

4. The bsearch function searches an array for a given value by splitting the array in half each time and searching within that subsection of the array. This search algorithm only works if the array is sorted from least to greatest.

bsearch: CMP R1, R2
BEQ true

R1 = i

R2 = j

R3 = v

R4 = h

ADD R5, R1, R2

LSR R4, R5

LD R6 [R4]

CMP R6 R3

BEQ return h

BGT bsearch1

BLE bsearch2

return h: MOV R0, R4

POP {LR}

BX LR

bsearch1: MOV R2, R4

BL bsearch

bsearch2: ADD R4, R4, #1

BL bsearch

true: LD R6 [R1]

CMP R6 R3

BEQ send i

MOV R0, #-1

POP {LR}

BX LR

send i: MOV R0, R1

POP {LR}

BX LR

ECE 3140

Eric Berg (eb645)

Problem Set 1

Feb. 24, 2017

3. a) mistakes

- missing "j" after $f_{n-1} = 1$ statement \rightarrow COMPILER ERROR
- missing $\{ \}$ around 3 lines within for loop \rightarrow NO COMPILER ERROR
- $\text{int } f_{n-2};$ statement has no assignment but is added with later \rightarrow COMPILER ERROR
- missing $\#include$ statement for function to be referenced \rightarrow NO ERROR elsewhere

b) mylib.h
 $\#ifndef$ MYLIB
 $\#define$ MYLIB

main.c
 $\#include$ "mylib.h"

mylib.c
 $\#include$ "mylib.h"
 $\text{int fib}(\text{int } n) \{$

$\text{int fib}(\text{int } n);$

$\#endif$

c) 102334155 $\xrightarrow{\text{hexidecimal}}$ 06197ECB
 (network) pad to make even
Big Endian

small address	06
	19
	7E
big address	CB

(local)
Little Endian

small address	CB	CB7E1906
	7E	11
	19	3419038790
big address	06	value computed by local computer (incorrectly)

d) $\text{for}(i=0; i<4; ++i) \{$

$\text{bytes_of_a}[i] = \text{bytes_of_b}[(\text{bytes_of_b.length}-1)-i];$

$\}$

d) AND R0, PC, #0x0F see A 7.7.8
 Rd Rn immediate

This instruction is not legal because the manual states that the Rn register cannot be R13-R15, so because the Rn = PC = R15 the operation is UNPREDICTABLE.

2. a) PC: R15
 LR: R14
 SP: R13
 arguments passed to function: R0-R3
 return value: R0

b) Stack after line 7

	High Address	
	...	
POPPED	5	// i5 } caller saved
POPPED	10	// i5 }
	2	// R4 } callee saved
	128	// R5 }
	1024	// R6 }
	LR LR	// LR
	3	// add 12
	12	// mul 34
	2	// div 56

Registers At end of function

R0	17
R1	2
R2	3
R3	4
R4	3
R5	12
R6	2

c) A double word argument can be passed to a function by passing a pointer to the first address where the first word is stored in memory

ECE 3140

Eric Berg

Problem Set 1

Feb. 24, 2017

1. a)

```

loop: CMP R2, #0
      BLT end
      LD R4, [R0]
      LSL R5, R4 // multiply x2
      LSL R6, R5 // multiply x2
      STR R6, [R1] // load output array
      SUB R2, R2, #1
      ADD R1, R1, #4 // skip to next array entry
      ADD R0, R0, #4 // skip to next array entry
      BL loop
    
```

R0 - 1st element input array
 R1 - 1st element output array
 R2 - # of elements in array

end: WFI

b) 10 lines \rightarrow 32 bits per instruction / 8 bits per byte
 $(10 \cdot 32) / 8 = 40 \text{ bytes}$

c) BL multby4.
 multby4: PUSH {LR}
 PUSH {R4-R11}
 loop: ~~~~~
 ~~~~~  
 ~~~~~  
 ~~~~~  
 ~~~~~  
 POP {R4-R11}
 POP {PC}