

CMPT 120 - FIC 201903 - Assignment 2

Due Date: Tuesday 5th November 2019 at 11:55PM

Instructor: Koopa Hakimi

Read this document in its entirety and carefully before you start anything and understand it. If you have any questions, don't hesitate to email me.

Copy **EXACTLY AS IT IS** the main program given below:

```
# Main Program
LENGTH = int(input("Please enter list length "))
a = int(input("Enter a "))
b = int(input("Enter b "))
c = a + b

L1 = decimalToTwosComplement(a, LENGTH)
print "The two complement representation of", a, "is", L1

L2 = decimalToTwosComplement(b, LENGTH)
print "The two complement representation of", b, "is", L2

L3 = twosComplementBinaryAddition(L1, L2)
print "The two complement addition of", a, "and", b, "is", L3

d = twosComplementToDecimal(L3)
print "Converting the two complement to decimal gives", d

if (c == d):
    print "Since", c, "==", d, ", it seems we did good job."
else:
    print "Since", c, "!=", d, ", either of a, b, c must be outside of the range."
```

Implement the three functions **decimalToTwosComplement**, **twosComplementBinaryAddition**, and **twosComplementToDecimal** so that the program will be able to add decimal numbers in two's complement. Description of the functions is as follows.

- **def decimalToTwosComplement(x, LENGTH):**
 - This function takes an integer **x** (negative, positive or zero) and integer **LENGTH** as arguments and returns a Python List of integers of length **LENGTH**. The list elements must be integer type. Each element of the list is a binary digit of 0 or 1 (which are integers) obtained by the conversion of the integer argument **x** to two's complement representation. For example, if **x** is -2358 and **LENGTH** is 16, then the returned list must be **[1,1,1,1,0,1,1,0,1,1,0,0,1,0,1,0]**.
 - If the integer **x** is too large (positive or negative) for its two complement to fit in a list of length **LENGTH**, then this function must discard the extra bits on the left and return a list containing **LENGTH** bits as discussed in class. See week 1 lab work examples.

➤ **def twosComplementBinaryAddition(L1, L2):**

- This function takes two list arguments **L1** and **L2** each of which has length **LENGTH**. The elements in the lists are binary digits representing decimals in two's complement. It returns a list of length **LENGTH** and whose elements are obtained by the addition of the two list arguments in binary addition format. **If there is an overflow during the addition operation, this function must ignore the overflow and return the LENGTH bit result as discussed in class** (see Week1 Lab Work). For example, if **LENGTH** is 16 and the arguments are **[1,1,1,1,0,1,1,0,1,1,0,0,1,0,1,0]** and **[0,0,0,0,0,1,0,0,1,1,1,0,0,1,1,0]** then this function must return **[1,1,1,1,1,0,1,1,1,0,1,1,0,0,0,0]**.

➤ **def twosComplementToDecimal(L3):**

- This function takes a list **L3** of length **LENGTH** that is the two's complement binary representation. The function converts it to decimal and returns the decimal integer number. For example, if **LENGTH** is 16 and the list argument is **[1,1,1,1,1,0,1,1,1,0,1,1,0,0,0,0]** then the function returns **-1104**.

Please note that the **LENGTH** can be any positive integer. So your functions must be designed to work for the value that will be specified by the user. Possible values of **LENGTH** can be 6, 8, 12, 16 or 21 or any other positive integer. You must also test your program for different input numbers a and b and different values of **LENGTH**.

Submission Format

You are required to submit your program online through Moodle. You will find a submission button for **Assignment 2** on **Moodle on Week 9** and you are required to upload your program written as Python Script there. No assignment is submitted through email or hard copy; you must upload your work before the deadline onto Moodle.

Submission Deadline

The deadline to upload your program online is **Tuesday 5th November 2019 at 11:55PM**. Moodle will not allow you to upload after this date and time.

Marking

A non working program will automatically get zero. A program that works but doesn't give right output or gives partial right output will lose marks depending how severe its shortcoming is. This assignment carries 6% of the total course marks.

Sample Run of the Main Program

Here are some sample runs

Please enter list length 6

Enter a 5

Enter b -24

The two complement representation of 5 is [0, 0, 0, 1, 0, 1]

The two complement representation of -24 is [1, 0, 1, 0, 0, 0]

The two complement addition of 5 and -24 is [1, 0, 1, 1, 0, 1]

Converting the two complement to decimal gives -19

Since $-19 == -19$, it seems we did good job.

Please enter list length 9

Enter a 54

Enter b -187

The two complement representation of 54 is [0, 0, 0, 1, 1, 0, 1, 1, 0]

The two complement representation of -187 is [1, 0, 1, 0, 0, 0, 1, 0, 1]

The two complement addition of 54 and -187 is [1, 0, 1, 1, 1, 1, 0, 1, 1]

Converting the two complement to decimal gives -133

Since $-133 == -133$, it seems we did good job.

Please enter list length 5

Enter a 8

Enter b 10

The two complement representation of 8 is [0, 1, 0, 0, 0]

The two complement representation of 10 is [0, 1, 0, 1, 0]

The two complement addition of 8 and 10 is [1, 0, 0, 1, 0]

Converting the two complement to decimal gives -14

Since $18 \neq -14$, either of a, b, c must be outside of the range.

Please enter list length 5

Enter a 35

Enter b -45

The two complement representation of 35 is [0, 0, 0, 1, 1]

The two complement representation of -45 is [1, 0, 0, 1, 1]

The two complement addition of 35 and -45 is [1, 0, 1, 1, 0]

Converting the two complement to decimal gives -10

Since $-10 == -10$, it seems we did good job.

Please enter list length 25

Enter a 34256

Enter b -76543

The two complement representation of 34256 is [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0]

The two complement representation of -76543 is [1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1]

The two complement addition of 34256 and -76543 is [1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1]

Converting the two complement to decimal gives -42287

Since $-42287 == -42287$, it seems we did good job.