CPEN 211 Introduction to Microcomputers, 2021
Lab Proficiency Test #2

Question 1 [3 marks, part marks possible]: Create a file named "**q1.s**" and inside it write ARM assembly to implement the C code for function `func` below. Assume that `i`, `j`, and `k` are 32-bit signed integers stored in registers `R0`, `R1`, and `R2` respectively, and that the base of array A is in `R3`.

```
int func(int i, int j, int k, int *A) {
  if( i < j ) {
    A[0] = 1;
  }
  if( i == k ) {
    A[1] = 2;
    if( A[2] > j ) {
      A[3] = 4;
    }
  }
  return i+j;
}
```

The autograder requires the result returned by `func` be in `R0` after your code executes. Your q1.s must contain the ARM code below where you must replace the comment "**`// ADD YOUR CODE HERE`**" with ARM code for `func`. Ensure the ARM code you add does not modify `R13` or `R14`. Your ARM code should work with any values of `i`, `j`, `k` and any array `A`, not just the values used in the ARM code below. To test your code, you may change the inputs to `func` by modifying the values in `R0` to `R3` by changing the lines before "`BL func`" and/or changing the array "`data`" in `q1.s`. Ignore warnings about "Function clobbered registers(s)" in the online simulator. Your solution for Question 1 will get zero if any of the following are true: (1) Your **last** "Lab Proficiency Test #2" attempt on Canvas does not include "`q1.s`"; (2) Your "`q1.s`" file does not compile with the Monitor Program configured to use the DE1-SoC Computer or the online simulator: https://cpulator.01xz.net/?sys=arm-de1soc

```
.global _start
_start:
  MOV R0, #1    // i=1
  MOV R1, #2    // j=2
  MOV R2, #1    // k=1
  LDR R3, =data // set base of A = first address of array "data"
  BL func
END: B END  // infinite loop; R0 should contain return value of func

.global func
func:
  // ADD YOUR CODE HERE
  MOV PC, LR

data:
  .word 0
  .word 0
  .word 3
  .word 0
```

CPEN 211 Introduction to Microcomputers, 2021
Lab Proficiency Test #2

Question 2 [2 marks, part marks possible]: Create a file named "**q2.s**" and inside it write ARM assembly to implement the C code for function loopy below. Assume that n is a 32-bit signed integer stored in register R0, that the base of array A is in R1, and that the base of array B is in R2.

```
int loopy(int n, int *A, int *B) {
  int L1norm=0;
  int i=0;
  while( i < n ) {
    int tmp = A[i];
    if( tmp < 0 ) {
      tmp = -tmp;
    }
    B[i] = tmp;
    L1norm = L1norm + tmp;
    i = i + 1;
  }
  return L1norm;
}
```

The autograder requires the result returned by loopy be in R0 after your code executes. Your q2.s must contain the ARM code below where you must replace comment "**// ADD YOUR CODE HERE**" with ARM code for loopy. Ensure the ARM code you add does not modify R13 or R14. Your ARM code should work with any value of n in R0 and any arrays A and B of length n input using R1 and R2. You may ignore "Function clobbered register(s)" warnings in the online simulator. To test your code, you may modify the values placed in R0 through R2 by changing the lines before "BL loopy" and/or changing arrays "input" and "output" in q2.s. Your **last** "Lab Proficiency Test #2" attempt must include "q2.s" and "q2.s" must compile.

```
.global _start
_start:
  MOV R0, #2     // n=2
  LDR R1, =input // base of A = first address of array "input"
  LDR R2, =output// base of B = first address of array "output"
  BL loopy
END: B END // infinite loop; R0 should contain return value of loopy

.global loopy
loopy:
  // ADD YOUR CODE HERE
  MOV PC, LR

input:
  .word -1
  .word 1
output:
  .word 0
  .word 0
```