Course : COMP-2650
Instructors : Dr. Boufama
Assignment : 04
Due date : Friday March 29, 11.59pm

Write a program in C/C++ (or Python or Java), call it **AddSub**, to add and subtract binary numbers of any number of bits, up to 100 bits.

Your assignment should meet the following requirements :

- The program takes 3 space-separated string arguments, i.e., a binary number, an operator (+ or -) and another binary number.
- The binary numbers can have up to 100 bits.
- If the user does not provide 3 arguments, your program should prompt a help message on how to use the program then, exits.
- A function, called **int parseBinary(char *)** should be implemented and used in the program to check if the binary strings contains 1s and 0s only. If the inputs have illegal characters, an error message should be printed and the program should exit.
- A function, called **void addZeros(char *, int);** should be implemented and used in the program to make the two binary strings of equal length. Basically, the shorter string should be padded with 0s on its left, so that both strings end up having the exact same length,
- A function, called **void complement(char *)** should be implemented and used in the program to obtain the 1's complement of a bit string. Basically, any 0 becomes 1 and any 1 becomes 0 in the tring.
- A function, called **char FullAdder(char A, char B, char C, char *S)** should be implemented and used in the program to add three bits, that is **A**, **B** and **C** (carry). The sum bit should be stored in ***S** and the carry out should be returned as a character. This function simulates the gate-based full adder we have seen.
- The addition/subtraction are performed in a function called **int AddSub(char *N1, char *N2, char C, char *R)** that takes as input two bit strings of the same length, **N1** and **N2** and adds them with the help of the function **FullAdder()**. Note that the first carry **C** is equal to 1 for the subtraction and 0 for the addition, like the Adder/Subtractor circuit we have seen. When **C==1**, an extra 1 is being added to make our 1's complement become 2's complement,
- Your program should print the result as a string of 1s and 0s.