

# Báo cáo thực hành Kiến trúc máy tính – IT3280

Họ và Tên	Thân Cát Ngọc Lan
MSSV	20225646

## Assignment 1:

- Giá trị khởi tạo:  $i = 1, j = 2, x = 2, y = 2, z = 2$ ;
- Sử dụng công cụ gỡ rối, debug, chạy từng lệnh và dừng lại

	Bước chạy	Thanh ghi thay đổi	Giá trị ban đầu	Giá trị mới
Khởi tạo	1	s1	0x00000000	0x00000001
	2	s2	0x00000000	0x00000002
	3	t1	0x00000000	0x00000002
	4	t2	0x00000000	0x00000002
	5	t3	0x00000000	0x00000002
Chạy chương trình	6			
	7			
	8	t1	0x00000002	0x00000003
	9	t3	0x00000002	0x00000001
	10			
	11			

Nhận xét: lệnh nhảy j endif đã được thực hiện → 2 câu lệnh trong miền else không được thực hiện.

## Assignment 2:

- Giá trị khởi tạo:  $n=4, \text{step}=1, a=\{1,2,3,4\}$
- Sử dụng công cụ gỡ rối, debug, chạy từng lệnh và dừng lại

	Bước chạy	Thanh ghi thay đổi	Giá trị ban đầu	Giá trị mới
Khởi tạo	1,2	s2, at	0x00000000	0x10010000
	3	s3	0x00000000	0x00000004
	4	s4	0x00000000	0x00000001
	5			
	6			
Chạy chương trình	1	t2	0x00000000	0x00000001
	2			
	3,4,5	t1	0x00000000	0x10010000

	6	t0	0x00000000	0x00000001
	7	s5	0x00000000	0x00000001
	8	s1	0x00000000	0x00000001

Tương tự như trên ta có:

Bước chạy	t2	t1	t0	s5	s1
9-16	0x00000001	0x10010004	0x00000002	0x00000003	0x00000002
17-24	0x00000001	0x10010008	0x00000003	0x00000006	0x00000003
25-32	0x00000001	0x1001000c	0x00000004	0x0000000a	0x00000004
33	0x00000000				

Nhận xét: vòng lặp chạy 4 lần.

### Assignment 3:

- Giá trị ban đầu: b=2, a=3;
- Sử dụng công cụ gỡ rối, debug, chạy từng lệnh và dừng lại

	Bước chạy	Thanh ghi thay đổi	Giá trị ban đầu	Giá trị mới
Khởi tạo	1,2	at, s0	0x00000000	0x10010000
	3	s1	0x00000000	0x00000001
	4	s3	0x00000000	0x00000002
	5	s2	0x00000000	0x00000003
	6	t1	0x00000000	0x00000001
	7	t2	0x00000000	0x00000002
test=0	1			
	2	s2	0x00000000	0x00000004
test=1	1,2			
	3	s2	0x00000000	0x00000002
test=2	1,2,3			
	4	s3	0x00000003	0x00000004

Nhận xét: chương trình chạy đúng chức năng câu lệnh switch case. Với các giá trị test khác lệnh nhảy j default được thực thi → kết thúc chương trình.

## mips1.asm

```

1  #Laboratory Exercise 3, Home Assignment 3
2  .data
3  test: .word 0
4  .text
5  la $s0, test #load the address of test variable
6  lw $s1, 0($s0) #load the value of test to register $t1
7  addi $s3, $zero, 2
8  addi $s2, $zero, 3
9  li $t0, 0 #load value for test case
10 li $t1, 1
11 li $t2, 2
12 beq $s1, $t0, case_0
13 beq $s1, $t1, case_1
14 beq $s1, $t2, case_2
15 j default
16 case_0: addi $s2, $s2, 1 #a=a+1
17 j continue
18 case_1: sub $s2, $s2, $t1 #a=a-1
19 j continue
20 case_2: add $s3, $s3, $s3 #b=2*b
21 j continue
22 default:
23 continue:

```

## Assignment 4:

a)  $i < j$ 

## mips1.asm\*

```

1  # assignment 4.a
2  .text
3          addi $s1, $zero, 1
4          addi $s2, $zero, 2
5  start:
6          bgt $s1, $s2, else
7          addi $t1, $t1, 1
8          addi $t3, $zero, 1
9          j endif
10 else:
11          addi $t2, $t2, -1
12          add $t3, $t3, $t3
13          endif:
14

```

b)  $i \geq j$

```
mips1.asm
1  # assignment 4.b
2  .text
3      addi $s1, $zero, 1
4      addi $s2, $zero, 2
5  start:
6      ble $s1, $s2, else
7      addi $t1, $t1, 1
8      addi $t3, $zero, 1
9      j endif
10 else:
11     addi $t2, $t2, -1
12     add  $t3, $t3, $t3
13 endif:
14
```

c)  $i+j < 0$

```
mips1.asm
1  # assignment 4.c
2  .text
3      addi $s1, $zero, 1
4      addi $s2, $zero, 2
5      add $s3, $s1, $s2
6  start:
7      bgez $s3, else
8      addi $t1, $t1, 1
9      addi $t3, $zero, 1
10     j endif
11 else:
12     addi $t2, $t2, -1
13     add  $t3, $t3, $t3
14 endif:
15
```

d)  $i+j > m+n$

```

mips1.asm
1  # assignment 4.d
2  .text
3      addi $s1, $zero, 1
4      addi $s2, $zero, 2
5      add $s3, $s1, $s2
6      addi $s4, $zero, 3
7      addi $s5, $zero, 4
8      add $s6, $s4, $s5
9  start:
10     blt $s3, $s6, else
11     addi $t1, $t1, 1
12     addi $t3, $zero, 1
13     j endif
14  else:
15     addi $t2, $t2, -1
16     add $t3, $t3, $t3
17  endif:
18

```

## Assignment 5:

a)

```

mips1.asm
1  # assignment 5.a
2  .data
3      a: .word 1, 2, 0, -4, 5, 6, -2, 8
4  .text
5      la $s2, a
6      addi $s3, $zero, 5 # n = 5
7      addi $s4, $zero, 1
8      sub $s3, $s3, $s4 # mang co chi so bat dau tu 0
9      addi $s5, $zero, 0 # sum = 0
10     addi $s1, $zero, -1 # i = 0
11  loop:
12     add $s1, $s1, $s4 # i=i+step
13     add $t1, $s1, $s1 # t1=2*s1
14     add $t1, $t1, $t1 # t1=4*s1
15     add $t1, $t1, $s2 # t1 store the address of A[i]
16     lw $t0, 0($t1) # load value of A[i] in $t0
17     add $s5, $s5, $t0 # sum=sum+A[i]
18     slt $t2, $s1, $s3
19     bne $t2, 0, loop
20

```

b)

```

# assignment 5.b
.data
    a: .word 1, 2, 0, -4, 5, 6, -2, 8
.text
    la $s2, a
    addi $s3, $zero, 5
    addi $s4, $zero, 1
    addi $s5, $zero, 0 # sum = 0
    addi $s1, $zero, -1 # i = 0
loop:
    add $s1, $s1, $s4 #i=i+step
    add $t1, $s1, $s1 #t1=2*s1
    add $t1, $t1, $t1 #t1=4*s1
    add $t1, $t1, $s2 #t1 store the address of A[i]
    lw $t0, 0($t1) #load value of A[i] in $t0
    add $s5, $s5, $t0 #sum=sum+A[i]
    slt $t2, $s1, $s3
    beq $t2, 1, loop

```

c)

```

mips1.asm*
1  # assignment 5.c
2  .data
3      a: .word 1, 2, 0, -4, 5, 6, -2, 8
4  .text
5      la $s2, a
6      addi $s3, $zero, 5 # n = 5
7      addi $s4, $zero, 1
8      sub $s3, $s3, $s4 # mang co chi so bat dau tu 0
9      addi $s5, $zero, 0 # sum = 0
10     addi $s1, $zero, -1 # i = 0
11 loop:
12     add $s1, $s1, $s4 #i=i+step
13     add $t1, $s1, $s1 #t1=2*s1
14     add $t1, $t1, $t1 #t1=4*s1
15     add $t1, $t1, $s2 #t1 store the address of A[i]
16     lw $t0, 0($t1) #load value of A[i] in $t0
17     add $s5, $s5, $t0 #sum=sum+A[i]
18     bgt $s5, $zero, loop
19

```

d)

```
# assignment 5.d
.data
    a: .word 0, 1, 2, 0, -4, 5, 6, -2, 8
.text
    la $s2, a
    addi $s3, $zero, 5 # n = 5
    addi $s4, $zero, 1
    sub $s3, $s3, $s4 # mang co chi so bat dau tu 0
    addi $s5, $zero, 0 # sum = 0
    addi $s1, $zero, -1 # i = 0
loop:
    add $s1, $s1, $s4 #i=i+step
    add $t1, $s1, $s1 #t1=2*s1
    add $t1, $t1, $t1 #t1=4*s1
    add $t1, $t1, $s2 #t1 store the address of A[i]
    lw $t0, 0($t1) #load value of A[i] in $t0
    add $s5, $s5, $t0 #sum=sum+A[i]
    beq $t0, $zero, loop
```

## Assignment 6:

```

mips1.asm
1  # assignment 6
2  .data
3      a: .word 1, 3, 5, -4, -5, -6, 2, 8, 7, 0, -18
4  .text
5      addi    $s0, $zero, 11        # số phần tử của mảng: n = 11
6      addi    $s1, $zero, 0         # i=0
7      addi    $s2, $zero, 0         # max GTTD
8      la      $s3, a                # vị trí phần tử thứ i
9      addi    $s4, $zero, 0         # vị trí phần tử có trị tuyệt đối lớn nhất: vt
10 start:
11     slt      $at, $s1, $s0         #kết thúc vòng lặp khi i>=n
12     beqz     $at, endloop          #lấy giá trị a[i] vào $t0
13     lw       $t0, 0($s3)           #bỏ qua tính trị tuyệt đối khi a[i]>0
14     bgtz     $t0, skip_abs         #tính trị tuyệt đối của a[i]
15     sub      $t0, $zero, $t0
16 skip_abs:
17     slt      $at, $s2, $t0         #so sánh |a[i]| với giá trị max
18     beqz     $at, next             #gán max=a[i]
19     addi     $s2, $t0, 0           #gán vt=i
20     addi     $s4, $s1, 0
21 next:
22     addi     $s3, $s3, 4           #chuyển đến phần tử tiếp theo
23     addi     $s1, $s1, 1           #tăng biến đếm
24     j        start                #quay về vị trí đầu vòng lặp
25 endloop:
26
27

```

Kết quả đầu ra:

\$s2: giá trị của trị tuyệt đối lớn nhất

\$s4: vị trí phần tử có trị tuyệt đối lớn nhất