Score in 35 minutes: 50

Time to get to 100: 48 minutes

Code is below

```
/*
;*****Andrew Kirk*******
; -5 points if you do not add your name

;This is Exam2_StringCompare
;EE319K Practice exam
;You edit this file only
*/
#include <stdint.h>
/*
;**************** Size***************************
; Determines the length of an ASCII string.
; Input parameter: buffer points to null-terminated string
; Output parameter: Return the length of string
; Error conditions: if string is empty, return 0
;Invariables: Y
;Test cases
;1. buffer "cat"                    ;size=3
;2. buffer "Ramesh is great."       ;size=16
;3. buffer "EE319K Exam2 was hard!" ;size=22
;4. buffer "My TA is nice,"         ;size=14
;5. buffer ""                       ;size=0
*/
uint32_t Size(const uint8_t *buffer){
```

```
// put your code here
uint8_t size=0;
        while(*buffer!=0){
                size=size+1;
                buffer=buffer+1;
                }



 return(size);  // change this line
}


/*
;***************** Compare****************************
; Compare two ASCII characters
; Input parameter:  first and
;             second are 8-bit ASCII character.
; Output parameter: Return -1 if the first is greater than the second,
;             Return 0 if the first equals the second, and
;             Return +1 if the first is less than the second
; Error conditions: none


;Invariables:
;Error conditions: none
;Test cases
;1    first='a', second='b'       ;Compare returns +1 because 'a' < 'b'
;2    first='a', second='B'        ;Compare returns -1 because 'A' > 'b'
;3    first='a', second='a'       ;Compare returns 0  because 'a' == 'a'
;4    first=200, second=199       ;Compare returns -1 because 200 > 199
```

```
;5    first=200, second=201          ;Compare returns 1 because 200 < 201
*/
int32_t Compare(uint8_t first, uint8_t second){
// put your code here
int8_t answer;
        if (first>second){
                answer=-1;
        }
        else if (first==second){
                answer=0;
        }
        else {
                answer=1;
        }

        return(answer);  // change this line

}

/*
;*********StringCompare*******************************
; Compares two ASCII strings, null-terminated(there is a 0 at the end of the string)
; Input parameter: A pointer to the first string is passed into your program in R0.
;                A pointer to the second string is passed into your program in R1.
; Output parameter: The result is returned as follows:
;    +1 if the first string is alphabetically before the second
;     0 if the two strings are equal
;    -1 if the first string is alphabetically after the second
; Error conditions: none
;Invariables:
;Test cases
;1. buffer1  "cat"     ; +1 because first letter 'c'< 'd'
;   buffer2  "dog"
;2. buffer1  "cattle"  ; +1 because second letter 'a'< 'o'
;   buffer2  "cobra"   ;(length doesn't matter)
;3. buffer1  "hose"    ; -1 because third letter 's'> 'r'
;   buffer2  "horse"   ; (length doesn't matter)
;4. buffer1  "cat"     ; +1 because all letters of the first string match,
;   buffer2  "cattle"  ; but the first string is shorter (length does matter)
;5. buffer1  "cattle"  ;-1 because all letters of the second string match,
;   buffer2  "cat"     ; but the second string is shorter (length does matter)
;6. buffer1  "horse"
;   buffer2  "horse"   ; 0 because the strings are equal
;7. buffer1  ""
;   buffer2  ""        ; 0 because the strings are equal and empty
*/

int32_t StringCompare(const uint8_t *buffer1,const uint8_t *buffer2){
        // put your code here
 int8_t i, answer1, size1=0, size2=0;
        while(*buffer1!=0){
                size1=size1+1;
                buffer1=buffer1+1;
                }
        while(*buffer2!=0){
                size2=size2+1;
```

```
    buffer2=buffer2+1;
    }
buffer1=buffer1-size1;
buffer2=buffer2-size2;

    if (size1>size2){
                for(i=0; i<size1; i++){
                        if (*buffer2==0){
                            answer1=-1;
                                break;
                                    }

                        else if (*buffer1>*buffer2){
                                answer1=-1;
                                break;
                                }
                        else if (*buffer1==*buffer2){
                                    answer1=0;
                             buffer1 ++;
                             buffer2 ++;
                                }

                        else {
                                    answer1=1;
                            break;
                            }
                    }
                }
```

```
else if (size1<size2){
    for(i=0; i<size2; i++){
        if (*buffer1==0){
     answer1=1;
     break;
            }

        else if (*buffer1>*buffer2){
                answer1=-1;
                break;
                }
        else if (*buffer1==*buffer2){
             buffer1 ++;
                    buffer2 ++;
                    answer1=0;
                }

        else {
                    answer1=1;
            break;
                    }
                }
            }

else if(size1==0){
    answer1=0;
}
```

```
   else {
         for(i=0; i<size2; i++){
         if (*buffer1==0){
       answer1=1;
       break;
            }

         else if (*buffer1>*buffer2){
               answer1=-1;
               break;
               }
         else if (*buffer1==*buffer2){
             buffer1 ++;
                  buffer2 ++;
                  answer1=0;
               }

         else {
                  answer1=1;
            break;
                  }
               }
            }

return(answer1);  // change this line
```

}