

Student #:

Name:

## Quiz #2 2018 (5 min.)

Note 1: **Return** the answer sheet (even the blank one) to show your attendance.

Consider a C application program and its disassembly view below.

```
func_time = (float)(xstop - xstart) / 333;
printf("%f us\n", func_time);

return flag;

}

int memtest_1()
{
    int i, flag=-1;
    short *Addr;

    short Pattern0;
    short Pattern1;

    print ("Test 1 :");

    XTime_GetTime(&xstart);

    // Fill your code here! //
    flag = 0;
    Addr = 0xffff0000;

    Pattern0 = ①;
    Pattern1 = ②;

    // Memory write
    for (i=0; i<1024;i++)
    {
        Addr[2*i+0] = Pattern0;
        Addr[2*i+1] = Pattern1;
    }

    // Memory Read & Check
    for (i=0; i<1024;i++)
    {
        if (Addr[2*i+0] != Pattern0 || Addr[2*i+1] != Pattern1)
        {
            flag = -1;
            break;
        }
    }

    XTime_GetTime(&xstop);
    func_time = (float)(xstop - xstart) / 333;
    printf("%f us\n", func_time);

    return flag;
}
```

```
00100700: mov r6, #0
00100704: b 0x1006a0 <memtest_0+116>
00100708: ; <UNDEFINED> instruction: 0x43a68000

memtest_1:
0010070c: push {r4, r5, r6, lr}
00100710: movw r0, #47340 ; 0xb8ec
00100714: movt r0, #16
00100718: bl 0x1008cc <print>
0010071c: movw r0, #16512 ; 0x4080
00100720: movt r0, #17
00100724: bl 0x1014f0 <XTime_GetTime>
00100728: mov r3, #0
0010072c: movw r2, #43690 ; 0xaaaa
00100730: movt r3, #65535 ; 0xffff
00100734: movt r2, #65535 ; 0xffff
00100738: movw r1, #21845 ; 0x5555
0010073c: strh r1, [r3]
00100740: add r3, r3, #4
00100744: cmn r3, #61440 ; 0xf000
00100748: strh r2, [r3, #-2]
0010074c: bne 0x10073c <memtest_1+48>
00100750: mov r3, #0
00100754: movw r1, #43690 ; 0xaaaa
00100758: movt r3, #65535 ; 0xffff
0010075c: movt r1, #65535 ; 0xffff
00100760: movw r0, #21845 ; 0x5555
00100764: b 0x100780 <memtest_1+116>
00100768: ldrsh r2, [r3, #2]
0010076c: add r3, r3, #4
00100770: cmp r2, r1
00100774: bne 0x10078c <memtest_1+128>
00100778: cmn r3, #61440 ; 0xf000
0010077c: beq 0x1007f0 <memtest_1+228>
00100780: ldrsh r2, [r3]
00100784: cmp r2, r0
00100788: beq 0x100768 <memtest_1+92>
0010078c: mvn r6, #0
00100790: movw r4, #16504 ; 0x4078
00100794: movt r4, #17
00100798: mov r0, r4
0010079c: bl 0x1014f0 <XTime_GetTime>
001007a0: movw r3, #16512 ; 0x4080
001007a4: movt r3, #17
001007a8: ldrd r4, [r4]
001007ac: ldrd r0, [r3]
001007b0: subs r0, r4, r0
001007b4: sbc r1, r5, r1
```

Handwritten notes on the right side of the disassembly:

- r3: 0x FFF\_0000
- r2: 0x FFF\_0000
- r1: 0x0000\_5555
- Addr[0]: 0x5555
- Addr[2]: 0x5555
- Addr[1]: 0xAAAA
- Addr[3]: 0xAAAA
- r3: 0x FFF\_0004
- r3: 0x FFF\_000B

(1) Fill out the blanks ① and ② in the above C application program [5pt].

① 0x5555

② 0xAAAA

(2) Provide **all** the values (e.g., 0x0000\_0000) of the register/memory locations below assuming that the program currently breaks at ③ (before executing the instruction of that line) **for the second time** [15pt].

## Before Running

Memory		Register File	
0xFFFF_0000	0x0000 <sub>2</sub> 0x0000 <sub>0</sub>	r1	0x0000_0000
0xFFFF_0004	0x0000 <sub>6</sub> 0x0000 <sub>4</sub>	r2	0x0000_0000
	⋮	r3	0x0000_0000
0xFFFF_0FFC	0x0000 <sub>E</sub> 0x0000 <sub>C</sub>		

## After Running

Memory		Register File	
0xFFFF_0000	0xAAAA <sub>2</sub> 0x5555 <sub>0</sub>	r1	0x0000_5555
0xFFFF_0004	0x0000 <sub>6</sub> 0x5555 <sub>4</sub>	r2	0xFFFF_AAAA
	⋮	r3	0xFFFF_000B
0xFFFF_0FFC	0x0000 <sub>E</sub> 0x0000 <sub>C</sub>		