

## Báo cáo Thực hành Kiến trúc máy tính - Thi Giữa Kỳ

### Bài 9B:

- Code:

```
1  .data
2  .align 4
3  matrixA: .space 5
4  .text
5
6  main:
7  la $a1,matrixA  #load a pointer to array into $a1
8
9  addi $t1,$t1,5  #size of the array(matrix in array form)
10 matA_loop:
11 addi $t1,$t1,-1  #subtract 1 from $t1, save to $t1
12 li $v0,5         #load 5 into $v0 (read integer)
13 syscall          #input from user
14 sw $v0,0($a1)    #store input int to array
15 addi $a1,$a1,4    #add 4 to $a1, save to $a1
16 bnez $t1,matA_loop  #if $t1 isn't zero, goto loop
17 la $a1,matrixA    #load array pointer into $a1
18
19
20
21 addi $t1,$0,5  #size of the array(matrix in array form)
22 # t2 la so chan nho nhat
23 #t3 la so le lon nhat nho hon t2
24 li $t2, 1000
25 li $t3, -999
26 loop:
27 addi $t1,$t1,-1  #subtract 1 from $t1, save to $t1
28 findEven:
29 lw $t4,0($a1)
```

```

29 lw $t4,0($a1)
30 addi $a1,$a1,4
31
32 slt $t5, $t4, $t2
33 beq $t5, 0, checkStepEven # neu t4 >t2 thi vao checkStepEven
34
35 andi $t0 , $t4 , 0x0001
36 bne $t0, $0, checkStepEven # neu khong chan
37
38 add $t2, $t4, $0
39 checkStepEven:
40 bnez $t1,loop #if $t1 isn't zero,goto loop
41 endFinEven:
42 endLoop:
43
44
45 addi $t1,$0,5 #size of the array(matrix in array form)
46 la $a1,matrixA
47 loopFindOdd:
48 addi $t1,$t1,-1 #subtract 1 from $t1, save to $t1
49 findOdd:
50 lw $t4,0($a1)
51 addi $a1,$a1,4
52 andi $t0 , $t4 , 0x0001
53 beq $t0, 0, checkStepOdd # neu chan thi checkStepOdd
54
55 slt $t6, $t4, $t2 # check co nho hon t2 khong
56 beq $t6, 0, checkStepOdd # neu t4 > t2 thi checkStepOdd

```

```

54
55 slt $t6, $t4, $t2 # check co nho hon t2 khong
56 beq $t6, 0, checkStepOdd # neu t4 > t2 thi checkStepOdd
57
58 slt $t5, $t3, $t4
59 beq $t5, 0, checkStepOdd # neu t3> t4 thi checkStepOdd
60
61 add $t3, $t4, $0
62 checkStepOdd:
63 bnez $t1,loopFindOdd
64 endFindOdd:
65 endLoopFindOdd:
66
67

```

- Kết quả:

Text Segment

Bkpt	Address	Code	Basic	Source
	4194304	0x3c011001	lui \$1,4097	7: la \$a1,matrixA #load a pointer to array in...
	4194308	0x34250000	ori \$5,\$1,0	
	4194312	0x21290005	addi \$9,\$9,5	9: addi \$t1,\$t1,5 #size of the array(matrix i...
	4194316	0x2129ffff	addi \$9,\$9,-1	11: addi \$t1,\$t1,-1 #subtract 1 from \$t1, sav...
	4194320	0x24020005	addiu \$2,\$0,5	12: li \$v0,5 #load 5 into \$v0 (read in...
	4194324	0x0000000c	syscall	13: syscall #input from user
	4194328	0xaca20000	sw \$2,0(\$5)	14: sw \$v0,0(\$a1) #store input int to array
	4194332	0x20a50004	addi \$5,\$5,4	15: addi \$a1,\$a1,4 #add 4 to \$a1, save to \$a1
	4194336	0x1520ffff	bne \$9,\$0,-6	16: bnez \$t1,matrixA_loop #if \$t1 isn't zero,g...
	4194340	0x3c011001	lui \$1,4097	17: la \$a1,matrixA #load array pointer in...

Labels

Label	Address
Bai 2asm	
main	4194304
matrixA_loop	4194316
loop	4194360
findEven	4194364
checkStepEven	4194396
endFinEven	4194400
endLoop	4194400
loopFindOdd	4194412
findOdd	4194412

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	6	8	1	3	5	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0

Registers

Coproc 1

Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	5
\$v1	3	0
\$a0	4	0
\$a1	5	268501012
\$a2	6	0
\$a3	7	0
\$t0	8	1
\$t1	9	0
\$t2	10	6
\$t3	11	5
\$t4	12	5
\$t5	13	1
\$t6	14	1
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194468
hi		0
lo		0

0x10010000 (.data)

Hexadecimal Addresses

Hexadecimal Values

ASCII

## Bài 6C:

### - Code:

```
1  .data
2      input: .space 101
3      message: .asciiz "So tu bat dau bang 'Ch' la: "
4      check_C: .byte 'C'
5      check_h: .byte 'h'
6  .text
7      main:
8          la $a0, input
9          li $a1, 101          #gioi han chi duoc nhap 100 ky tu
10         li $v0, 8            #doc xau va luu vao $a0
11         syscall
12         nop
13         jal check_Ch
14         li $v0, 56
15         la $a0, message
16         add $a1, $t3, $zero
17         syscall
18         nop
19         li $v0, 10
20         syscall
21         nop
22     check_Ch:
23         la $s0, input          #luu lai vao s0
24         #Lay ra 2 ky tu C va h
25         la $s1, check_C
26         la $s2, check_h
27         lb $t1, 0($s1)
28         lb $t2, 0($s2)
29         li $t3, 0              # so tu bat dau bang 'Ch'
30
31         li $t3, 0              # so tu bat dau bang 'Ch'
32         subi $s0, $s0, 1       # tam thoi tru 1 de vong lap co the chi den ky tu dau tien
33     loop:
34         addi $s0, $s0, 1
35         lb $t0, 0($s0)
36         beq $t0, $zero, endLoop
37     sosanh_C:
38         bne $t0, $t1, skip     # neu ky tu dau tien khac C thi skip
39     sosanh_h:
40         addi $s0, $s0, 1
41         lb $t0, 0($s0)
42         beq $t0, $zero, endLoop
43         bne $t0, $t2, skip     # neu ky tu tiep theo khac h thi ski
44         addi $t3, $t3, 1
45     skip:
46         addi $s0, $s0, 1
47         lb $t0, 0($s0)
48         beq $t0, $zero, endLoop
49         bne $t0, 32, skip
50         j loop
51     endLoop:
52     endcheck_Ch:
53         jr $ra
```

## -Kết quả:

+ Nhập vào string:

**MIPS Keyboard Input**

Enter a string of maximum length 100 (syscall 8)

Cha chi Chu che Cho empi

OK

**Mars Messages**

\*\*\*\* user input : Cha chi Chu che Cho empi

-- program is finished running --

\*\*\*\* user input : Cha chi Chu che Cho empi

## + Kết quả:

**Số từ bắt đầu bằng 'Ch' là: 3**

OK

**Mars Messages**

\*\*\*\* user input : Chì hì Chu cha Cho empi

-- program is finished running --

\*\*\*\* user input : Cha chi Chu che Cho empi

## - Phân tích cách thực hiện:

+ Kiểm tra từ bắt đầu bằng “Ch” bằng cách kiểm tra chữ đầu tiên của từ có phải là “C” không:

Nếu bằng “C” thì kiểm tra chữ tiếp sau có phải là “h” không. Nếu chữ tiếp theo khác “h” thì skip kiểm tra từ tiếp theo; nếu bằng “h” thì tăng dần giá trị thanh ghi t3; sau đó skip để kiểm tra từ tiếp theo.

Nếu không bằng “C” thì skip kiểm tra từ tiếp theo.

+ Kết quả số từ bắt đầu bằng “Ch” được lưu trong thanh ghi a1