CISC 260 Machine Organization and Assembly Language
(Spring 2018)

Assignment # 1

(Due: February 22, 2018)

Instruction for submission:

- **Submit via Canvas**
- **For problem 4, your code should be in plain text file by itself.**

1. [25 points] Convert the following numbers to other data representations. The binary is 8-bit, interpreted as two's complement.

Decimal	**Hex**	**Binary**
-47	0xD1	1101 0001
-91	0xA5	1010 0101
89	0x59	0101 1001
90	0x5A	0101 1010
-11	0xF5	1111 0101

2. [25 points] ASCII code.
   a) Decode the following bit sequence (expressed in hexadecimal): X41524D2069732066756E21
   b) Encode the following word to bit sequence (expressed in hexadecimal): Hello

a)ARM is fun!

b)X48656C6C6F

3. [20 points] With x = $0111\ 1001_{two}$ and y = $1100\ 0101_{two}$ representing two's complement signed integers, perform the following operations, showing all work:
   a. x + y
   b. x - y

   State if an overflow occurs.

a. x+y=00111110

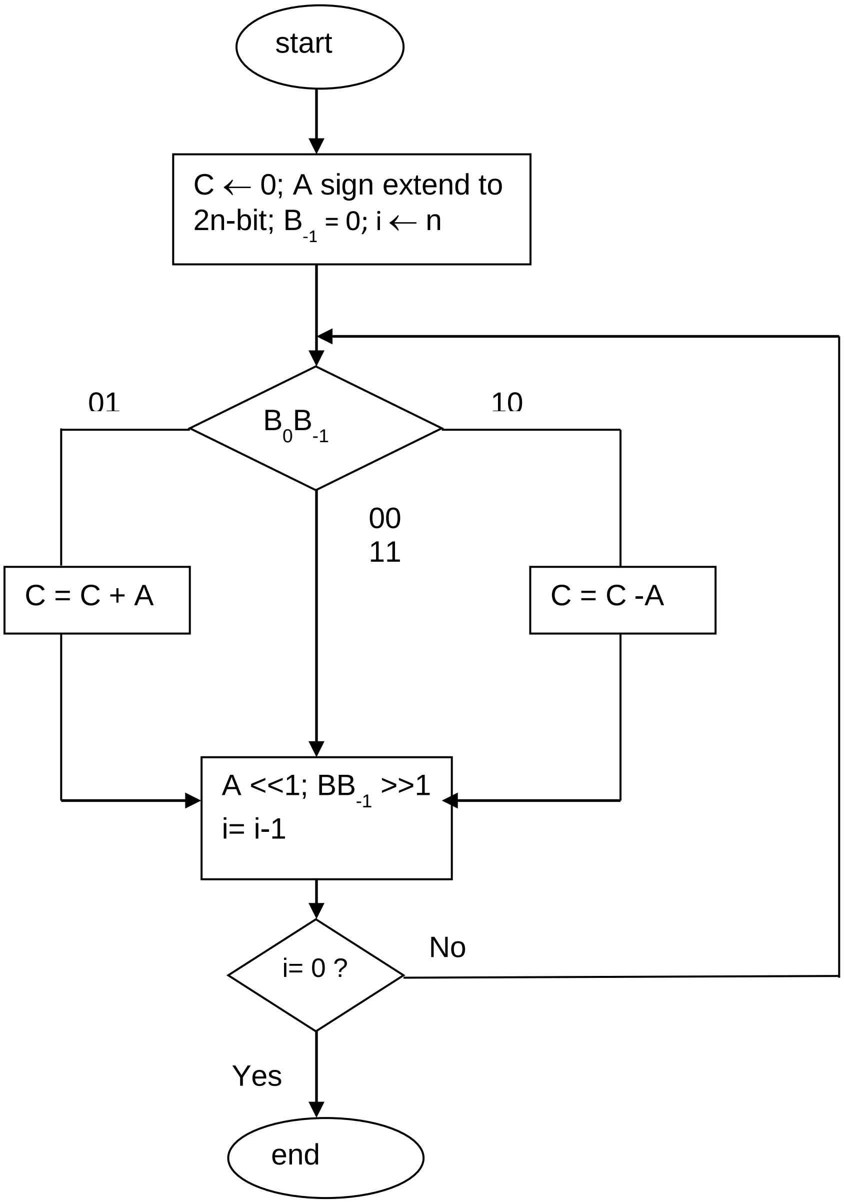
b. x-y=10110101—overflow occurs

4. [30 points] Write a C program to implement the Booth algorithm for multiplication of signed integers, as discussed in class. You may assume the input a and b are small enough, i.e., only require 16-bit, so that the product can fit into 32-bit machine. The following is a template for reading two integers a and b, and printing the product c = a x b.

```
#include <stdio.h>

void main() {

  int a, b, c;
  printf("Enter an integer:\n");
  scanf("%d", &a);
  printf("Enter an integer:\n");
  scanf("%d", &b);

  c = 0;  // product, initialized as 0.
```

```
    // your code goes here

    printf("the product = %d\n", c);

  }
```

The flow chart of Booth algorithm is given here. For this assignment, **n = 16**.


start
C ← 0; A sign extend to 2n-bit; B-1 = 0; i ← n
01
B0B-1
10
00
11
C = C + A
C = C -A
A <<1; BB-1 >>1
i= i-1
i= 0 ?
No
Yes
end