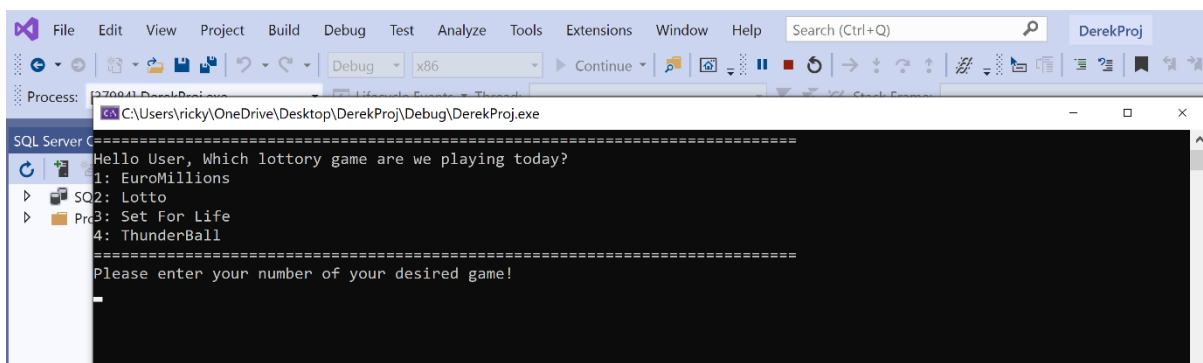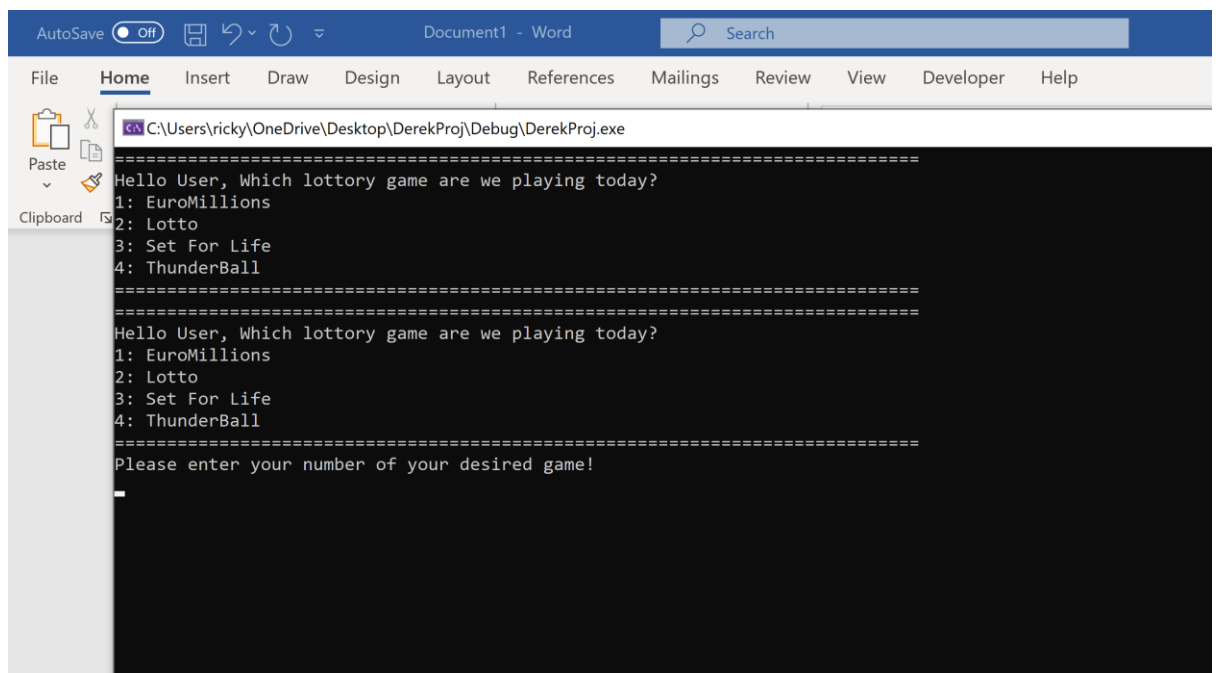
As the first image above, the user will select one of the lottery games and press enter to input it.

Upon selecting one as the second image shows, a message will display with the generated lottery numbers. An optional instruction has also bee provided to enable the user play another game.

Upon clicking (y) for yes and enter, the previous game and the generated numbers will be cleared. The code below clears the previous command and generated lottery numbers before instructing the user to choose another game or replay.

```
void Clear()
    {
            #if defined _WIN32
            system("cls");
            #elif defined (__LINUX__) || defined(__gnu_linux__) ||
defined(__linux__)
            system("clear");
            #elif defined (__APPLE__)
            system("clear");
            #endif
    }
```

User selected game three for set for life and it goes through the same process to generate numbers for the game.


```
C:\Users\ricky\OneDrive\Desktop\DerekProj\Debug\DerekProj.exe
===============================================================
Hello User, Which lottory game are we playing today?
1: EuroMillions
2: Lotto
3: Set For Life
4: ThunderBall
===============================================================
===============================================================
Hello User, Which lottory game are we playing today?
1: EuroMillions
2: Lotto
3: Set For Life
4: ThunderBall
===============================================================
Please enter your number of your desired game!
3
===============================================================
Set for life, thats adventerous! thats a 6 number game!
best of luck with you numbers!
3 4 43 39 18 30
===============================================================
Wana different game? [y/n]
```

VECTOR DATA STRUCTURE

Vectors are sequence containers representing arrays that can change in size and are uses contiguous storage locations for their elements. This means that the elements can also be accessed using offsets on regular pointers to its elements.
Vectors can resize itself automatically when an element is inserted or deleted depending on the need of the task to be executed.

In this lottery game, the push_back() modifier is used to push elements into the vector from the back. If the type of object pass as a parameter in the push_back() is not same as that of the vector, then an exception is thrown. For example, the vector of my lottery games is a string, so if an integer is passed as a parameter then an exception will be thrown.
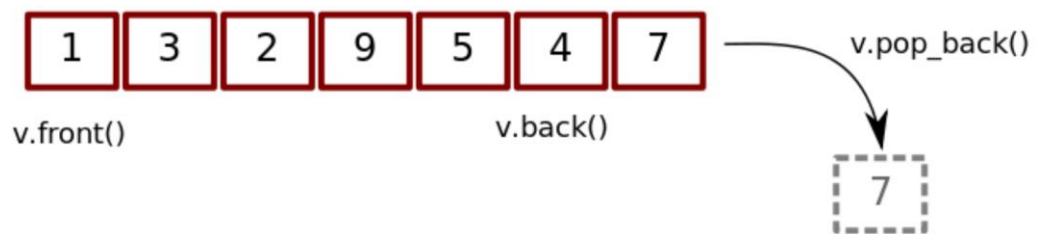
LINKED LIST DATA STRUCTURE

A linked list is a linear dynamic data structure to store data items. Unlike arrays, the linked list does not store data items in contiguous memory locations. A linked list consists of items called "Nodes" which contain two parts. The first part stores the actual data and the second part has a pointer that points to the next node. This structure is usually called "Singly linked list".
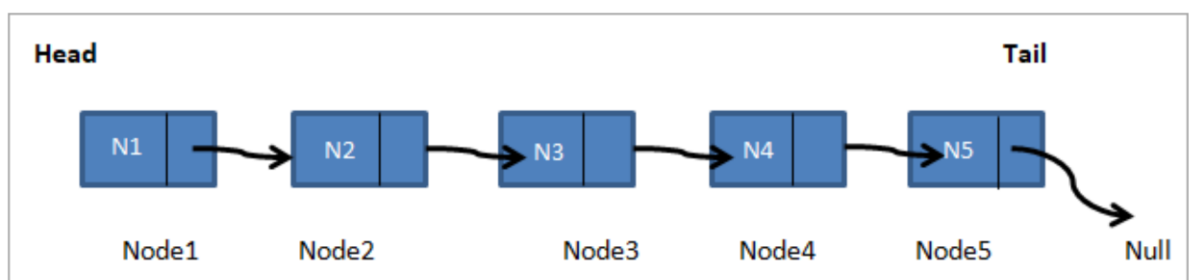
In this program, some of the lottery games generates 6 numbers and others generate 7 numbers. There are separate link list for the game that generates 6 numbers and a different lint list that generates the 7 numbers.

Types of Linked List
There are different types of link list, this include;
1.
   Singly Linked List : In this type of linked list, every node stores address or reference of next node in list and the last node has next address or reference as NULL.
2. **Doubly Linked List :** In this type of Linked list, there are two references associated with each node, One of the reference points to the next node and one to the previous node. Advantage of this data structure is that we can traverse in both the directions and for deletion we don't need to have explicit access to previous node.
3. **Circular Linked List :** Circular linked list is a linked list where all nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be a singly circular linked list or doubly circular linked list. Advantage of this data structure is that any node can be made as starting node. This is useful in implementation of circular queue in linked list

During the programming, I decided to use cases instead of if-else statements because of the following reasons.

- Easier to debug
- Easier to read
- Easier to understand, and therefore
- Easier to maintain Faster execution potential

This made it easier for my program to debug and errors can be spotted easily.

CONCLUSION

Developing this program was something difficult for me. I used a lot of resources from online tutorials (Data structures and Algorithm on Udemy and YouTube) and books to understand very well how I can make the program work as I had some few errors compiling.

Linked lists are the data structures that are used to store data items in this program but in non-contiguous locations. A linked list is a collection of nodes that contain a data part and a next pointer that contains the memory address of the next element in the list. Linked lists are expensive as far as their traversal is concerned since we cannot randomly access the elements like arrays.