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

Bài 9B:

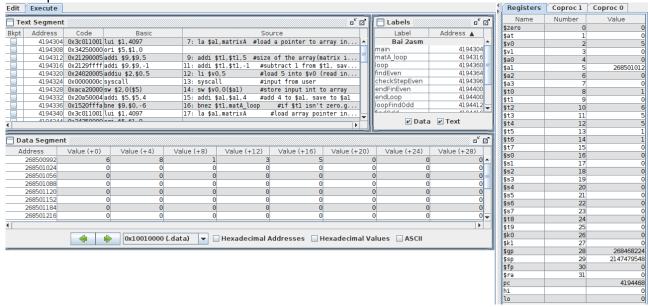
- Code:

```
.data
   align 4
   matrixA: .space 5
3
4
   .text
5
6 main:
7 la $a1,matrixA #load a pointer to array into $a1
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
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:



Bài 6C:

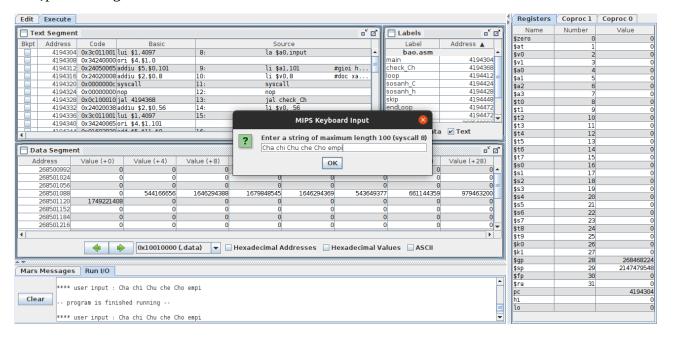
- Code:

```
1 .data
2
             input:
                     space 101
             message: .asciiz "So tu bat dau bang 'Ch' la: "
3
            check_C: .byte 'C'
4
            check_h: .byte 'h'
5
6
   .text
            main:
7
                      la $a0,input
8
                      li $a1.101
                                         #gioi han chi duoc nhap 100 ky tu
9
10
                      8,0v2 il
                                         #doc xau va luu vao $a0
                      syscall
11
                      nop
12
                      jal check_Ch
13
14
                      li $v0, 56
                      la $a0, message
15
                      add $a1, $t3, $zero
16
                      syscall
17
18
                      nop
                      li $v0, 10
19
                     syscall
20
                     nop
22
            check_Ch:
                     la $s0, input
                                                #luu lai vao s0
23
                      #Lay ra 2 ky tu C va h
24
                      la $s1, check_C
25
26
                      la $s2, check_h
                      lb $t1, 0($s1)
27
                      lb $t2, 0($s2)
28
29
                    li $t3, O
                                        # so tu bat dau bang 'Ch'
```

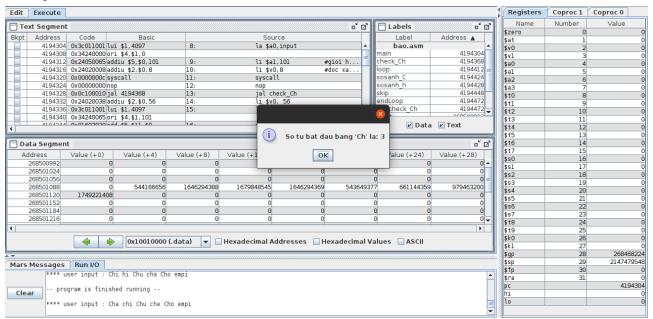
```
li $t3, 0
                                       # so tu bat dau bang 'Ch'
29
                       subi $s0, $s0, 1
                                           # tam thoi tru 1 de vong lap co the chi den ky tu dau tien
30
31
             loop:
                       addi $s0, $s0, 1
32
                      Ib $t0, 0($s0)
33
                      beq $t0, $zero, endLoop
34
             sosanh_C:
35
                      bne $t0, $t1, skip # neu ky tu dau tien khac C thi skip
36
37
             sosanh_h:
                      addi $s0, $s0, 1
38
39
                      lb $t0, 0($s0)
                      beq $t0, $zero, endLoop
40
41
                      bne $t0, $t2, skip # neu ky tu tiep theo khac h thi ski
                      addi $t3, $t3, 1
42
             skip:
43
                      addi $s0, $s0, 1
44
45
                       lb $t0, 0($s0)
                      beq $t0, $zero, endLoop
46
                      bne $t0, 32, skip
47
                      j loop
48
             endLoop:
49
50
             endcheck_Ch:
                      jr $ra
51
52
```

-Kết quả:

+ Nhập vào string:



+ Kết quả:



- 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