

# Data Structures and Fundamentals of Programming

## Problem #1

In C++ implement a **generic** class, called `Stack<T>`, that uses a dynamically allocated **single-linked list** implementation. This must implement the stack ADT. It should be generic on the type of the data to be stored. Give all class definitions and implement the following for `Stack`:

- Default constructor
- Destructor
- Copy-constructor
- Assignment operator
- `push(T)` – takes an parameter of type `T` and adds it to the stack
- `T pop()` – removes a node from the stack

Note: Your implementation can **NOT** use STL or any other libraries (standard or otherwise).

## Problem #2

Given a string ADT as defined below, write a method that returns a string that is a substring starting at a given position (`start`) and ending at a position (`end`). Do all error checking.

```
class string
{
public:
    string() {s[0] = 0;};
    string substring(int start, int end) const;

private:
    char s[256]; //null terminated character array
};
```

## Problem #3

In C++ implement a **generic** class, called `Queue<T>`, that uses a fixed sized **circular array** implementation. This must implement the queue ADT. It should be generic on the type of the data to be stored. The implementation must be able to utilize the entire array in storing items. Give all class definitions and implement the following for `Queue`:

- Default constructor
- `push(T)` – takes an parameter of type `T` and adds it to the queue
- `T pop()` – removes a item from the queue
- `isEmpty()` – returns true when the queue is empty.
- `isFull()` – returns true when the queue is full.

Note: Your implementation can **NOT** use STL or any other libraries (standard or otherwise).