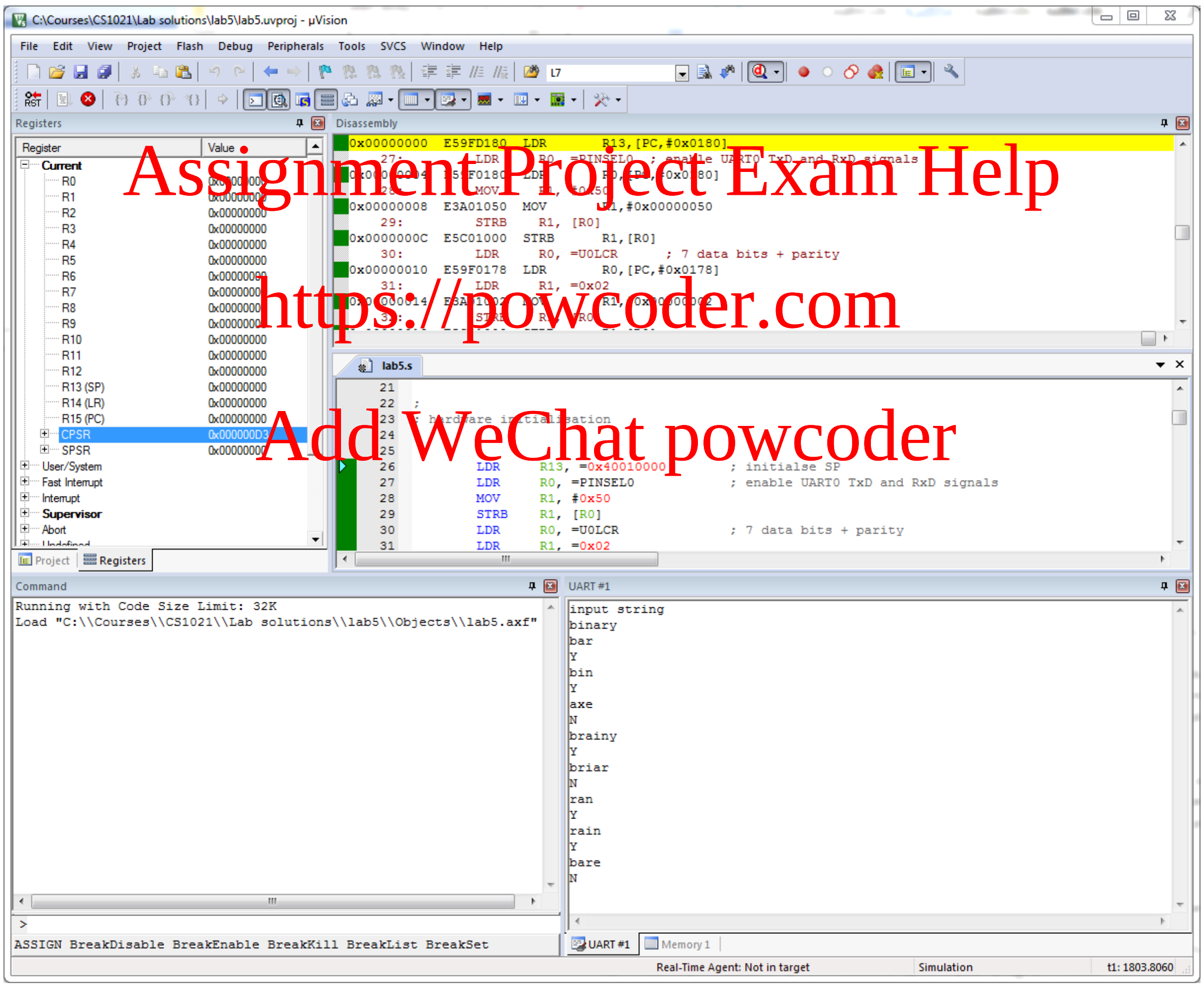
CS1021 Lab 5

Anagrams

This is an individual Lab and will be marked out of 20 (Lab1 to Lab4 were marked out of 10).

Objective

The objective of lab5 is to allow users enter a string s0 in the uVision console window followed by a number strings which are each checked if they are anagrams of s0. A sample screenshot from the program is shown below. The console (labelled UART #1) shows that the user has entered "binary" for s0 followed by "bar", "bin", "axe", "brainy", "briar", "ran", "rain" and "bare" which are each checked if they are anagrams of s0. The program output "Y" or "N" after each string.


C:\Courses\CS1021\Lab solutions\lab5\lab5.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
L7
Registers
Disassembly
lab5.s
Project
Registers
Command
UART #1
Memory 1
Real-Time Agent: Not in target
Simulation
t1: 1803.8060

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	

```
0x00000000  E59FD180  LDR      R13,[PC,#0x0180]
    27:        LDR     R0, =PINSEL0   ; enable UART0 TxD and RxD signals
0x00000004  E59F0180  LDR      R0,[PC,#0x0180]
    28:        MOV     R1, #0x50
0x00000008  E3A01050  MOV      R1,#0x00000050
    29:        STRB    R1, [R0]
0x0000000C  E5C01000  STRB     R1,[R0]
    30:        LDR     R0, =U0LCR      ; 7 data bits + parity
0x00000010  E59F0178  LDR      R0,[PC,#0x0178]
    31:        LDR     R1, =0x02
0x00000014  E3A01002  MOV      R1,#0x00000002
    32:        STRB    R1, [R0]
```

```
;
; hardware initialisation
;

        LDR     R13, =0x40010000        ; initialse SP
        LDR     R0, =PINSEL0            ; enable UART0 TxD and RxD signals
        MOV     R1, #0x50
        STRB    R1, [R0]
        LDR     R0, =U0LCR              ; 7 data bits + parity
        LDR     R1, =0x02
```

```
Running with Code Size Limit: 32K
Load "C:\\Courses\\CS1021\\Lab solutions\\lab5\\Objects\\lab5.axf"

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
input string
binary
bar
Y
bin
Y
axe
N
brainy
Y
briar
N
ran
Y
rain
Y
bare
N
```

Test your program with the following strings:

binary - bar bin axe brainy briar ran rain bare
astronomer - moon moonstarer toast storms rooms axe smarter streams

Console Input Output

Lab5.s contains code to initialise the hardware, code for subroutines GET, PUT and PUTS and a simple "main" program that demonstrates how to (i) write a zero terminated ASCII string to the console window (ii) read ASCII character typed in the console window and (iii) write an ASCII character to the console window.

```
;
; uses PUTS to write "ECHO ECho echo ..." to the console
; echo typed characters (CTRL j for a line feed)
;
        LDR     R0, =STR0                       ; R0 -> "ECHO ECho echo ..."
        BL      PUTS                            ; put ASCII string
L       BL      GET                             ; get ASCII char
        BL      PUT                             ; put ASCII char
        B       L                               ; forever

STR0  DCB       "ECHO ECho echo...", 0x0a, 0, 0 ; zero (NUL) terminated string
```

A zero terminated ASCII string is written to the console by executing BL PUTS where R0 contains the address of the ASCII string. A single ASCII character is read by executing BL GET. This instruction only completes/returns when the user types a character in the console window. The ASCII character code corresponding to the key is returned in R0. The typed key by the user is not automatically displayed in the console window so it must be "echoed" by executing BL PUT which writes the ASCII character contained in R0 to the console.

Example: If BL GET is executed and the user types '7' on the keyboard, then its ASCII code 0x37 is returned in R0. If BL PUT is executed (with 0x37 in R0) then '7' is displayed on the console. BL is short for "branch and link".

Executing BL GET or BL PUT or BL PUTS may overwrite the contents of R0 - R3, so these registers should not be used to store values that are needed after these instructions are executed (also avoid using R13 - R15).

Using the µVision Console

To display the console window in Debug mode, select the menu option [View][Serial Windows][UART #1]. Make sure you can get the initial code provided for lab5 working.

When testing your program, you can either step through the program one instruction at a time or use F5 to run the program without stopping or until the next breakpoint is reached. If you are stepping through one instruction at a time, after executing BL GET you will need to mouse-click in the "UART #1" console window and press a key. Your program will then continue to the next instruction.

Suggested Approach

Start by write writing pseudo-code for the program. Example pseudo-code is shown below:

```
L:
    PUTS("input string\n");
    s0 = readString();
    if (isEmpty(s0))
    goto L;
while(1) {
    s1 = readString();
    if (isEmpty(s1))
      goto L;
    if (isAnagram(s0, s1)) {
      PUTS("Y\n");
    } else {
      PUTS("N\n");
    }
}
```

Break the program in to parts and implement and test each part one by one. Start by writing code for readString() and use it to read in strings s0 and s1 and store them in memory. Next, write the code to output the result ("Y" or "N"). Finally write the code for testing if s1 is an anagram of s0. Read a string by reading characters until the enter key is pressed (ASCII 0x0D). Output a new line character by writing a line feed (ASCII 0x0A). The only complication with testing if s1 is an anagram of s0 is the need to check that each character in s0 is used only once – this is why "briar" is not an anagram of "binary" as "binary" has only one "r".

Extra Mile

Additional features you may consider adding.

- Convert uppercase letters to lowercase (or vice versa) so the "bar" would be anagram of "BINARY".

- Handle backspace and delete characters

- Allow the strings to contain spaces so that it possible to check that "I am not active" is an anagram of "Vacation time"

Submission

This lab will count towards your final CS1021 coursework mark. Submit your solution via Blackboard no later than **9am on Fri 16-Nov-2018**. You should submit lab5.s and one .pdf file which gives a brief summary of your work and a clear statement as to whether your code works or not. The .pdf file must include evidence that your program works (e.g. use the Windows Snipping tool to get screen shots of the Keil IDE which show that your program generates the correct results).