Assignment 6

Recursive conversion	25 pts.
Sum of numbers	30 pts.
IntArray Class with Exceptions	45 pts.
TOTAL	**100 pts.**

Part 1 (Recursion – Class 11)

Recursive conversion

Convert the following function into a one that uses recursion.

```
void sign(int n)
{
      while (n-->0)
            cout << "No Parking \n";
}
```

Demonstrate function with the driver program.

Sum of numbers

Write a recursive function that accepts an integer argument and returns the sum of all the integers from 1 up to the number passed as an argument. For example, if 50 was passed as an argument, the function will return the sum: 1+2+3+…+50.
Demonstrate the function in a program.

Part 2 (Exceptions – Class 12)

IntArray Class with Exceptions

1. Modify the definition of the IntArray class (code provided in **IntArray.cpp file**), so it has an exception class *OutOfRange* defined in the public section. The exception class should have a private *int* member and a private *string* member. It should have a public constructor that has *int* and *string* arguments, and a set of accessor functions.

2. Modify the implementation of *operator []* of the IntArray class so it throws exceptions of type *OutOfRange* instead of calling subscriptError() function when index is out of range. The offending index value along with a message should be stored in the exception object. Choose the appropriate message to describe the situation.
3. Test your code. Write a small driver program that gets an array index from the user and calls [] operator with that index in the try block. Write a catch block that catches exceptions of type *OutOfRange* , and outputs appropriate messages on the screen.
4. Test-run your code, calling [] operator with an invalid index, one that is out of range for the array in your code
    - example: table[-5].

   See whether the exceptions is being thrown and caught.