

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
```