

Last NAME:

Quiz 2 , March 23, 2022

First Name:

Computer Science C.Sc. 342

Quiz No.2 To be performed

5:00-6:15 PM on March 23, 2022

Submit by 6:15 PM 03/23/2022 on Slack to Instructor

Please write your Last Name on every page:

NO CORRECTIONS ARE ALLOWED IN ANSWER CELLS!!!!

You may use the back page for computations.

Please answer all questions. **Not all questions are of equal difficulty.**

Please review the entire quiz first and then budget your time carefully.

Please hand write and sign statements affirming that you will not cheat:

"I will neither give nor receive unauthorized assistance on this exam.

I will use only one computing device to perform this test"

Please hand write and sign here:

This quiz has 6 pages.

Question	Your Grade	Max Grade
1.1		5
1.2		10
1.3		10
1.4		10
2.1.1		15
2.1.2		15
2.1.3		15
2.2.1		5
2.2.2		5
2.2.3		5
2.3		5

Total: 100

Last NAME:

Quiz 2 , March 23, 20222

First Name:

Question 1.

A student, while debugging his program, unintentionally displayed partially corrupted DISSASSEMBLY windows in MS Visual Studio Debug environment.

He was able to display correctly Register window, and two Memory windows.

His task was to determine addresses of variables in the expression **result = LocalInt + StatInt** in Memory at the instance of the snapshot.

He is not allowed to restart the debug session.

Can you help him to answer the following questions:

The screenshot displays the Visual Studio Debug environment with three windows:

- Assembly Window:** Shows the disassembly of the `main` function. The code includes static variables `result` and `StatInt`, and a `main` function that initializes `StatInt` to -7, `LocalInt` to 2, and then attempts to calculate `result = LocalInt + StatInt`. The instruction at address `00DF1793` is highlighted.
- Memory 2 Window:** Displays memory addresses from `0x00CFF81B` to `0x00CFF836`. The values are mostly `cc` (corrupted), except for `0x00CFF828` which contains `f9` (hex for -7).
- Memory 1 Window:** Displays memory addresses from `0x00DFA177` to `0x00DFA180`. The values are mostly `ff` (corrupted), except for `0x00DFA17C` which contains `02` (hex for 2).
- Registers Window:** Shows the current state of registers: `EAX = 00000000`, `EBX = 00B6C000`, `ECX = 00DFC000`, `EDX = 00000001`, `ESI = 00DF1023`, `EDI = 00CFF830`, `EIP = 00DF1793` (highlighted), `ESP = 00CFF74C`, and `EBP = 00CFF830`.

Last NAME:

Quiz 2 , March 23, 20222

First Name:

1.1 [5 points] What is the address of the instruction that will be executed next instance?

1.2 [10 points] Can you determine the address of variable **StatInt** in the expression? **YES or NO.**

Please circle around your answer. IF No is your answer, then go to the next question

ELSE Please compute the address of variable **StatInt** in memory ,
and determine the value of variable **StatInt** you can read from memory:

Address of **StatInt** is

Value of **StatInt** in memory is

Please justify your answers.

1.3 [10points] Can you determine the address of variable **LocalInt** in the expression? **YES or NO.**

Please circle around your answer. IF No is your answer, then go to the next question

ELSE Please compute the address of variable **LocalInt** in memory ,
and determine the value of variable **LocalInt** you can read from memory:

Address of **LocalInt** is

Value of **LocalInt** in memory is....

Please justify your answers.

1.4 [10 points] Can you determine the address of variable **result** in the expression? **YES or NO.**

Please circle around your answer. IF No is your answer, then go to the next question

ELSE Please compute the address of variable **result** in memory ,
and determine the value of variable **result** you can read from memory:

Address of **result** is

Value of **result** in memory is

Please justify your answers.

Question 2.

A student wrote MIPS assembly program and executed it in MARS simulator.

```
.data
array1: .word -1,0x7fffffff,0x10000080,0x80000010
.text
main:
    la $t1,array1
# create Frame pointer
    add $fp,$zero,$sp
#Store the address of the first element on stack
using frame pointer
    sw $t1,0($fp)
#allocate memory on Stack for 6 integers
    addi $sp,$sp,-24
#load FIRST element from array1[0] to register $s0
    lw $s0,0($t1)
#push $s0 (NO PUSH!) i.e. store register $s0
on #top of the stack
    sw $s0,0($sp)
#load SECOND element from array1[1] to register $s0
    lw $s0,4($t1)
#create new top of the stack
    addi $sp,$sp,-4
    sw $s0,0($sp)
    #
#load third element from array1[2] to register
$s0
    lw $s0,8($t1)
#create new top of the stack
    addi $sp,$sp,-4
    sw $s0,0($sp)
#load forth element from array1[3] to register
$s0
    lw $s0,12($t1)
#create new top of the stack
    addi $sp,$sp,-4
    sw $s0,0($sp)
```

Last NAME:

Quiz 2 , March 23, 20222

First Name:

After execution of the program in MARS simulator, he displayed the following memory windows and register file:

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x7ffffc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x80000010	0x10000080
0x7ffffe0	0x7fffffff	0xffffffff	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x10010000
0x7ffff00	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff20	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff40	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff60	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff80	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffa0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Data Segment					
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)
0x10010000	0xffffffff	0x7fffffff	0x10000080	0x80000010	0x10000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Registers		
Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x0000000a
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x10010000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x80000010
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffd8
\$fp	30	0x7fffffc
\$ra	31	0x00000000
pc		0x00400044
hi		0x00000000
lo		0x00000000

Figure 2. Register file and memory windows in MARS simulator.

Last NAME:

Quiz 2 , March 23, 20222

First Name:

Based on the information displayed in **Figure 2.** memory windows and register file above, please answer the following questions

2.1.1 [15 points] What is the address of an integer that was **first** pushed on to stack?

2.1.2 [15 points] What is the value in Hex and signed decimal of an integer that was **first** pushed on to stack?

2.1.3 [15 points] What is the offset from FRAME POINTER to an integer that was **first** pushed on to stack?

2.2.1 [5 points] What is the address of an integer that was **Last** pushed on to stack?

2.2.2 [5 points] What is the value in Hex and signed decimal of an integer that was **Last** pushed on to stack?

2.2.3 [5 points] What is the offset from FRAME POINTER to an integer that was **Last** pushed on to stack?

2.3 [5 points] Based on the data shown Figure 2., Can you determine if Frame pointer points to an **address** or a **value**? Please circle around your answer.
Please explain.