Assignment 1

- Code

```
start:
addi $s1,$zero,1 #i=1
addi $s2,$zero,2 #j=2

slt $t0,$s2,$s1 # j<i
bne $t0,$zero,else # branch to else if j<i
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
j endif # skip "else" part
else: addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
endif:
```

- Register

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	0
$t1	9	1
$t2	10	0
$t3	11	1
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	1
$s2	18	2
$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		4194340
hi		0
lo		0

- Lệnh addi $s1,$zero,1
  ⇨ $s1=1 (i=1)
- Lệnh addi $S2,,$zero,2
  ⇨ $s2 = 2 (j=2)
- Lệnh slt $t0,$s2,$s1
  ⇨ $t0 = 0 (vì j<i false)
- Lệnh bne $t0,$zero,else

- Lệnh addi $t1 ,$t1,1
  ⇨ $t1 = 1 (x = x+1)
- Lệnh addi $t3, $zero,1
  ⇨ $t3 = 1 (z = 1)
- j endif
  ⇨ skip “else” part

Assignment 2

- code

```
.data
        A:      .word   1,2,3,4,5

#Laboratory 3, Home Assigment 2
.text
addi $s5, $zero, 0 # sum = 0
addi $s1, $zero, 0 # i = 0
addi $s3, $zero, 5 # n = 5
la $s2,A
addi $s4, $zero,1


loop: slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0
beq $t2, $zero, endloop
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]
add $s1, $s1, $s4 # i = i + step
j loop # goto loop
endloop:
```

- Register

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	5
$t1	9	268501008
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	5
$s2	18	268500992
$s3	19	5
$s4	20	1
$s5	21	15
$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		4194364
hi		0
lo		0

- Lệnh addi $s5, $zero, 0
  ⇨ $s5 = 0 (sum = 0)

- Lệnh addi $s1, $zero, 0
  ⇨ $s1 = 0 (i = 0)

- Lệnh addi $s3, $zero, 5
  ⇨ $s3 = 5 (# n = 5)

- Lệnh la $s2,A
  ⇨ $s2 = 0x10010000

- Lệnh addi $s4, $zero,1
  ⇨ $s4 =1 (step =1)

- Lệnh loop: slt $t2, $s1, $s3
  ⇨ $t2 =1 (# $t2 = i < n ? 1 : 0)

- Lệnh beq $t2, $zero, endloop
  ⇨ $t2 != $zero => tiếp tục vòng lặp

- Lệnh add $t1, $s1, $s1
  add $t1, $t1, $t1 ($t1 = 4$s1)
  ⇨ $t1 = 0

- Lệnh add $t1, $t1, $s2
  ⇨ $t1 = 0x10010000 (# $t1 store the address of A[i])

- Lệnh lw $t0, 0($t1)
  ⇨ $t0 = 1 (# load value of A[i] in $t0)

- Lệnh add $s5, $s5, $t0
  ⇨ $s5 = 1 (# sum = sum + A[i])

- Lệnh add $s1, $s1, $s4
  ⇨ $s1 =1 (# i = i + step)

- j loop # goto loop
  ⇨ tiến hành lặp lại vòng lặp cho tới khi $s1>= $s3 (I >= n)

Assignment 3

- Code

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

- Register

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	0
$t1	9	1
$t2	10	2
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	268500992
$s1	17	1
$s2	18	-1
$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		4194368
hi		0
lo		0

- Lệnh la $s0, test
  ⇨ $s0 = 268500992 (#load the address of test variable)

- Lệnh lw $s1, 0($s0)
  ⇨ $s1 = 1 (#load the value of test to register $t1)

- Lệnh li $t0 0
  li $t1, 1
  li $t2, 2
  ⇨ $t0 = 0, $t1 = 1, $t2 = 2

- beq $s1, $t0, case_0
  ⇨ $s1 != $t0 =>không lọt vào case_0

- Lệnh beq $s1, $t1, case_1
  ⇨ $s1 = $t1 => case_1

- Lệnh case_1: sub $s2, $s2, $t1
  ⇨ $s2 =-1 (#a=a-1)

Assignment 4

a. i<j

```
start:
addi $s1,$zero,1 #i=1
addi $s2,$zero,2 #j=2

slt $t0,$s1,$s2 # i<j 1
bne $t0,$zero,else # branch to else if
addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
j endif # skip “else” part
else:
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
endif:
```

b. i>= j

```
start:
addi $s1,$zero,1 #i=1
addi $s2,$zero,2 #j=2

slt $t0,$s1,$s2 # i < j
bne $t0,$zero,else # branch to else if
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
```

```
j endif # skip “else” part
else: addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
endif:
```

c. i + j<=0

```
start:
addi $s1,$zero,1 #i=1
addi $s2,$zero,-2 #j=-2
add $s3,$1,$s2 # i+j

slt $t0,$zero,$s3 #  i +j >0
bne $t0,$zero,else # branch to else if
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
j endif # skip “else” part
else: addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
endif:
```

d. i + j < m + n

```
start:
addi $s1,$zero,1 #i=1
addi $s2,$zero,-2 #j=-2
add $s3,$1,$s2 # i+j
```

```
addi $s4,$zero,4 # m=4
addi $s5,$zero,5 # n=5
add $s6 ,$s4,$s5 # m+n

slt $t0,$s3,$s6 #  i +j < m + n
bne $t0,$zero,else # branch to else if
addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z

j endif # skip “else” part
else:
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
endif:
```

Assignment 5

a. i < n

```
.data
    A:      .word 1,2,3,4,5

#Laboratory 3, Home Assigment 2
.text
addi $s5, $zero, 0 # sum = 0
addi $s1, $zero, 5 # i = 0
```

```
addi $s3, $zero, 5 # n = 5
la $s2,A
addi $s4, $zero,1

loop: slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0
bne $t2, $zero, endloop
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]
add $s1, $s1, $s4 # i = i + step
j loop # goto loop
endloop:
```

b. i <= n

```
.data
        A:      .word 1,2,3,4,5

#Laboratory 3, Home Assigment 2
.text
addi $s5, $zero, 0 # sum = 0
addi $s1, $zero, 0 # i = 0
addi $s3, $zero, 5 # n = 5
la $s2,A
addi $s4, $zero,1

loop: slt $t2, $s3, $s1 # $t2 = n < i ? 1 : 0
beq $t2, $zero, endloop
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]
    add $s1, $s1, $s4 # i = i + step
    j loop # goto loop
    endloop:
```

c. sum >= 0

```
.data
    A:      .word -1,-2,-3,-4,5,6,7

#Laboratory 3, Home Assigment 2
.text
addi $s5, $zero, 0 # sum = 0
addi $s1, $zero, 0 # i = 0
addi $s3, $zero, 7 # n = 5
la $s2,A
addi $s4, $zero,1

loop:
slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0
beq $t2, $zero, endloop

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]
add $s1, $s1, $s4 # i = i + step

slt $t3 ,$s5,$zero # $t3 = sum < 0 ? 1 : 0
beq $t3, $zero, endloop # $t3 = 0 => sum>=0 => end loop

j   loop # goto loop
endloop:
```

d. A[i] == 0

```
    .data
            A:      .word 1,2,3,4,5,6,0,7

    #Laboratory 3, Home Assigment 2
    .text
    addi $s5, $zero, 0 # sum = 0
    addi $s1, $zero, 0 # i = 0
    addi $s3, $zero, 7 # n = 5
    la $s2,A
    addi $s4, $zero,1

    loop:
    slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0
    beq $t2, $zero, endloop

    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

    beq $t0, $zero, endloop # $t0 == 0 ( A[i] ==0 )  => end loop

    add $s5, $s5, $t0 # sum = sum + A[i]
```

```
        add $s1, $s1, $s4 # i = i + step
        j   loop # goto loop
        endloop:
```

Assignment 6

```
.data
NhapN: .asciiz "Nhap so phan tu mang: "
Xuat_Mang: .asciiz "Mang vua nhap la: \n"
Xuat: .asciiz "Nhap phan tu thu "
Xuat_Max: .asciiz "Phan tu co gia tri tuyet doi lon nhat la : "

tong: .word 0
a: .word 0:100

.text
.globl main
main:


la $a0, NhapN     # in NhapN
li $v0, 4
syscall


li $v0, 5       # nhap n va luu vao $s0
```

```
syscall
move $s0, $v0

li $t0, 0       # khoi tao vong lap , i = 0
la $a1, a     #load a vao $a1

 li $s1,0       # khoi tao phan tu max =0

Enter:

    blt $t0, $s0, enter_Element             # Neu i < n thi enter_Element
    j out
enter_Element:

    la $a0, Xuat         # print string Xuat
    li $v0, 4
    syscall



    li $v0, 1       # print index
    move $a0, $t0
    syscall



    li $v0, 5       # nhap so nguyen va luu vao mang
    syscall
    sw $v0, ($a1)
```

```
    addi $t0, $t0, 1              #Tang chi so
    addi $a1, $a1, 4



    abs $s2,$v0      # bien ve gia tri tuyet doi: $s2 = |v0=a[i]|

    # so sanh voi p tu max
    slt $t2,$s1,$s2          # $t2 = $s1 < $s2 ( max <a[i]) ? 1: 0
    beq $t2,$zero,Enter  # t2=0 (  max>=a[i] ) => quay lai enter
    add $s1,$s2,$zero    # t2=1 (  max < a[i] ) => gan $s1=$s2 ( max =
a[i] )

    j Enter

out:

    la $a0,Xuat_Max          #print string Xuat_Max
    li $v0,4
    syscall

    li $v0, 1  # print phan tu max
    move $a0, $s1
    syscall
```