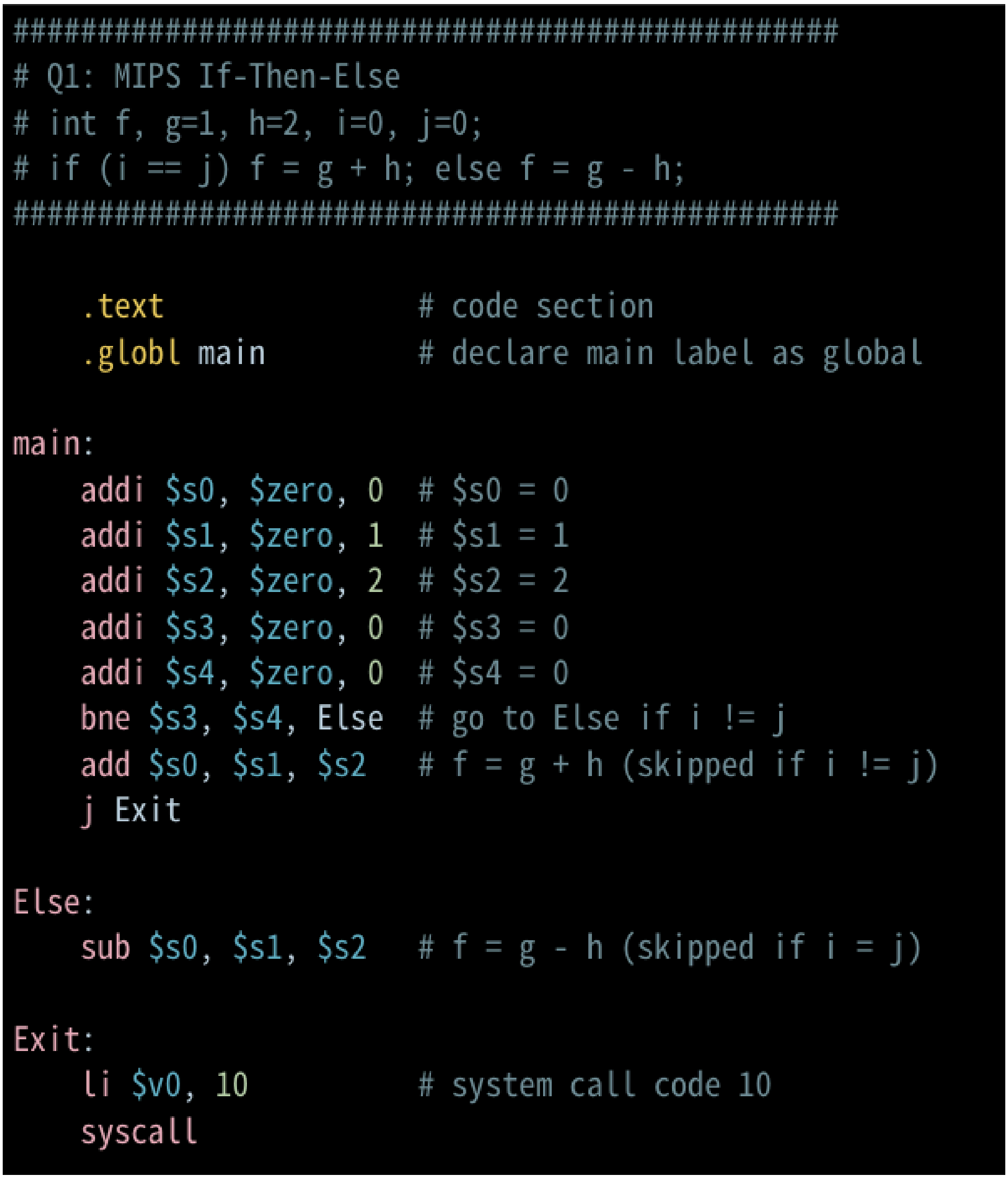
114學年 第一學期 計算機組織與結構 作業報告 日期:2025/12/02
班級: 資工三 學號: B1243030 姓名: 王晨洋

1. 撰寫MIPS程式 (共2題, 100分, 滿分100分)
請於2025/12/02前上傳至M數位園區作業區繳交

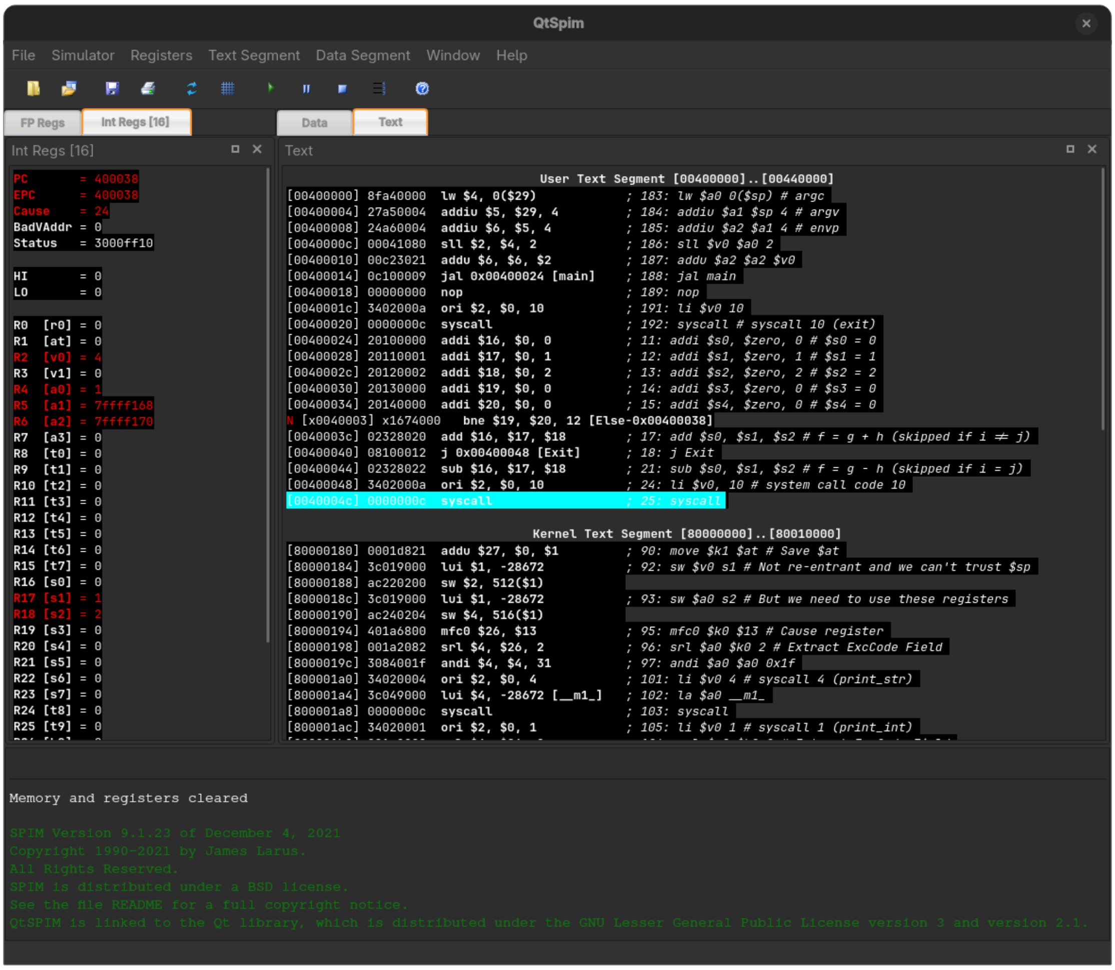
請安裝QtSpim (http://spimsimulator.sourceforge.net/)模擬器, 並請詳細參考課本第二章及附錄A的介紹, 於QtSpim模擬器環境下, 撰寫一完整的MIPS核心指令集版本的程式。(需貼完整程式碼, 截圖呈現結果並文字說明。)

(1) 實作第二章2.7小節範例if-then-else, 請自行完成變數設定, 觀察暫存器及記憶體狀態並說明程式之運作。(50分)



```
#################################################
# Q1: MIPS If-Then-Else
# int f, g=1, h=2, i=0, j=0;
# if (i == j) f = g + h; else f = g - h;
#################################################

    .text               # code section
    .globl main         # declare main label as global

main:
    addi $s0, $zero, 0  # $s0 = 0
    addi $s1, $zero, 1  # $s1 = 1
    addi $s2, $zero, 2  # $s2 = 2
    addi $s3, $zero, 0  # $s3 = 0
    addi $s4, $zero, 0  # $s4 = 0
    bne $s3, $s4, Else  # go to Else if i != j
    add $s0, $s1, $s2   # f = g + h (skipped if i != j)
    j Exit

Else:
    sub $s0, $s1, $s2   # f = g - h (skipped if i = j)

Exit:
    li $v0, 10          # system call code 10
    syscall
```

初始化 f =$s0, g=$s1, h=$s2, i=$s3, j=$s4


QtSpim
File Simulator Registers Text Segment Data Segment Window Help
FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text

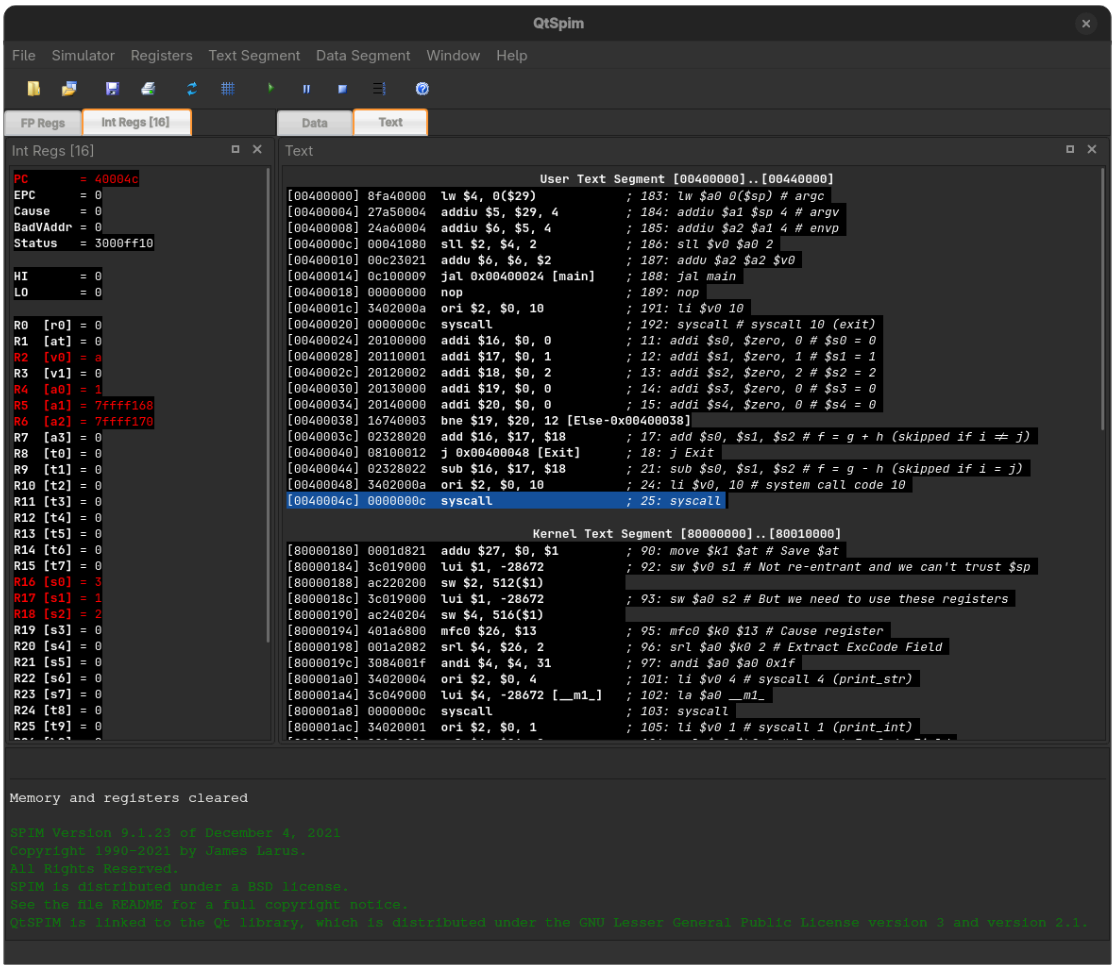
```
PC       = 400038
EPC      = 400038
Cause    = 24
BadVAddr = 0
Status   = 3000ff10

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 4
R3  [v1] = 0
R4  [a0] = 1
R5  [a1] = 7ffff168
R6  [a2] = 7ffff170
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 1
R18 [s2] = 2
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

```
                    User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)            ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004  addiu $5, $29, 4         ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004  addiu $6, $5, 4          ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080  sll $2, $4, 2            ; 186: sll $v0 $a0 2
[00400010] 00c23021  addu $6, $6, $2          ; 187: addu $a2 $a2 $v0
[00400014] 0c100009  jal 0x00400024 [main]    ; 188: jal main
[00400018] 00000000  nop                      ; 189: nop
[0040001c] 3402000a  ori $2, $0, 10           ; 191: li $v0 10
[00400020] 0000000c  syscall                  ; 192: syscall # syscall 10 (exit)
[00400024] 20100000  addi $16, $0, 0          ; 11: addi $s0, $zero, 0 # $s0 = 0
[00400028] 20110001  addi $17, $0, 1          ; 12: addi $s1, $zero, 1 # $s1 = 1
[0040002c] 20120002  addi $18, $0, 2          ; 13: addi $s2, $zero, 2 # $s2 = 2
[00400030] 20130000  addi $19, $0, 0          ; 14: addi $s3, $zero, 0 # $s3 = 0
[00400034] 20140000  addi $20, $0, 0          ; 15: addi $s4, $zero, 0 # $s4 = 0
N [x0040003] x1674000   bne $19, $20, 12 [Else-0x00400038]
[0040003c] 02328020  add $16, $17, $18        ; 17: add $s0, $s1, $s2 # f = g + h (skipped if i ≠ j)
[00400040] 08100012  j 0x00400048 [Exit]      ; 18: j Exit
[00400044] 02328022  sub $16, $17, $18        ; 21: sub $s0, $s1, $s2 # f = g - h (skipped if i = j)
[00400048] 3402000a  ori $2, $0, 10           ; 24: li $v0, 10 # system call code 10
[0040004c] 0000000c  syscall                  ; 25: syscall

                    Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1         ; 90: move $k1 $at # Save $at
[80000184] 3c019000  lui $1, -28672           ; 92: sw $v0 s1 # Not re-entrant and we can't trust $sp
[80000188] ac220200  sw $2, 512($1)
[8000018c] 3c019000  lui $1, -28672           ; 93: sw $a0 s2 # But we need to use these registers
[80000190] ac240204  sw $4, 516($1)
[80000194] 401a6800  mfc0 $26, $13            ; 95: mfc0 $k0 $13 # Cause register
[80000198] 001a2082  srl $4, $26, 2           ; 96: srl $a0 $k0 2 # Extract ExcCode Field
[8000019c] 3084001f  andi $4, $4, 31          ; 97: andi $a0 $a0 0x1f
[800001a0] 34020004  ori $2, $0, 4            ; 101: li $v0 4 # syscall 4 (print_str)
[800001a4] 3c049000  lui $4, -28672 [__m1_]   ; 102: la $a0 __m1_
[800001a8] 0000000c  syscall                  ; 103: syscall
[800001ac] 34020001  ori $2, $0, 1            ; 105: li $v0 1 # syscall 1 (print_int)
```

```
Memory and registers cleared

SPIM Version 9.1.23 of December 4, 2021
Copyright 1990-2021 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

在斷點中檢查暫存器狀態，確保數值正確 $s1 = 1, $s2 = 2，其餘爲 0

判斷式使用 bne, else，判斷式中 i == j 因此 f = g + h，3 = 1 + 2。



QtSpim
File Simulator Registers Text Segment Data Segment Window Help
FP Regs
Int Regs [16]
Data
Text

Int Regs [16]

```
PC       = 40004c
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 3000ff10

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = a
R3  [v1] = 0
R4  [a0] = 1
R5  [a1] = 7ffff168
R6  [a2] = 7ffff170
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 3
R17 [s1] = 1
R18 [s2] = 2
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

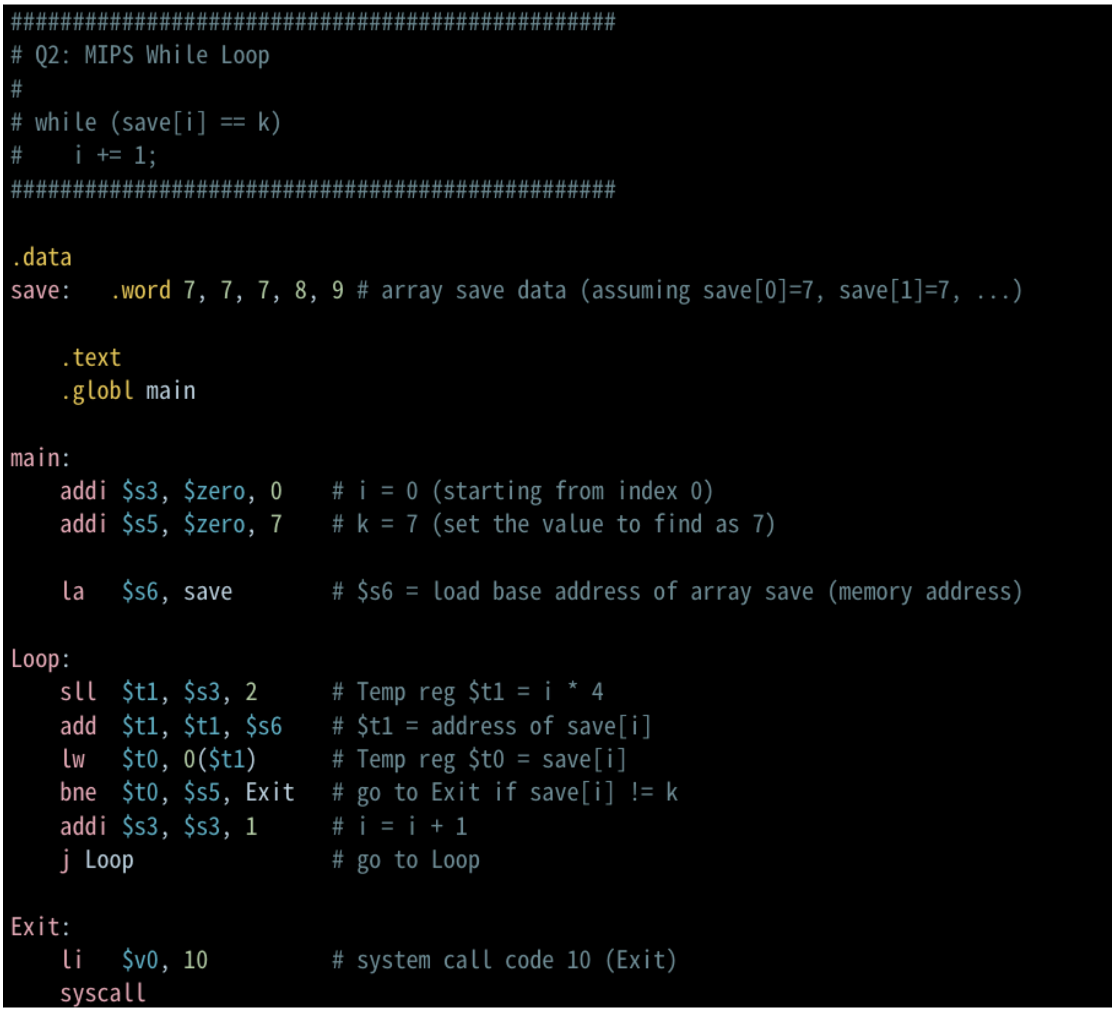
Text

```
                         User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)            ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004  addiu $5, $29, 4         ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004  addiu $6, $5, 4          ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080  sll $2, $4, 2            ; 186: sll $v0 $a0 2
[00400010] 00c23021  addu $6, $6, $2          ; 187: addu $a2 $a2 $v0
[00400014] 0c100009  jal 0x00400024 [main]    ; 188: jal main
[00400018] 00000000  nop                      ; 189: nop
[0040001c] 3402000a  ori $2, $0, 10           ; 191: li $v0 10
[00400020] 0000000c  syscall                  ; 192: syscall # syscall 10 (exit)
[00400024] 20100000  addi $16, $0, 0          ; 11: addi $s0, $zero, 0 # $s0 = 0
[00400028] 20110001  addi $17, $0, 1          ; 12: addi $s1, $zero, 1 # $s1 = 1
[0040002c] 20120002  addi $18, $0, 2          ; 13: addi $s2, $zero, 2 # $s2 = 2
[00400030] 20130000  addi $19, $0, 0          ; 14: addi $s3, $zero, 0 # $s3 = 0
[00400034] 20140000  addi $20, $0, 0          ; 15: addi $s4, $zero, 0 # $s4 = 0
[00400038] 16740003  bne $19, $20, 12 [Else-0x00400038]
[0040003c] 02328020  add $16, $17, $18        ; 17: add $s0, $s1, $s2 # f = g + h (skipped if i ≠ j)
[00400040] 08100012  j 0x00400048 [Exit]      ; 18: j Exit
[00400044] 02328022  sub $16, $17, $18        ; 21: sub $s0, $s1, $s2 # f = g - h (skipped if i = j)
[00400048] 3402000a  ori $2, $0, 10           ; 24: li $v0, 10 # system call code 10
[0040004c] 0000000c  syscall                  ; 25: syscall

                        Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1         ; 90: move $k1 $at # Save $at
[80000184] 3c019000  lui $1, -28672           ; 92: sw $v0 s1 # Not re-entrant and we can't trust $sp
[80000188] ac220200  sw $2, 512($1)
[8000018c] 3c019000  lui $1, -28672           ; 93: sw $a0 s2 # But we need to use these registers
[80000190] ac240204  sw $4, 516($1)
[80000194] 401a6800  mfc0 $26, $13            ; 95: mfc0 $k0 $13 # Cause register
[80000198] 001a2082  srl $4, $26, 2           ; 96: srl $a0 $k0 2 # Extract ExcCode Field
[8000019c] 3084001f  andi $4, $4, 31          ; 97: andi $a0 $a0 0x1f
[800001a0] 34020004  ori $2, $0, 4            ; 101: li $v0 4 # syscall 4 (print_str)
[800001a4] 3c049000  lui $4, -28672 [__m1_]   ; 102: la $a0 __m1_
[800001a8] 0000000c  syscall                  ; 103: syscall
[800001ac] 34020001  ori $2, $0, 1            ; 105: li $v0 1 # syscall 1 (print_int)
```

```
Memory and registers cleared

SPIM Version 9.1.23 of December 4, 2021
Copyright 1990-2021 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

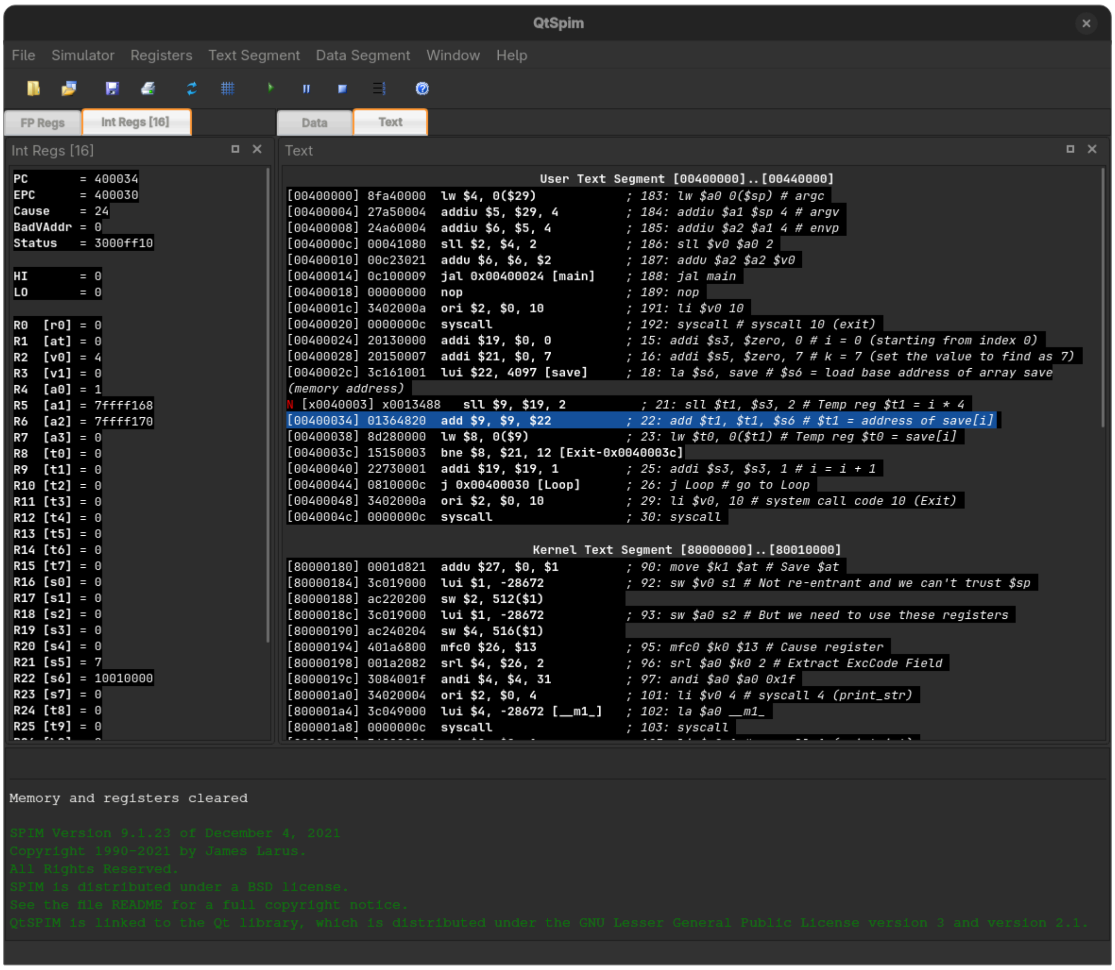
同樣因為 if 判斷式成立，執行 THEN 內容後跳轉至 Exit。
此時結束時 $s0 (f) 的值爲 3。

(2) 實作第二章2.7小節範例while迴圈，請自行完成變數設定，觀察暫存器及記憶體狀態並說明程式之運作。(50分)



```
##################################################
# Q2: MIPS While Loop
#
# while (save[i] == k)
#     i += 1;
##################################################

.data
save:   .word 7, 7, 7, 8, 9 # array save data (assuming save[0]=7, save[1]=7, ...)

    .text
    .globl main

main:
    addi $s3, $zero, 0    # i = 0 (starting from index 0)
    addi $s5, $zero, 7    # k = 7 (set the value to find as 7)

    la   $s6, save        # $s6 = load base address of array save (memory address)

Loop:
    sll  $t1, $s3, 2      # Temp reg $t1 = i * 4
    add  $t1, $t1, $s6    # $t1 = address of save[i]
    lw   $t0, 0($t1)      # Temp reg $t0 = save[i]
    bne  $t0, $s5, Exit   # go to Exit if save[i] != k
    addi $s3, $s3, 1      # i = i + 1
    j Loop                # go to Loop

Exit:
    li   $v0, 10          # system call code 10 (Exit)
    syscall
```

```
i = 0, k = 7
$s6 =  save []
$s3 = i, $s5 = k
save =  [7, 7, 7, 8, 9]  # 把陣列 save 的內容設爲 [7, 7, 7, 8, 9]
```



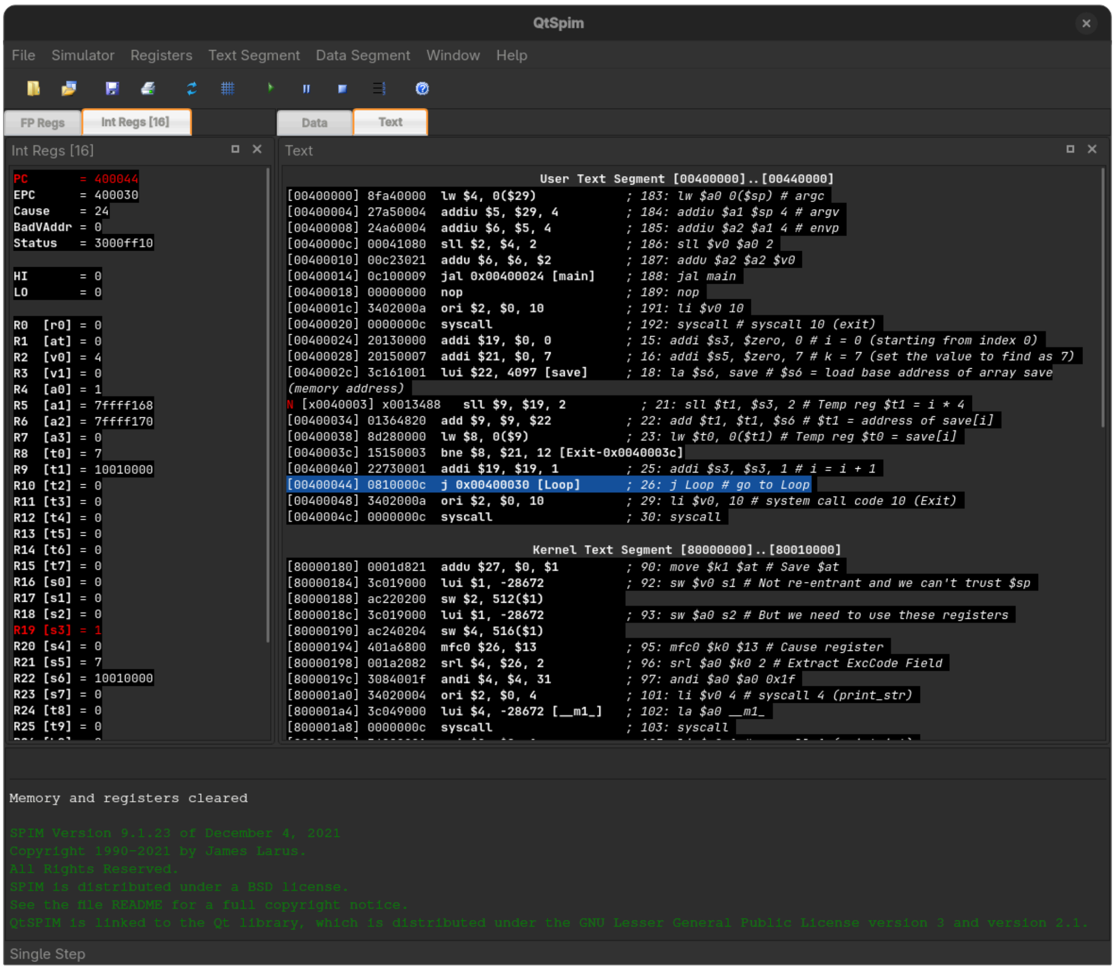
QtSpim
File Simulator Registers Text Segment Data Segment Window Help
FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text

```
PC       = 400034
EPC      = 400030
Cause    = 24
BadVAddr = 0
Status   = 3000ff10

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 4
R3  [v1] = 0
R4  [a0] = 1
R5  [a1] = 7ffff168
R6  [a2] = 7ffff170
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 7
R22 [s6] = 10010000
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

```
                         User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)            ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004  addiu $5, $29, 4         ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004  addiu $6, $5, 4          ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080  sll $2, $4, 2            ; 186: sll $v0 $a0 2
[00400010] 00c23021  addu $6, $6, $2          ; 187: addu $a2 $a2 $v0
[00400014] 0c100009  jal 0x00400024 [main]    ; 188: jal main
[00400018] 00000000  nop                      ; 189: nop
[0040001c] 3402000a  ori $2, $0, 10           ; 191: li $v0 10
[00400020] 0000000c  syscall                  ; 192: syscall # syscall 10 (exit)
[00400024] 20130000  addi $19, $0, 0          ; 15: addi $s3, $zero, 0 # i = 0 (starting from index 0)
[00400028] 20150007  addi $21, $0, 7          ; 16: addi $s5, $zero, 7 # k = 7 (set the value to find as 7)
[0040002c] 3c161001  lui $22, 4097 [save]     ; 18: la $s6, save # $s6 = load base address of array save
(memory address)
N [x0040003] x0013488   sll $9, $19, 2          ; 21: sll $t1, $s3, 2 # Temp reg $t1 = i * 4
[00400034] 01364820  add $9, $9, $22          ; 22: add $t1, $t1, $s6 # $t1 = address of save[i]
[00400038] 8d280000  lw $8, 0($9)             ; 23: lw $t0, 0($t1) # Temp reg $t0 = save[i]
[0040003c] 15150003  bne $8, $21, 12 [Exit-0x0040003c]
[00400040] 22730001  addi $19, $19, 1         ; 25: addi $s3, $s3, 1 # i = i + 1
[00400044] 0810000c  j 0x00400030 [Loop]      ; 26: j Loop # go to Loop
[00400048] 3402000a  ori $2, $0, 10           ; 29: li $v0, 10 # system call code 10 (Exit)
[0040004c] 0000000c  syscall                  ; 30: syscall

                        Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1         ; 90: move $k1 $at # Save $at
[80000184] 3c019000  lui $1, -28672           ; 92: sw $v0 s1 # Not re-entrant and we can't trust $sp
[80000188] ac220200  sw $2, 512($1)
[8000018c] 3c019000  lui $1, -28672           ; 93: sw $a0 s2 # But we need to use these registers
[80000190] ac240204  sw $4, 516($1)
[80000194] 401a6800  mfc0 $26, $13            ; 95: mfc0 $k0 $13 # Cause register
[80000198] 001a2082  srl $4, $26, 2           ; 96: srl $a0 $k0 2 # Extract ExcCode Field
[8000019c] 3084001f  andi $4, $4, 31          ; 97: andi $a0 $a0 0x1f
[800001a0] 34020004  ori $2, $0, 4            ; 101: li $v0 4 # syscall 4 (print_str)
[800001a4] 3c049000  lui $4, -28672 [__m1_]   ; 102: la $a0 __m1_
[800001a8] 0000000c  syscall                  ; 103: syscall
```

```
Memory and registers cleared

SPIM Version 9.1.23 of December 4, 2021
Copyright 1990-2021 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

i = 0 開始迴圈。每輪計算 save[i] 的地址 $t1 載入值到 $t0。


QtSpim
File Simulator Registers Text Segment Data Segment Window Help
FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text
Single Step

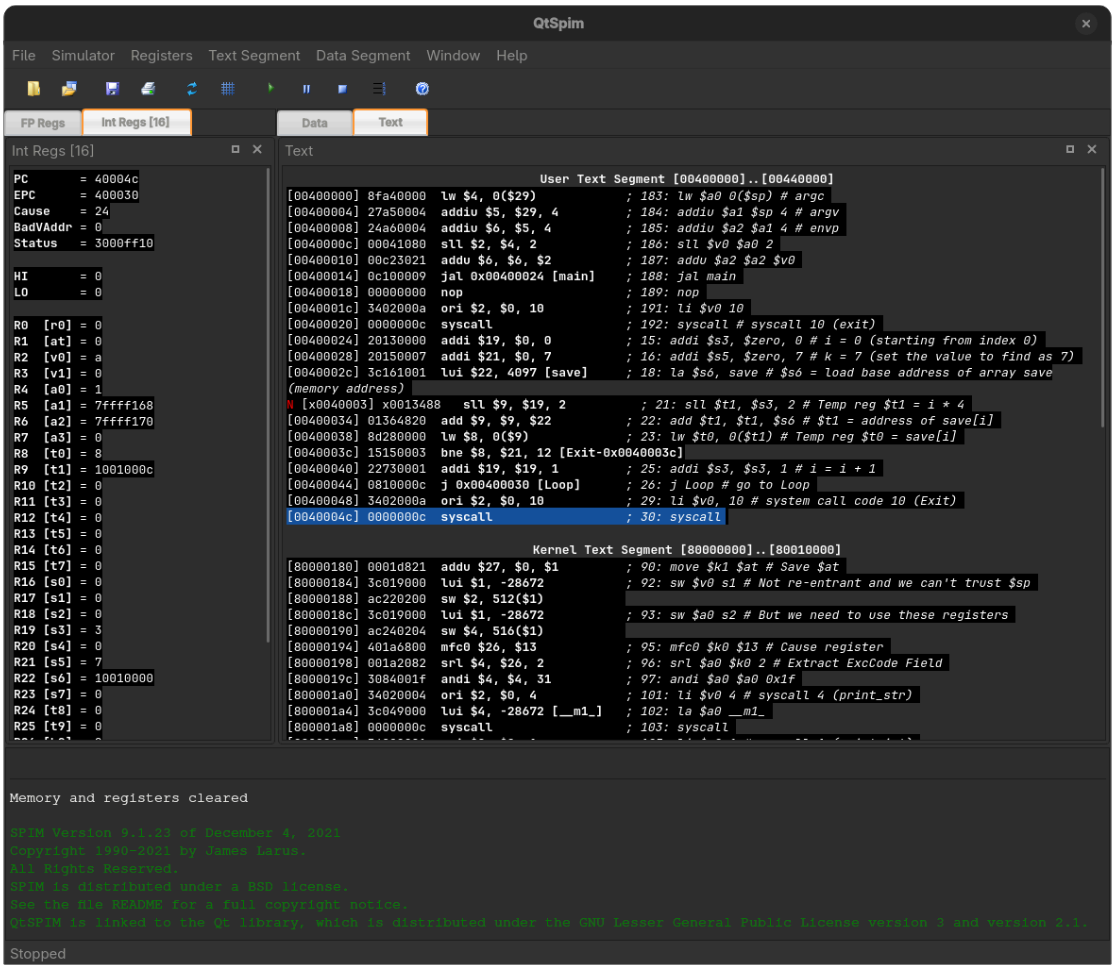
```
PC       = 400044
EPC      = 400030
Cause    = 24
BadVAddr = 0
Status   = 3000ff10

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = 4
R3  [v1] = 0
R4  [a0] = 1
R5  [a1] = 7ffff168
R6  [a2] = 7ffff170
R7  [a3] = 0
R8  [t0] = 7
R9  [t1] = 10010000
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 1
R20 [s4] = 0
R21 [s5] = 7
R22 [s6] = 10010000
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

```
                       User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)            ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004  addiu $5, $29, 4         ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004  addiu $6, $5, 4          ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080  sll $2, $4, 2            ; 186: sll $v0 $a0 2
[00400010] 00c23021  addu $6, $6, $2          ; 187: addu $a2 $a2 $v0
[00400014] 0c100009  jal 0x00400024 [main]    ; 188: jal main
[00400018] 00000000  nop                      ; 189: nop
[0040001c] 3402000a  ori $2, $0, 10           ; 191: li $v0 10
[00400020] 0000000c  syscall                  ; 192: syscall # syscall 10 (exit)
[00400024] 20130000  addi $19, $0, 0          ; 15: addi $s3, $zero, 0 # i = 0 (starting from index 0)
[00400028] 20150007  addi $21, $0, 7          ; 16: addi $s5, $zero, 7 # k = 7 (set the value to find as 7)
[0040002c] 3c161001  lui $22, 4097 [save]     ; 18: la $s6, save # $s6 = load base address of array save
(memory address)
N [x0040003] x0013488   sll $9, $19, 2          ; 21: sll $t1, $s3, 2 # Temp reg $t1 = i * 4
[00400034] 01364820  add $9, $9, $22          ; 22: add $t1, $t1, $s6 # $t1 = address of save[i]
[00400038] 8d280000  lw $8, 0($9)             ; 23: lw $t0, 0($t1) # Temp reg $t0 = save[i]
[0040003c] 15150003  bne $8, $21, 12 [Exit-0x0040003c]
[00400040] 22730001  addi $19, $19, 1         ; 25: addi $s3, $s3, 1 # i = i + 1
[00400044] 0810000c  j 0x00400030 [Loop]      ; 26: j Loop # go to Loop
[00400048] 3402000a  ori $2, $0, 10           ; 29: li $v0, 10 # system call code 10 (Exit)
[0040004c] 0000000c  syscall                  ; 30: syscall

                      Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1         ; 90: move $k1 $at # Save $at
[80000184] 3c019000  lui $1, -28672           ; 92: sw $v0 s1 # Not re-entrant and we can't trust $sp
[80000188] ac220200  sw $2, 512($1)
[8000018c] 3c019000  lui $1, -28672           ; 93: sw $a0 s2 # But we need to use these registers
[80000190] ac240204  sw $4, 516($1)
[80000194] 401a6800  mfc0 $26, $13            ; 95: mfc0 $k0 $13 # Cause register
[80000198] 001a2082  srl $4, $26, 2           ; 96: srl $a0 $k0 2 # Extract ExcCode Field
[8000019c] 3084001f  andi $4, $4, 31          ; 97: andi $a0 $a0 0x1f
[800001a0] 34020004  ori $2, $0, 4            ; 101: li $v0 4 # syscall 4 (print_str)
[800001a4] 3c049000  lui $4, -28672 [__m1_]   ; 102: la $a0 __m1_
[800001a8] 0000000c  syscall                  ; 103: syscall
```

```
Memory and registers cleared

SPIM Version 9.1.23 of December 4, 2021
Copyright 1990-2021 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

逐行執行直到 $19 += 1

i = 0, 1, 2 , bne 條件不成立，PC 留在迴圈中，$s3 ($19) += 1。當 i == 3 , $t0 載入值 8。此時 $t0 ≠ $s5（即 8 ≠ 7）, bne $t0, $s5, Exit 條件成立，PC 跳轉到 Exit 結束迴圈。



QtSpim
File Simulator Registers Text Segment Data Segment Window Help
FP Regs
Int Regs [16]
Data
Text
Int Regs [16]
Text
Stopped

```
PC       = 40004c
EPC      = 400030
Cause    = 24
BadVAddr = 0
Status   = 3000ff10

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = a
R3  [v1] = 0
R4  [a0] = 1
R5  [a1] = 7ffff168
R6  [a2] = 7ffff170
R7  [a3] = 0
R8  [t0] = 8
R9  [t1] = 1001000c
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 3
R20 [s4] = 0
R21 [s5] = 7
R22 [s6] = 10010000
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
```

```
                         User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)            ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004  addiu $5, $29, 4         ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004  addiu $6, $5, 4          ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080  sll $2, $4, 2            ; 186: sll $v0 $a0 2
[00400010] 00c23021  addu $6, $6, $2          ; 187: addu $a2 $a2 $v0
[00400014] 0c100009  jal 0x00400024 [main]    ; 188: jal main
[00400018] 00000000  nop                      ; 189: nop
[0040001c] 3402000a  ori $2, $0, 10           ; 191: li $v0 10
[00400020] 0000000c  syscall                  ; 192: syscall # syscall 10 (exit)
[00400024] 20130000  addi $19, $0, 0          ; 15: addi $s3, $zero, 0 # i = 0 (starting from index 0)
[00400028] 20150007  addi $21, $0, 7          ; 16: addi $s5, $zero, 7 # k = 7 (set the value to find as 7)
[0040002c] 3c161001  lui $22, 4097 [save]     ; 18: la $s6, save # $s6 = load base address of array save
(memory address)
N [x0040003] x0013488   sll $9, $19, 2          ; 21: sll $t1, $s3, 2 # Temp reg $t1 = i * 4
[00400034] 01364820  add $9, $9, $22          ; 22: add $t1, $t1, $s6 # $t1 = address of save[i]
[00400038] 8d280000  lw $8, 0($9)             ; 23: lw $t0, 0($t1) # Temp reg $t0 = save[i]
[0040003c] 15150003  bne $8, $21, 12 [Exit-0x0040003c]
[00400040] 22730001  addi $19, $19, 1         ; 25: addi $s3, $s3, 1 # i = i + 1
[00400044] 0810000c  j 0x00400030 [Loop]      ; 26: j Loop # go to Loop
[00400048] 3402000a  ori $2, $0, 10           ; 29: li $v0, 10 # system call code 10 (Exit)
[0040004c] 0000000c  syscall                  ; 30: syscall

                       Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1         ; 90: move $k1 $at # Save $at
[80000184] 3c019000  lui $1, -28672           ; 92: sw $v0 s1 # Not re-entrant and we can't trust $sp
[80000188] ac220200  sw $2, 512($1)
[8000018c] 3c019000  lui $1, -28672           ; 93: sw $a0 s2 # But we need to use these registers
[80000190] ac240204  sw $4, 516($1)
[80000194] 401a6800  mfc0 $26, $13            ; 95: mfc0 $k0 $13 # Cause register
[80000198] 001a2082  srl $4, $26, 2           ; 96: srl $a0 $k0 2 # Extract ExcCode Field
[8000019c] 3084001f  andi $4, $4, 31          ; 97: andi $a0 $a0 0x1f
[800001a0] 34020004  ori $2, $0, 4            ; 101: li $v0 4 # syscall 4 (print_str)
[800001a4] 3c049000  lui $4, -28672 [__m1_]   ; 102: la $a0 __m1_
[800001a8] 0000000c  syscall                  ; 103: syscall
```

```
Memory and registers cleared

SPIM Version 9.1.23 of December 4, 2021
Copyright 1990-2021 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

最後 $s3 (i) = 3，即連續相等元素的個數。