

---

# HO CHI MINH UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

---



## COMPUTER ARCHITECTURE

### PRACTICAL SESSION - WEEK 3

LECTURERS: ASSOC. PROF. DR. CUONG PHAM QUOC

MR. KIEU DO NGUYEN BINH

CLASS: CC04      GROUP: 8

Members: Nguyễn Minh Hùng

Student ID: 1952737

Phạm Nhật Hoàng

Student ID: 1952737

Nguyễn Bá Minh Hưng

Student ID: 1952737

**Question 1.** Write a MIPS program with the following requirements:

1. Declare an integer array with 10 synthetic data elements.
2. Calculate the sum of all array elements.
3. Print the result to the terminal.

```
1 .text
2 .globl main
3
4 main:
5
6 la      $s0,array      # load array's address in $s0
7 li      $t0,0          # initialize sum = 0
8
9 li      $t1,0          # count = 0
10 loop:
11 beq     $t1,10,exit
12 lw      $t2,0($s0)
13 add     $t0,$t0,$t2
14 addi    $s0,$s0,4
15 addi    $t1,$t1,1
16 j      loop
17
18 exit:
19 li      $v0,4          # print_string
20 la      $a0,msg
21 syscall
22 li      $v0,1
23 move    $a0,$t0        # print sum
24 syscall
25
26
27
28
29
30 .data
31 msg: .asciiz "Result: "
32 array: .word 1,2,3,4,5,6,7,8,9,10
```

mips1.asm

**Question 2.** Write a MIPS program with the following requirements:

1. Declare an integer array with 10 synthetic data elements.
2. Calculate the sum of all odd elements (a[1], a[3],...).
3. Calculate the sum of all even elements (a[0], a[2],...).
4. Print the results to the terminal.

```

1 .text
2 .globl main
3
4 main:
5
6 la      $s0,array      # load array's address in $s0
7 li      $t0,0          # initialize even_sum = 0
8 li      $t1,0          # initialize odd_sum = 0
9 li      $t2,0          # count = 0
10 loop:
11 beq     $t2,10,exit    # count < 10 ?
12 lw      $t3,0($s0)
13
14 xori    $t4,$t2,1      # count XOR 1 == count + 1 ? -> even
15 addi    $t5,$t2,1
16
17 beq     $t4,$t5,even
18 add     $t1,$t1,$t3
19 j      next
20 even:
21 add     $t0,$t0,$t3
22 next:
23 addi    $t2,$t2,1
24 addi    $s0,$s0,4
25 j      loop
26
27 exit:
28
29 li      $v0,4          # print_string
30 la      $a0,msg
31 syscall
32 li      $v0,1
33 move    $a0,$t0        # print even_sum
34 syscall
35 li      $v0,4          # print space
36 la      $a0,msr
37 syscall
38 li      $v0,1
39 move    $a0,$t1        # print odd_sum
40 syscall
41
42
43
44
45 .data
46 msg: .asciiz "Result: Even: "
47 msr: .asciiz "\t Odd: "
48 array: .word 1,2,3,4,5,6,7,8,9,10

```

mips2.asm

**Question 3.** Write a MIPS program that receives 10 integer numbers from users through the terminal and store these numbers into an array. Print the sum of all array elements to the terminal.

```

1 .text
2 .globl main
3
4 main:
5
6 li      $v0,4
7 la      $a0,msg      #Print string
8 syscall
9
10 la      $s0,arr
11 li      $t1,0        #counter for loop $t1: i
12
13 loop:
14 beq     $t1,10,exit   # i <10
15
16 li      $v0,5
17 syscall      #input
18 move    $t0,$v0
19
20 sw      $t0,0($s0)    #store
21 addi    $s0,$s0,4
22
23 addi    $t1,$t1,1
24 j loop
25
26 exit:
27 li      $t1,0
28 li      $t3,0
29
30 addi    $s0,$s0,-4
31
32
33 loop1:   #loop to compute sum
34 beq     $t1,10,exit1
35
36 lw      $t2,0($s0)
37 add     $t3,$t3,$t2
38 addi    $s0,$s0,-4
39
40 addi    $t1,$t1,1
41
42 j loop1
43
44 exit1:
45 li      $v0,4
46 la      $a0,msr      #Print string
47 syscall
48
49 li      $v0,1
50 move    $a0,$t3
51 syscall
52
53 .data
54 msg:    .asciiz "Input: \n"
55 msr:    .asciiz "Sum = "
56 arr:    .word 0:10

```

mips3.asm

**Question 4.** Write a MIPS program with the following requirements:

1. Declare an integer array with 10 synthetic data elements.
2. Print a sentence to terminal to request an integer number that is greater than 0 and less than 10 (assume that user strictly follow this rule).
3. Print value of the element at index collected from the previous step.

```
1 .text
2 .globl main
3
4 main:
5
6 li    $v0,4          # print_string
7 la    $a0, msg
8      syscall
9
10 li    $v0,5          # read int
11      syscall
12 move  $t1,$v0        # $t1 : index from user
13
14 la    $s0,array      # load array's address in $s0
15
16
17 loop:
18 beq   $t1,0,exit
19 addi  $s0,$s0,4
20 addi  $t1,$t1,-1
21 j     loop
22
23 exit:
24 lw    $t2,0($s0)
25
26 li    $v0,4          # print_string
27 la    $a0, msg
28      syscall
29 li    $v0,1
30 move  $a0,$t2        # print result
31      syscall
32
33
34
35
36
37 .data
38 msg: .asciiz "Enter the index: "
39 msg: .asciiz "Result: "
40 array: .word 1,2,3,4,5,6,7,8,9,10
```

mips5.asm

**Question 5.** Write a MIPS program that reverses an 10 elements integer array. For example, the array initially stores 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 , the program will change the array to be 19, 17, 15, 13, 11, 9, 7, 5, 3, 1.

```
1 .text
2 .globl main
3
4 main:
5
6 li    $t1,0      # counter i=0
7 la    $s0,arr    # load address arr to $s0 : arr[0]
8
9 addi  $s1,$s0,36  # $s1: arr[9]
10
11 loop:
12 beq   $t1,5,exit # check condition i < 5 ?
13
14 # swap 2 elements
15 lw    $t0,0($s1)
16 lw    $t2,0($s0)
17
18 sw    $t0,0($s0)
19 sw    $t2,0($s1)
20
21 addi  $s0,$s0,4
22 addi  $s1,$s1,-4
23
24 addi  $t1,$t1,1  # i=i+1
25
26 j     loop
27
28 exit:
29
30 la    $s3,arr    # load address of arr[0] to $s3
31
32 li    $v0,4
33 la    $a0,mstr    #Print " Output array: "
34 syscall
35
36 li    $t1,0      # counter i=0
37
38 loop1:
39 beq   $t1,10,exit1 # check condition i < 10 ?
40 lw    $t3,0($s3)   # load elements from arr
41
42 li    $v0,1
43 move  $a0,$t3      # print the elements
44 syscall
45
46 li    $v0,4
47 la    $a0,mst      #Print space
48 syscall
49
50 addi  $s3,$s3,4
51 addi  $t1,$t1,1
52 j     loop1
53
54 exit1:
```

```
55 | li    $v0,4
56 | la    $a0,msg    #Print endlne
57 | syscall
58 |
59 | .data
60 | arr: .word 1,3,5,7,9,11,13,15,17,19
61 | msr: .asciiz "Output array: "
62 | mst: .asciiz "  "
63 | msg: .asciiz "\n"
```

mips4.asm