

# BÁO CÁO THỰC HÀNH

Họ tên	Lê Minh Nhựt	Lớp: IT012.N22.1
MSSV	22521060	STT: 21
Bài Thực Hành	LAB 5	
CBHD	Trương Văn Cường	

## 1 Yêu cầu thực hành

### Về phần trình bày:

- Sinh viên trình bày đúng theo định dạng báo cáo mà CBHD đưa ra.
- Cần chú thích bảng, hình (nếu có).
- Sử dụng chức năng Insert Caption và Cross-reference cho chú thích Bảng, Hình
- Sử dụng tính năng Screenshot để chụp kết quả mô phỏng.

### Quy trình thực hành:

- Sinh viên chuẩn bị bài ở nhà, và có mặt đúng giờ tại phòng LAB.
- Sinh viên thực hành theo hướng dẫn, và nộp bài đúng hạn.
- Hoàn thành bài tập về nhà (nếu có)
- Tất cả các bài báo cáo có hành vi sao chép của nhau sẽ bị **điểm 0**

### Điểm buổi thực hành

<b>Chuyên cần (20%)</b>		
<b>Trình bày (20%)</b>		
<b>Nội dung thực hành (60%)</b>		
<b>Câu 1:</b>		
<b>Câu 2:</b>		
<b>Tổng (100%)</b>		

## 2. Bài tập thực hành:

### 1. Thực hành theo hướng dẫn video

Thực hành với mảng

## 1. Thực hành với mảng

Mảng với  $n$  phần tử là một chuỗi  $n$  phần tử liên tiếp nhau trong bộ nhớ. Thao tác với mảng trong MIPS là thao tác trực tiếp với byte/word trong bộ nhớ.

- Để cấp phát chuỗi word hoặc byte trong bộ nhớ, có giá trị khởi tạo sử dụng “.word” hoặc “.byte” trong “.data”
- Để cấp phát chuỗi byte không có giá trị khởi tạo trước, sử dụng “.space” trong “.data”

Cho ba mảng với cấp phát dữ liệu trong bộ nhớ như sau:

```
.data
array1: I .word 5, 6, 7, 8, 1, 2, 3, 9, 10, 4
size1:   .word 10

array2: .byte 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
size2:   .word 16

array3: .space 8
size3:   .word 8
```

Mảng array1 có 10 word, kích thước được lưu trong size1; Mảng array2 có 16 byte, kích thước được lưu trong size2; Mảng array3 có 8 byte, kích thước được lưu trong size3.

Viết code trong phần “.text” thực hiện riêng từng phần việc:

- ✓ In ra cửa sổ I/O của MARS tất cả các phần tử của mảng array1 và array2
- ✓ Gán các giá trị cho mảng array3 sao cho
$$array3[i] = array2[i] + array2[size2 - 1 - i]$$
- ✓ Người sử dụng nhập vào mảng thứ mấy và chỉ số phần tử cần lấy trong mảng đó, chương trình xuất ra phần tử tương ứng.

```

Lab5_bai1 Lab5_bai2
1  .data
2      S1: .asciiz "Cac phan thu cua mang array1 la: "
3      array1: .word 5,6,7,8,1,2,3,9,10,4
4      size1: .word 10
5
6      S2: .asciiz "\nCac phan thu cua mang array2 la: "
7      array2: .byte 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
8      size2: .word 16
9
10     S3: .asciiz "\nCac phan thu cua mang array3 la: "
11     array3: .space 8
12     size3: .word 8
13
14     S4: .asciiz "\nNhap mang can tim (mang 1,2,3): "
15     S5: .asciiz "Nhap chi so cua phan tu can tim trong mang: "
16     S6: .asciiz "Phan tu can tim la: "
17     Space: .asciiz " "
18 .text
19 main:
20     #xuat ra chuoai S1
21     li $v0, 4
22     la $a0, S1
23     syscall
24
25     add $t0, $0, $0          #$t0 = i= 0

```

Hình 1: Khởi tạo các giá trị cho mảng

```
Lab5_bai1 Lab5_bai2
25      add $t0, $0, $0          #$t0 = i= 0
26      la $s0, array1          #$s0 = $array[0]
27      lw $t1, size1($0)        #$t1 = size1
28  LOOP1:
29      bge $t0, $t1, EXIT1      #if (i >= size1) EXIT1
30      sll $t2, $t0, 2          #t2 = t0 * 4
31      add $t3, $s0, $t2        #t3 = &array1[0] + t2
32      lw $t4, 0($t3)          #$t4 = $t3
33      # xuất giá trị của phần tử trong mảng array1
34      li $v0, 1
35      move $a0, $t4
36      syscall
37      # xuất các khoảng trắng giữa các phần tử
38      li $v0, 4
39      la $a0, Space
40      syscall
41
42      addi $t0, $t0, 1          # i++
43      j LOOP1                  # nhảy tới LOOP1
44
45  EXIT1:
46      # xuất ra chuỗi S2
47      li $v0, 4
48      la $a0, S2
49      syscall
```

Line: 178 Column: 26 ☒ Show Line Numbers

Hình 2: Xuất ra mảng array1

```
Lab5_bai1 Lab5_bai2
49      syscall
50
51      add    $t0, $0, $0          #$t0 = i = 0
52      la     $s0, array2          $$s0 = &array2[0]
53      lw     $t1, size2($0)        #$t1 = size2
54  LOOP2:
55      bge    $t0, $t1, EXIT2       #if (i>=size2) EXIT2
56      add    $t2, $s0, $t0         #$t2 = &array2[0] + t0
57      lb     $t3, 0($t2)           #$t3 = $t2 (load byte)
58      #xuat ra gia tri cac phan tu trong mang array2
59      li     $v0, 1
60      move   $a0, $t3
61      syscall
62      #xuat ra khoang trang giua cac phan tu trong mang array2
63      li     $v0, 4
64      la     $a0, Space
65      syscall
66
67      addi   $t0, $t0, 1           # i++
68      j      LOOP2               # nhay toi LOOP2
69
70  EXIT2:
71      add    $t0, $0, $0          #i = 0
72      la     $s0, array3          $$s3 = &array3[0]
73      lw     $s1, size3($0)        $$s1 = size3
```

Hình 3: Xuất ra mảng array2

```
Lab5_bai1 Lab5_bai2
73      lw      $s1, size3($0)      # $s1 = size3
74
75      add     $t1, $0, $0          # j=0
76      la      $s2, array2          # s2 = &array2[0]
77      lw      $s3, size2($0)      # s3 = size2
78
79      subi    $t2, $s3, 1          # k = size2-1
80  LOOP3:
81      bge     $t0, $s1, EXIT3      # if(j >= size3) EXIT3
82      add     $t5, $s2, $t1        # $t5 = $s2 + $t1
83      add     $t6, $s2, $t2        # $t6 = $s2 + $t2
84      add     $t7, $s0, $t0        # $t7 = $s0 + $t0
85
86      lb      $s4, 0($t5)          # $s4 = $t5
87      lb      $s5, 0($t6)          # $s5 = $t6
88      add     $s7, $s4, $s5        # $s7 = $s4 + $s5
89      sb      $s7, 0($t7)          # lưu $t7 vào bộ nhớ thông qua $s7
90
91      addi    $t0, $t0, 1          # i++
92      addi    $t1, $t1, 1          # j++
93      subi    $t2, $t2, 1          # k--
94      j       LOOP3               # nhảy tới LOOP3
95  EXIT3:
96      # xuất ra chuỗi s3
97      li      $v0, 4
```

Hình 4: Khởi tạo các giá trị cho array 3 theo yêu cầu

```
Lab5_bai1 Lab5_bai2
97      li      $v0, 4
98      la      $a0, s3
99      syscall
100
101      add     $t0, $0, $0      # i=0
102      la      $s0, array3     # $s0 = &array3
103      lw      $t1, size3($0)  # $t1 = size3
104  LOOP4:
105      bge     $t0, $t1, EXIT4
106      add     $t2, $s0, $t0
107      lb      $t3, 0($t2)
108      # xuất giá trị các phần tử của mảng array3
109      li      $v0, 1
110      move     $a0, $t3
111      syscall
112      # xuất khoảng trắng giữa các phần tử
113      li      $v0, 4
114      la      $a0, Space
115      syscall
116
117      addi    $t0, $t0, 1      # i++
118      j       LOOP4          # nhảy tới LOOP4
119  EXIT4:
120      # xuất ra chuỗi s4
121      li      $v0, 4
```

Hình 5: Xuất mảng array3

```
Lab5_bai1 Lab5_bai2
121      li      $v0, 4
122      la      $a0, S4
123      syscall
124      # nhap vao mang can tim
125      li      $v0, 5
126      syscall
127      move     $t0, $v0
128
129      # xuat ra chuoai S5
130      li      $v0, 4
131      la      $a0, S5
132      syscall
133      # nhap vao chi so cua phan tu can tim trong mang
134      li      $v0, 5
135      syscall
136      move     $t8, $v0
137      # khoi tao cac lua chon 1,2,3
138      addi     $t1, $0, 1      #$t1=1
139      addi     $t2, $0, 2      #$t2=2
140      addi     $t3, $0, 3      #$t3=3
141      # xuat ra chuoai S6
142      li      $v0, 4
143      la      $a0, S6
144      syscall
145
```

Hình 6



```
Lab5_bai1 Lab5_bai2
145
146     bne     $t0, $t1, JUMP1          # if (luachon != 1) JUMP1
147     # duyet qua mang array1
148     la      $s0, array1($0) #s0 = &array1[0]
149     lw      $s2, size1($0) #size1
150
151     sll     $t4, $t8, 2
152     add     $t5, $s0, $t4
153     lw      $t6, 0($t5)
154     # xuat ra phan tu can tim
155     li      $v0, 1
156     move    $a0, $t6
157     syscall
158
159 JUMP1:
160     bne     $t0, $t2, JUMP2          # if (luachon != 2) JUMP2
161     # duyet qua mang array2
162     la      $s0, array2($0) #s1=&array2[0]
163     lw      $s2, size2($0) #size2
164
165     add     $t5, $s1, $t8
166     lb      $t6, 0($t5)
167     # xuat ra phan tu can tim
168     li      $v0, 1
169     move    $a0, $t6
```

Hình 7

```
167      # xuất ra phần tử cần tìm
168      li $v0, 1
169      move $a0, $t6
170      syscall
171  JUMP2:
172      bne $t0, $t3, EXIT5      # if (luachon != 3) EXIT5
173      # duyệt qua mảng array3
174      la    $s1, array3($0)    #s1 = &array3[0]
175      lw    $s2, size3($0)     #s2 = size3
176
177      add   $t5, $s1, $t8      #$t5= $t1+ $t8
178      lb    $t6, 0($t5)       #$t6= $t5
179      # xuất phần tử cần tìm
180      li $v0, 1
181      move $a0, $t6
182      syscall
183  EXIT5:
184      # kết thúc chương trình
185      li $v0, 10
186      syscall
187
188
189
190
```

Line: 178 Column: 26 ☒ Show Line Numbers

Hình 8

Hình 6,7,8: chọn mảng và phần tử cần tìm trong mảng đó rồi xuất ra

```
Mars Messages  Run I/O
Cac phan thu cua mang array1 la: 5 6 7 8 1 2 3 9 10 4
Cac phan thu cua mang array2 la: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Cac phan thu cua mang array3 la: 17 17 17 17 17 17 17 17
Nhap mang can tim (mang 1,2,3): 1
Nhap chi so cua phan tu can tim trong mang: 7
Phan tu can tim la: 9
-- program is finished running --
```

Hình 9: Xuất kết quả chương trình

## 2. Bài tập

Nhập từ bàn phím 5 số nguyên, in ra cửa sổ I/O của MARS theo từng yêu cầu sau:

- ✓ Số chan/le lớn nhất
- ✓ Tổng các số chan/le trong mảng

```
Lab5_bai1 Lab5_bai2
1  .data
2      S1: .asciiz    "\nNhap 5 so nguyen can tinh toan: "
3      S2: .asciiz    "\nSo le lon nhat la: "
4      S3: .asciiz    "\nSo chan lon nhat la: "
5      S4: .asciiz    "\nTong cac so le la: "
6      S5: .asciiz    "\nTong cac so chan la: "
7      Size: .word 5
8      Arr: .word 5
9  .text
10     li $v0, 4
11     la $a0, S1
12     syscall
13     # nhap cac phan tu cua mang
14     la $s0, Arr          #s0 = &arr[0]
15     lw $t0, Size         #$t0 = size
16
17     sll $t1, $t0, 2      #$t1=$t0 * 4
18
19     add $s1, $s0, $t1    #s1=$s0 + $t1
20 LOOP1:
21     bge $s0, $s1, EXIT   #if ($s0>=$s1) EXIT
22
23     li $v0, 5
24     syscall
25     sw $v0, ($s0)
```

```
Lab5_bai1 Lab5_bai2
25      sw $v0, ($s0)
26
27      addi $s0, $s0, 4
28      j LOOP1      #nhay toi lenh LOOP1
29 EXIT:
30      # xuất mảng
31      la $s0, Arr    #s0 tro den array
32      sll $t1, $t0, 2 # $t1 = $t0 * 4
33      add $s1, $s0, $t1
34 LOOP2:
35      bge $s0, $s1, EXIT2    #if ($s0 >= $s1) EXIT2
36
37      li $v0, 1
38      lw $a0, 0($s0)    #$a[0] = arr[i]
39      syscall
40
41      li $v0, 11 #Space
42      li $a0, 0x20
43      syscall
44
45      addi $s0, $s0, 4
46      j LOOP2      #nhay toi lenh LOOP2
47 EXIT2:
48      # xu ly cac phan tu trong mang
49      la $s0, Arr
```

Line: 50 Column: 29 ☒ Show Line Numbers

```
Lab5_bai1 Lab5_bai2
49      la $s0, Arr
50      sll $t1, $t0, 2      #$t1=$t0*4
51      add $s1, $s0, $t1
52      li $s4, 0 # Khoi tao s4 = sumOdd = 0
53      li $s5, 0 # Khoi tao s5 = sumEven = 0
54      li $s2, -2147483648
55      li $s3, -2147483648
56  LOOP3:
57      bge $s0, $s1, EXIT3   #if ($s0>=$s1) EXIT3
58
59      lw $t2, 0($s0)
60
61      andi $t3, $t2, 1      #$t3=$t2+1;
62      beq $t3, $0, IF_CHAN  #if ($t3==0) IF_CHAN
63      add $s4, $s4, $t2     #$s4+= $t2
64      bgt $s2, $t2, JUMP
65      move $s2, $t2
66      j JUMP #nhay toi lenh JUMP
67  IF_CHAN:
68      add $s5, $s5, $t2
69      bgt $s3, $t2, JUMP
70      move $s3, $t2
71  JUMP:
72      addi $s0, $s0, 4
73      j LOOP3 #nhay toi lenh LOOP3

Line: 50 Column: 29 ☒ Show Line Numbers
```

```

Edit  Execute
Lab5_bai1  Lab5_bai2
73      j LOOP3 #nhay toi lenh LOOP3
74  EXIT3:
75      #xuat chuoai S2
76      li $v0, 4
77      la $a0, S2
78      syscall
79      #xuat ra so le lon nhat
80      li $v0, 1
81      move $a0, $s2
82      syscall
83      # xuat chuoai S3
84      li $v0, 4
85      la $a0, S3
86      syscall
87      #xuat ra so chan lon nhat
88      li $v0, 1
89      move $a0, $s3
90      syscall
91      # xuat chuoai S4
92      li $v0, 4
93      la $a0, S4
94      syscall
95      #xuat ra tong cac so le
96      li $v0, 1
97      move $a0, $s4

```

Line: 50 Column: 29 ☒ Show Line Numbers

```

97      move $a0, $s4
98      syscall
99      # xuat chuoai S5
100     li $v0, 4
101     la $a0, S5
102     syscall
103     # xuat ra tong cac so chan
104     li $v0, 1
105     move $a0, $s5
106     syscall
107
108

```

Line: 50 Column: 29 ☒ Show Line Numbers

Các hình trên thực hiện yêu cầu xuất ra số chẵn, lẻ lớn nhất, tổng chẵn lẻ

```
Nhap 5 so nguyen can tinh toan: 3
4
2
6
5
3 4 2 6 5
So le lon nhat la: 5
So chan lon nhat la: 6
Tong cac so le la: 8
Tong cac so chan la: 12
-- program is finished running (dropped off bottom) --
```

Clear

Hình 10: Xuất kết quả của chương trình ở trên  
✓ Sắp xếp mảng tăng/giảm