CS 280
Fall 2022
Recitation Assignment 6
October 20, 2022

Due Date: Monday, October 24, 2022, 23:59
Total Points: 5

Write a C++ function, called *BinToDec()*, using recursion to convert a number in binary to a decimal number. The algorithm assumes that all passed strings for a binary number to the function are correct. The algorithm states that each successive digit in the number is multiplied by the base (2) raised to the power corresponding to its position in the number. The low-order digit is in position 0. We sum together all of these products to get the decimal value. For example, if we have the binary number 111001, we convert it to decimal as follows:

$1 * 2^0 = 1$
$0 * 2^1 = 0$
$0 * 2^2 = 0$
$1 * 2^3 = 8$
$1 * 2^4 = 16$
$1 * 2^5 = 32$
Decemal value = 1 + 0 + 0 + 8 + 16 + 32 = 57

A recursive formulation of this algorithm can extract each digit and compute the corresponding power of the base by which to multiply. Once the base case (the last digit is extracted) is reached, the function can sum the products as it returns.

The `BinToDec` has the following header definition:

```
int BinToDec (string binary);
```

where the function returns an integer and accepts a string representing the binary number.

Vocareum Automatic Grading

- Implement the `BinToDec()` function in a file, called "`BinToDec`.cpp" and upload it to Vocareum.
- You can implement any driver program for testing your implementation. However, a driver program is provided on Vocareum for testing the implementation, called "prog.cpp". The "prog.cpp" will be propagated to your work directory. The program asks the user to enter a binary number, calls the `BinToDec()` function, then prints out the result of conversion from binary to decimal.
- The testing is based on the correct results returned by the call to your recursive function `BinToDec()`, and the satisfaction of using recursive approach for the solution of the problem, rather than an iterative one. Test case files will be used instead of the input from the keyboard on Vocareum. There are 4 testing files associated with RA 6. Vocareum

automatic grading will be based on these testing files. You may use them to check and test your implementation. These are available in a compressed archive "RA6 Test Cases.zip" on Canvas assignment. The testing case of each file is defined in the Grading table below.

- "prog.cpp" is available with the other assignment material on Canvas.

Submission Guidelines

- Please upload your implementation of the function in a file to Vocareum as a "`BinToDec`.cpp" or any other file name you choose. The file should include the implementation of the function `BinToDec()`.

- **Submissions after the due date are accepted with a fixed penalty of 25% from the student's score. No submission is accepted after Wednesday 11:59 pm, October 26, 2022.**

Grading Table

Item		Points
Function Implementation	Case 1	1
	Case 2	1
	Case 3	1
	Case 4	1
	Compiles Successfully	1
Total		5