Stephen Corya
ECE 362
Pre-Lab #3

Introduction:
The purpose of this lab is to performing arithmetic operations on the HC(S)12 board. Learned concepts will include signed and unsigned numerical computation, as well as loops.

Lab 3.1.1:
Objective/Purpose:
The purpose of this experiment is to solve a linear equation, the result of which will be rounded to the nearest integer.

Expected Results:
Given an equation in slope-intercept form, with values with slope and intercept values provided, calculate ‘y’.

Code:
```
      XDEF  Entry
      XREF  __SEG_END_SSTACK

Variables:        Section
Val:              ds.b  0     the ‘x’ variable in y=mx+b

Constants:        Section
Slope:            dc.b  68    ;the integer value of slope ‘m’
Intercept         dc.b  12    ;the y-intercept value ‘b’

Code:             Section

Entry:
                  LDS         #__SEG_END_SSTACK

                  LDAA        Val         ;store variable ‘x’ in acc. A
                  LDAB        Slope       ;store the constant ‘m’ in acc. B
                  MUL                     ;multiply ‘x’ * ‘m’
                  ADDD        Intercept   ;add constant ‘b’
                  nop
```

Lab 3.1.2:
Objective/Purpose:
Similar to 3.1.1, we will calculate a result given an equation, for which all requisite variables will be defined.

Expected Results:
Given a parabolic equation, we will calculate its output ‘y’ given

Code:
```
      XDEF  Entry
      XREF  __SEG_END_SSTACK
      Variables:        Section
      Val:        ds.b  0     the ‘x’ variable in y=mx+b

      Constants:  Section
      Slope:      dc.b  68    ;the integer value of slope ‘m’
      Intercept   dc.b  12    ;the y-intercept value ‘b’

      Code:       Section
```

```
      Entry:
                  LDS         #__SEG_END_SSTACK

                  LDAA        Val         ;store variable ‘x’ in acc. A
                  LDAB        Val         ;store the same variable `x` in acc. B
                  MUL                     ;acc. A * acc. B => D = x^2
                  LDY         Slope       ;store ‘m’ in register Y
                  EMUL                    ;multiply D (x^2) x reg. Y (m)
                  ADDD        Intercept   ;add constant ‘b’
                  nop
```

Lab 3.2:

Objective/Purpose:
The purpose of this experiment is to test how sundry arithmetic operations affect the condition code register or CCR. We will then change the code and determine the value of accumulator A depending on the various branch instructions.

Expected Results:
Depending on input variables, the flags within the CCR should change.

Code:

```
      XDEF  Entry
      XREF  __SEG_END_SSTACK
      MyConstant:       section
      num_1:      dc.b  $40
      num_2:      db.b  $50
      MyCode:           section
      Entry:      ldaa  num_1
                  adda  num_2
                  nop
;--------------------------------------
      XDEF  Entry
      XREF  __SEG_END_SSTACK
                  ldaa  #$D3
                  adda  #$F2
                  bvs   done
                  ldaa  #0
            done: nop
;--------------------------------------
      XDEF  Entry
      XREF  __SEG_END_SSTACK
                  ldaa  #$D3
                  adda  #$F2
                  bcs   done
                  ldaa  #0
            done: nop
;--------------------------------------
      XDEF  Entry
      XREF  __SEG_END_SSTACK
                  ldaa  #$D3
                  adda  #$F2
                  bvs   done
                  ldaa  #0
            done: nop
;--------------------------------------
      XDEF  Entry
      XREF  __SEG_END_SSTACK
                  ldaa  #$41
```

```
            adda  #$5A
            bvs   done
            ldaa  #0
      done: nop
```

Lab 3.3:

Objective/Purpose:
We will write a program in assembly language to turn a stepper motor, connection to Port P. We will learn how to implement a delay using a loop.

Expected Results:
The assembly loop should repeat and the motor should spin clockwise.

Code:

```
      XDEF  Entry
      XREF  __SEG_END_SSTACK
            MACROS:
            spin: macro             ;move the stepper motor clockwise turn
                  movb  #$0A,DDRP   ;1/4 turn clockwise
                  movb  #$12,DDRP   ;1/4 turn clockwise
                  movb  #$14,DDRP   ;1/4 turn clockwise
                  movb  #$0C,DDRP   ;1/4 turn clockwise
                  endm


            LOOP: equ   $8000       ;set the location of the program
                  spin              ;run the macro that turns the motor
                  bra   LOOP        ;return to the beginning of the program
```