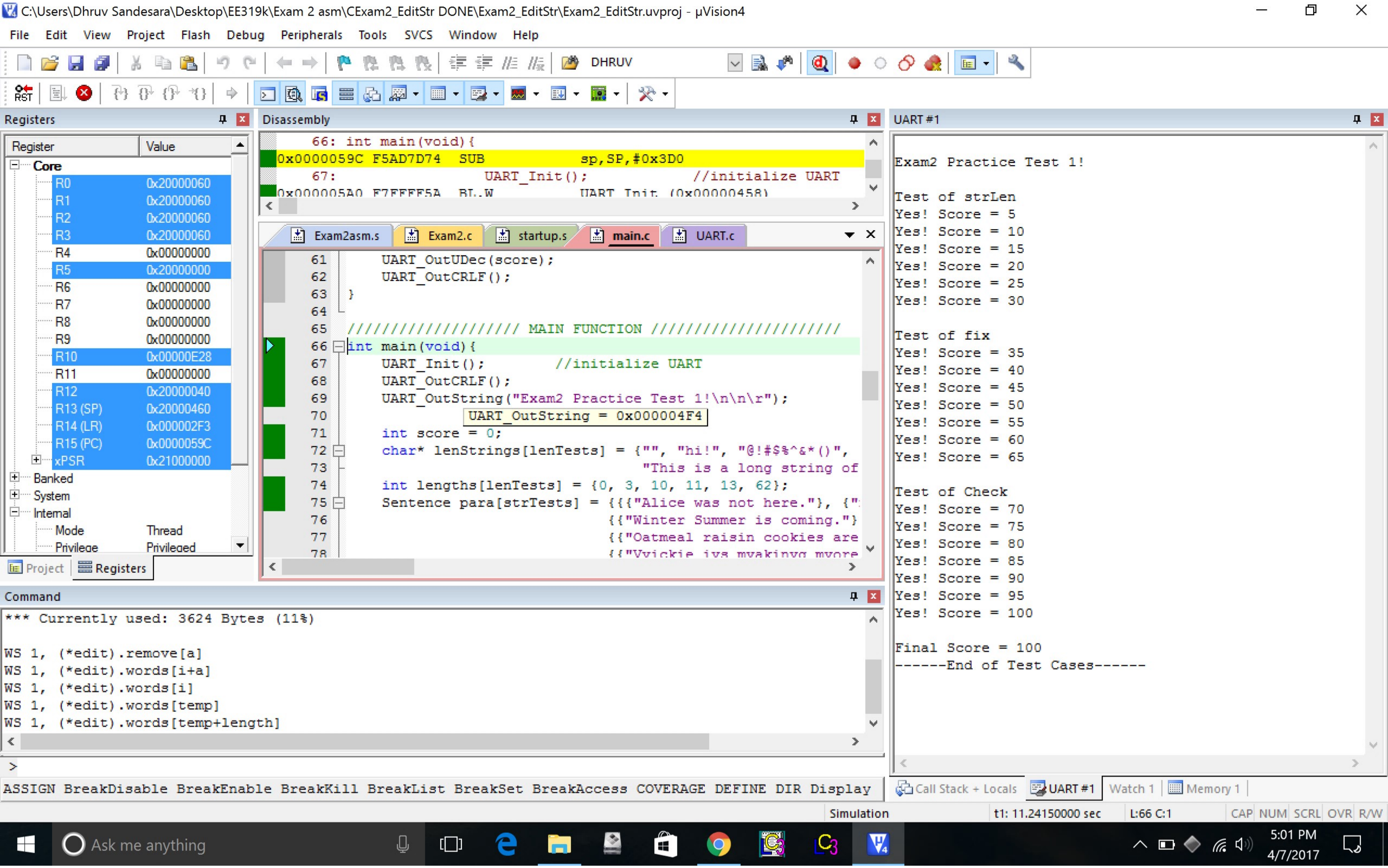
C:\Users\Dhruv Sandesara\Desktop\EE319k\Exam 2 asm\CExam2_EditStr DONE\Exam2_EditStr\Exam2_EditStr.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
DHRUV
RST
Registers
Disassembly
UART #1
Exam2asm.s
Exam2.c
startup.s
main.c
UART.c
Project
Registers
Command
Call Stack + Locals
UART #1
Watch 1
Memory 1
Ask me anything

Register	Value
Core	
R0	0x20000060
R1	0x20000060
R2	0x20000060
R3	0x20000060
R4	0x00000000
R5	0x20000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000E28
R11	0x00000000
R12	0x20000040
R13 (SP)	0x20000460
R14 (LR)	0x000002F3
R15 (PC)	0x0000059C
xPSR	0x21000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged

```
    66: int main(void){
0x0000059C F5AD7D74  SUB          sp,SP,#0x3D0
    67:              UART_Init();          //initialize UART
0x000005A0 F7FFFF5A  BL.W         UART_Init (0x00000458)
```

```
    UART_OutUDec(score);
    UART_OutCRLF();
}

/////////////////// MAIN FUNCTION //////////////////////
int main(void){
    UART_Init();        //initialize UART
    UART_OutCRLF();
    UART_OutString("Exam2 Practice Test 1!\n\n\r");
                   UART_OutString = 0x000004F4
    int score = 0;
    char* lenStrings[lenTests] = {"", "hi!", "@!#$%^&*()",
                                  "This is a long string of
    int lengths[lenTests] = {0, 3, 10, 11, 13, 62};
    Sentence para[strTests] = {{{"Alice was not here."}, {":
                              {{"Winter Summer is coming."}
                              {{"Oatmeal raisin cookies are
                              {{"Vvickie ivs mvakinvg mvore
```

```
*** Currently used: 3624 Bytes (11%)

WS 1, (*edit).remove[a]
WS 1, (*edit).words[i+a]
WS 1, (*edit).words[i]
WS 1, (*edit).words[temp]
WS 1, (*edit).words[temp+length]
```

```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display
```

```

Exam2 Practice Test 1!

Test of strLen
Yes! Score = 5
Yes! Score = 10
Yes! Score = 15
Yes! Score = 20
Yes! Score = 25
Yes! Score = 30

Test of fix
Yes! Score = 35
Yes! Score = 40
Yes! Score = 45
Yes! Score = 50
Yes! Score = 55
Yes! Score = 60
Yes! Score = 65

Test of Check
Yes! Score = 70
Yes! Score = 75
Yes! Score = 80
Yes! Score = 85
Yes! Score = 90
Yes! Score = 95
Yes! Score = 100

Final Score = 100
------End of Test Cases------
```

Simulation t1: 11.24150000 sec L:66 C:1 CAP NUM SCRL OVR R/W

5:01 PM
4/7/2017

```
; Exam2asm.s
; YOUR NAME:
; DHRUV SANDESARA
;
;   '-> DO NOT FORGET TO PUT YOUR NAME IN THE
;       CODE ON THE EXAM OR IT WILL BE -5 POINTS!!!!
; Date Modified:
    EXPORT   strLen
    EXPORT   Check

    AREA    |.text|, CODE, READONLY, ALIGN=2
    PRESERVE8
    THUMB

        ; EDIT THIS FILE!
;-------------------------- INSTRUCTIONS ------------------------
; Implement the following functions strLen and Check
; The grader will display in UART1 when you run your code in the debugger

;--------------- strLen ---------------
; Finds the length of a char string
; INPUT: R0 is pointer to a null-terminated char string
; OUTPUT: the number of chars in string
; TESTCASES: (there are 6 testcases for strLen)
;   ""              Should Return: 0
;   "hi!"           Should Return: 3
;   "@!#$%^&*()"    Should Return: 10
;   "trick 0 you"   Should Return: 11
;   "funny/b/b  /n" Should Return: 13
;   "This is a long string of words to test your strLen function :)" Should Return: 62

strLen
    PUSH {R1-R4}
    MOV R1,#0;
    MOV R2,#0;
STRLOOP
    LDRB R1,[R0]
    CMP R1,#0;
    BEQ STRDONE;
    ADD R2,#1;
    ADD R0,#1;
    B STRLOOP

STRDONE
    MOV R0,R2;

    POP {R1-R4}
    BX LR
    ; your code here

;--------------- Check ---------------
; Function checks if two char strings are equal
; INPUT: R0 holds a pointer to the first null-terminated char string
;        R1 holds a pointer to the second null-terminated char string
; OUTPUT: 1 if strings are equal
;         0 if strings are not equal
; TESTCASES: (there are 7 testcases for Check)
;   (1) Should Output: 1    ""
;                           ""
;   (2) Should Output: 1    "The University of Texas at Austin"
;                           "The University of Texas at Austin"
;   (3) Should Output: 1    "123456789"
;                           "123456789"
;   (4) Should Output: 0    "this is same"
;                           "this is not same"
;   (5) Should Output: 0    "what if there is no space"
;                           "whatifthereisnospace"
;   (6) Should Output: 0    "how about a space at the end"
;                           "how about a space at the end "
;   (7) Should Output: 1    "!@#$%^ ^ &*()__--"
;                           "!@#$%^ ^ &*()__--"

```

```
Check
    ;your code here
    PUSH {R2-R8,LR}
CHECKLOOP
    LDRB R2,[R0]
    LDRB R3,[R1]
    CMP R2,R3;
    BNE NOTEQUAL
    CMP R2,#0;
    BEQ EQUAL;
    ADD R0,#1;
    ADD R1,#1;
    B CHECKLOOP;


EQUAL
    MOV R0,#1;
    B CHECKDONE
NOTEQUAL
    MOV R0,#0;
CHECKDONE
    POP {R2-R8,LR}
    BX LR

;------------------------------------------------------
; End of file Exam2asm.s
    ALIGN
    END
```

```
//Exam2.c
//YOUR NAME:
//DHRUV SANDESARA
//   '-> DO NOT FORGET TO PUT YOUR NAME IN THE
//       CODE ON THE EXAM OR IT WILL BE -5 POINTS!!!!
//Date Modified:

#include <stdint.h>
#include "Exam2.h"
int strLen (char* string); //prototype for ASM function

        // EDIT THIS FILE!
/***************************************************
 * INSTRUCTIONS:
 *    Implement the following function fix
 *    The grader will display in UART1 when
 *      you run your code in the debugger
 **************************************************
 * NOTE: the sentence struct looks like this:
 *       (the struct is actually defined in Exam2.h)
 *  typedef struct sentence{
 *      char words[100];  //NULL-terminated string of words (a sentence)
 *      char remove[10];  //NULL-terminated string to remove from sentence
 *      int num_removed;  //number of edits, has -1 if nothing has been removed yet
 *  } Sentence;
***************************************************/

/********************** fix **********************
 * Removes all instances of the remove string from the words string
 * Function updates the words string in struct
 *          does not change the remove string
 *          stores the number of edits in num_removed
 * INPUT: Sentence*  A pointer to a sentence structure (defined in comments above)
 * OUTPUT: void      BUT words string is updated with the edited string
 *                   num_removed has the number of times remove was taken out of words */
 void fix(Sentence* edit){
   uint32_t i=0,a=0,match=0,temp=0,temp2=0;;
   int32_t length=0;



   while (((*edit).words[i]!=0)&&(i<100))
     {
        while (((*edit).remove[a]!=0)&&(a<10)){
          match=1;
          if((*edit).remove[a]!=(*edit).words[i+a]){
            match=0;
            break;
          }
            length++;
          a++;

        }



        if(match){
          temp=i;
          while(((*edit).words[temp]!=0)){


            (*edit).words[temp]= (*edit).words[temp+length];

            temp++;

          }
          temp2++;


          }
          else{
            i++;
          }
      match=0;
      a=0;
      length=0;

   }


  (*edit).num_removed= temp2;
```

```c


    // your code here
}

/* fix TESTCASES: (there are 7 testcases for fix)
 *  for each test, row one is the struct at input
 *                row two is what the struct should look like when you return
 *    (1) {{"Alice was not here."}, {"not "}, -1}
 *        {{"Alice was here."},     {"not "},  1}
 *    (2) {{"Winter Summer is coming."}, {"Winter "}, -1}
 *        {{"Summer is coming."},        {"Winter "},  1}
 *    (3) {{"Oatmeal raisin cookies are the bestest."}, {"est."}, -1}
 *        {{"Oatmeal raisin cookies are the best"},      {"est."},  1}
 *    (4) {{"Vvickie ivs mvakinvg mvore vvvtest cvvasvvesv..v."}, {"v"}, -1}
 *        {{"Vickie is making more test cases..."},                {"v"}, 14}
 *    (5) {{"Puppies"}, {""}, -1}
 *        {{"Puppies"}, {""}, 0}
 *    (6) {{""}, {"a"}, -1}
 *        {{""}, {"a"},  0}
 *    (7) {{"SOSOk SO SOSthiSOSs iS SOSthe laSOst oneS0S!SOS"}, {"SOS"}, -1}
 *        {{"Ok SO this iS the laSOst oneS0S!"},               {"SOS"},  5}
 */


```