

## Bai 1:

```
1 .eqv IN_ADRESS_HEXА_KEYBOARD 0xFFFF0012
2 .eqv OUT_ADRESS_HEXА_KEYBOARD 0xFFFF0014
3
4
5 .text
6 main:
7     li $t1, IN_ADRESS_HEXА_KEYBOARD
8     li $t2, OUT_ADRESS_HEXА_KEYBOARD
9
10
11 start_polling_1:
12     li $t3, 0x01 # check row 1 with key 0, 1, 2, 4
13     sb $t3, 0($t1) # must reassign expected row
14     jal polling
15
16 start_polling_2:
17     li $t3, 0x02 # check row 2 with key 4, 5, 6, 7
18     sb $t3, 0($t1) # must reassign expected row
19     jal polling
20
21 start_polling_3:
22     li $t3, 0x04 # check row 3 with key 8, 9, A, B
23     sb $t3, 0($t1) # must reassign expected row
24     jal polling
25
26 start_polling_4:
27     li $t3, 0x08 # check row 4 with key C, D, E, F
28     sb $t3, 0($t1) # must reassign expected row
29     jal polling
30
31 check_after_polling_4:
32     beq $a0, 0x0, print
33     j start_polling_1
34
35 polling:
36     lb $a0, 0($t2) # read scan code of key button
37     bne $a0, 0x0, print
38     jr $ra
```

```
39
40 print:
41     li $v0, 34 # print integer (hexa)
42     syscall
43
44 sleep:
45     li $a0, 3000 # sleep 100ms
46     li $v0, 32
47     syscall
48
49 back_to_start_polling:
50     j start_polling_1      # back to check row 1
51
```

Line: 37 Column: 22 ☒ Show Line Numbers

Mars Messages Run I/O

0x000000410x000000110x000000410x000000110x000000220x000000110x000000410x00000024

Clear

Yêu cầu: Check toàn bộ các ký tự từ 0 -> F

In ra kết quả khi nhập mã số sinh viên từ bàn phím

0x41 là số 2;

0x11 là số 0;

0x22 là số 5;

0x24 là số 9;

Từ kết quả => MSSV của em: 20205029

## Bai 2:

The screenshot displays the Digital Lab Sim interface. The main window shows MIPS assembly code for a program that prints a message and handles an interrupt. A modal window titled "Digital Lab Sim, Version 1.0 (Didier Teifreto)" is overlaid, featuring a digital display showing "8.8.", a numeric keypad, and a "Tool Control" section with buttons for "Disconnect from MIPS", "Reset", "Help", and "Close". The registers window on the right lists various MIPS registers and their values. The bottom panel shows the "Mars Messages" window with the output "Pham Minh Thong--20205029." and a status message "-- program is finished running (dropped off bottom) --".

**Assembly Code (demolab11.asm):**

```

1 .eqv IN_ADDRESS_HEX_KEYBOARD 0xFFFF0012
2
3 .data
4 Message: .asciiz "Pham Minh Thong--20205029.\n"
5
6 .text
7 main:
8
9     li $t1, IN_ADDRESS_HEX_KEYBOARD
10    li $t3, 0x80 # bit 7 of = 1 to enable interrupt
11    sb $t3, 0($t1)
12
13    #-----
14    # No-end loop, main program, to demo the effective of interrupt
15    #-----
16
17 Loop:
18     nop
19     nop
20     nop
21     b Loop # Wait for interrupt
22
23 end_main:
24 #-----
25 # GENERAL INTERRUPT SERVED ROUTINE for all interrupts
26 #-----
27 .ktext 0x80000180
28 #-----
29 # Processing
30 #-----
31 IntSR:
32     addi $v0, $zero, 4 # show message
33     la $a0, Message
34     syscall
35
36     #-----
37     # Evaluate the return address of main routine
38     # epc <= epc + 4
39     #-----S
40 next_pc:
41     mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
42     addi $at, $at, 4 # $at = $at + 4 (next instruction)
43     mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
44
45 return:
46     eret # Return from exception

```

**Registers:**

Register	Value
\$0	0x00000000
\$1	0x00400024
\$2	0x00000004
\$3	0x00000000
\$4	0x10010000
\$5	0x00000000
\$6	0x00000000
\$7	0x00000000
\$8	0x00000000
\$9	0xffff0012
\$10	0x00000000
\$11	0x00000080
\$12	0x00000000
\$13	0x00000000
\$14	0x00000000
\$15	0x00000000
\$16	0x00000000
\$17	0x00000000
\$18	0x00000000
\$19	0x00000000
\$20	0x00000000
\$21	0x00000000
\$22	0x00000000
\$23	0x00000000
\$24	0x00000000
\$25	0x00000000
\$26	0x00000000
\$27	0x00000000
\$28	0x10008000
\$29	0x7ffffcfc
\$30	0x00000000
\$31	0x00000000
pc	0x00400024
hi	0x00000000
lo	0x00000000

**Mars Messages:**

```

Pham Minh Thong--20205029.
-- program is finished running (dropped off bottom) --

```

## Yêu cầu:

Khi nhấn phím bất kì từ 0 -> F thì sẽ hiện ra tên và mssv

### Bài 3:

```

demolab11.asm lab11_1.asm mips1.asm*
1  .eqv IN_ADRESS_HEX_A_KEYBOARD 0xFFFF0012
2  .eqv OUT_ADRESS_HEX_A_KEYBOARD 0xFFFF0014
3
4  .data
5  Message: .asciiz "Key scan code "
6
7  .text
8  main:
9      li $t1, IN_ADRESS_HEX_A_KEYBOARD
10     li $t3, 0x80 # bit 7 = 1 to enable
11     sb $t3, 0($t1)
12
13
14     xor $s0, $s0, $s0 # count = $s0 = 0
15
16 Loop:
17     addi $s0, $s0, 1 # count = count + 1
18 prn_seq:
19     addi $v0, $zero, 1
20     add $a0, $s0, $zero # print auto sequence number
21     syscall
22
23 prn_eol:
24     addi $v0, $zero, 11
25     li $a0, '\n' # print endofline
26     syscall
27 sleep:
28     addi $v0, $zero, 32
29     li $a0, 3000 # sleep 300 ms
30     syscall
31     nop # WARNING: nop is mandatory here.
32     b Loop # Loop
33 end_main:
34
35 # Interrupt
36 .ktext 0x80000180
37 #-----
38 # SAVE the current REG FILE to stack
39 #-----
40 IntSR:
41     addi $sp, $sp, 4 # Save $ra because we may change it later
42     sw $ra, 0($sp)
43     addi $sp, $sp, 4 # Save $ra because we may change it later
44     sw $at, 0($sp)
45     addi $sp, $sp, 4 # Save $ra because we may change it later
46     sw $v0, 0($sp)
47     addi $sp, $sp, 4 # Save $a0, because we may change it later
48     sw $a0, 0($sp)
49     addi $sp, $sp, 4 # Save $t1, because we may change it later
50     sw $t1, 0($sp)
51     addi $sp, $sp, 4 # Save $t3, because we may change it later
52     sw $t3, 0($sp)
53 #-----
54 # Processing
55 #-----
56 prn_msg:
57     addi $v0, $zero, 4
58     la $a0, Message
59     syscall
60
61 get_cod:
62     li $t1, IN_ADRESS_HEX_A_KEYBOARD
63     li $t2, OUT_ADRESS_HEX_A_KEYBOARD
64
65 start_inter_1:
66     li $t3, 0x81 # check row 1 with key 0, 1, 2, 4
67     sb $t3, 0($t1) # must reassign expected row
68     jal inter
69

```

```

69
70 start_inter_2:
71     li $t3, 0x82 # check row 2 with key 4, 5, 6, 7
72     sb $t3, 0($t1) # must reassign expected row
73     jal inter
74
75 start_inter_3:
76     li $t3, 0x84 # check row 3 with key 8, 9, A, B
77     sb $t3, 0($t1) # must reassign expected row
78     jal inter
79
80 start_inter_4:
81     li $t3, 0x88 # check row 4 with key C, D, E, F
82     sb $t3, 0($t1) # must reassign expected row
83     jal inter
84
85 check_after_inter_4:
86     beq $a0, 0x0, prn_cod
87     j start_inter_1
88
89 inter:
90     lb $a0, 0($t2) # read scan code of key button
91     bne $a0, 0x0, prn_cod
92     jr $ra
93
94 prn_cod:
95     li $v0, 34
96     syscall
97
98     li $v0, 11
99     li $a0, '\n' # print endofline
100    syscall
101    #-----

```

```

101    #-----
102    # Evaluate the return address of main routine
103    # epc <= epc + 4
104    #-----
105 next_pc:
106     mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
107     addi $at, $at, 4 # $at = $at + 4 (next instruction)
108     mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
109     #-----
110     # RESTORE the REG FILE from STACK
111     #-----
112 restore:
113     lw $t3, 0($sp) # Restore the registers from stack
114     addi $sp, $sp, -4
115     lw $t1, 0($sp) # Restore the registers from stack
116     addi $sp, $sp, -4
117     lw $a0, 0($sp) # Restore the registers from stack
118     addi $sp, $sp, -4
119     lw $v0, 0($sp) # Restore the registers from stack
120     addi $sp, $sp, -4
121     lw $ra, 0($sp) # Restore the registers from stack
122     addi $sp, $sp, -4
123 return:
124     eret # Return from exception

```

Line: 107 Column: 29 ☒ Show Line Numbers

Mars Messages

Run I/O

Clear

```

Key scan code 0x00000041
Key scan code 0x00000011
Key scan code 0x00000041
Key scan code 0x00000011
Key scan code 0x00000022
Key scan code 0x00000011
Key scan code 0x00000041
Key scan code 0x00000024

```

Yêu cầu: In ra kết quả khi nhập mssv vào Lab Sim

Kết quả ra như sau:

0x41 là số 2;

0x11 là số 0;

0x22 là số 5;

0x24 là số 9;

=> Kết quả là MSSV của em: 20205029