

**CS 210**  
**Assignment 07**  
**String functions**

**October 20, 2016**  
**Due: November 1, 2016**

Reminder: This is a programming project, and work on this assignment should be done individually. Assistance from other students is limited to questions about specific issues as noted in the syllabus.

For this assignment you will write three "bullet-proof" string function: `Str2Double`, `Int2Bin`, and `Bin2Int`.

`Str2Double` – this function accepts a string less than 80 characters long, parses the string, and converts the result to a double which is returned in an output parameter. The syntax is:

```
status = Str2Double(char *s, double *d);
```

where

status is an int such that

status = 0 => Unable to convert the input to a double. Sets d to zero.

status = 1 => Successfully converted input to a double. There were no invalid characters in the input stream.

status = -1 => Converted input to a double but there were invalid characters in the input stream.

d is a double that contains the return value. If the status is 0 then d = 0.

s is the input string.

`Str2Double` attempts to parse the input string and convert it to a double. If there is more than a single decimal point in the input string or, if there are no digits in the string, `Str2Double` returns a status of 0.

If there is at least one digit and no more than one decimal point, the other non-digit characters are ignored and remaining characters are assembled into a double. The status, in this case, should be set to 1 if the only characters are digits and a decimal point – otherwise it should be set to -1. This is summarized in the table below.

Status	double	Explanation
+1	Value	String has 0 or 1 decimal point and all other characters are digits
0	0	String has more than one decimal point or there are no digits in the string.
-1	Value	String has 0 or 1 decimal points, at least one digit, and there are additional non-digit characters which are ignored.

`Bin2Int` – This function accepts a char array argument consisting of '1' and '0' characters (only). It interprets the array a binary number with the MSB at index 0 and returns the equivalent int. The binary input is in twos complement form and may be up to 16 digits long. The output will range from  $-2^{31}$  to  $2^{31}-1$ .

The syntax is:

```
b = Bin2Int(char *bin);
```

where

b is the integer result ranging from  $-2^{31}$  to  $+2^{31}-1$

bin is a char array holding only '1' and '0' characters which is up to 16 characters long.

If bin is longer than 16 characters your program should ignore all of the excess characters. If bin is empty your program should return 0.

Int2Bin – this function converts a 16-bit integer to a binary equivalent represented by a character string. You can implement this function using either of the following two forms:

```
char *Int2Bin(int n)
```

Or

```
void Int2Bin(char *s, int n)
```

In the first instance you are returning a string but this is a little tricky. In the second instance the string is defined in the main program and passed as a pointer. In either case the function produces a twos complement 16-character string which represents the binary value of n. If n cannot be represented by a 16-bit binary twos complement number you should return all zeros.

To test your three functions you should write a main program that prints results for all of the following cases with appropriate comments:

Str2Double has input cases of "0", "123", "1.2345", "ABC", "A1.B23"

Bin2Int has input cases of "00101", "11011", "", "00000000000000011"

Int2Bin has input cases of 5, -5, and 32768

Turn in a zipped project file. Name your zipped file Asn07XXX.zip where XXX are your three initials. Upload your zipped project file to

\\cecsfp01\users\everyone\CS210.