Assignment 1:

```
1  #Laboratory Exercise 7 Home Assignment 1
2  .text
3  main:
4          li      $a0, -45        #load input parameter
5          jal     abs             #jump and link to abs procedure
6          nop
7          add     $s0, $zero, $v0
8          li      $v0, 10         #terminate
9          syscall
10 endmain:
11 abs:
12         sub     $v0,$zero,$a0     #put -(a0) in v0; in case (a0)<0
13         bltz  $a0,done          #if (a0)<0 then done
14         nop
15         add   $v0,$a0,$zero     #else put (a0) in v0
16 done:
17         jr    $ra
```

$a0	4	-45
$s0	16	45

$a0	4	25
$s0	16	25

Assignment 2:

mips2.asm

```
1  #Laboratory Exercise 7, Home Assignment 2
2  .text
3  main:
4          li    $a0,2            #load test input
5          li    $a1,6
6          li    $a2,9
7          jal   max              #call max procedure
8          nop
9          li    $v0, 10          #terminate
10         syscall
11 endmain:
12
13 max:
14         add     $v0,$a0,$zero   #copy (a0) in v0; largest so far
15         sub     $t0,$a1,$v0     #compute (a1)-(v0)
16         bltz    $t0,okay        #if (a1)-(v0)<0 then no change
17         nop
18         add     $v0,$a1,$zero   #else (a1) is largest thus far
19
20 okay:
21         sub     $t0,$a2,$v0     #compute (a2)-(v0)
22         bltz    $t0,done        #if (a2)-(v0)<0 then no change
23         nop
24         add     $v0,$a2,$zero   #else (a2) is largest overall
25 done:
26         jr      $ra             #return to calling program
27
```

Khi bắt đầu so sánh sẽ lần lượt gán các giá trị tăng dần cho $v0

$v0	2	2
$v0	2	6
$v0	2	9

Và kết quả cuối cùng của $v0 đó là giá trị lớn nhất trong 3 giá trị được lưu trên $s0, $s1, $s2

Assignment 3:

Khởi tạo giá trị cho thanh ghi $s0, $s1

$s0	16	10
$s1	17	5

Sau khi chạy trương trình, giá trị 2 thanh ghi sẽ hoán đổi.

$s0	16	5
$s1	17	10

Assignment 4:

Thanh ghi $v0 sẽ lưu lại kết quả của các phép tính 1!, 2!, 3!, …, n!

Sau đó kết quả của n! sẽ được lưu vào thanh ghi $a1

- *N = 3*

$a1	5	0x00000006

- *N = 6*

$a1	5	0x000002d0

- *N = 8*

$a1	5	0x00009d80

Assignment 5:

```
1   .data
2           largest_str: .asciiz "Largest: "
3           smallest_str: .asciiz "\nSmallest: "
4           s1: .ascii ", "
5
6   .text
7           li $s0, 5
8           li $s1, 3
9           li $s2, 1
10          li $s3, -8
11          li $s4, 2
12          li $s5, -4
13          li $s6, 16
14          li $s7, 7
15
16  main:
17          # store in stack
18          la $fp, 0($sp)
19          addi $sp, $sp, -4
20          sw $s0, 0($sp)
21          addi $sp, $sp, -4
22          sw $s1, 0($sp)
23          addi $sp, $sp, -4
24          sw $s2, 0($sp)
25          addi $sp, $sp, -4
26          sw $s3, 0($sp)
27          addi $sp, $sp, -4
28          sw $s4, 0($sp)
29          addi $sp, $sp, -4
30          sw $s5, 0($sp)
31          addi $sp, $sp, -4
32          sw $s6, 0($sp)
33          addi $sp, $sp, -4
34          sw $s7, 0($sp)
35          addi $sp, $sp, -4
36          sw $s0, 0($sp)          # base value = $s0
37          addi $sp, $sp, -4
38          sw $zero, 0($sp)        # base index = 0
39
```

```
40          jal find_largest_number
41
42          move $t0, $v0
43          div $t0, $t0, 4
44          la $a0, largest_str
45          li $v0, 4
46          syscall
47          move $a0, $t0
48          li $v0, 1
49          syscall
50          la $a0, s1
51          li $v0, 4
52          syscall
53          move $a0, $v1
54          li $v0, 1
55          syscall
```

```
57          jal find_smallest_number
58
59          move $t0, $v0
60          div $t0, $t0, 4
61
62          la $a0, smallest_str
63          li $v0, 4
64          syscall
65          move $a0, $t0
66          li $v0, 1
67          syscall
68          la $a0, s1
69          li $v0, 4
70          syscall
71          move $a0, $v1
72          li $v0, 1
73          syscall
```

```
75  end_main:       li $v0, 10
76                          syscall
77
78  find_largest_number:
79          la $t0, 0($sp)
80          addi $t0, $t0, 8
81
82  loop:   slt $t9, $t0, $fp
83                  beq $t9, $zero, end_func
84                  lw $t1, 4($sp)
85                  lw $t2, 0($t0)
86                  addi $t0, $t0, 4
87                  slt $t3, $t1, $t2
88                  beq $t3, $zero, end_loop
89                  sw $t2, 4($sp)
90                  sub $t4, $fp, $t0
91                  sw $t4, 0($sp)
92  end_loop:
93                  j loop
```

```
94  end_func:
95                  lw $v0, 0($sp)
96                  lw $v1, 4($sp)
97                  jr $ra
98
99  find_smallest_number:
100         la $t0, 0($sp)
101         addi $t0, $t0, 8
102
103 loop_:  slt $t9, $t0, $fp
104                 beq $t9, $zero, end_func_
105                 lw $t1, 4($sp)
106                 lw $t2, 0($t0)
107                 addi $t0, $t0, 4
108                 slt $t3, $t2, $t1
109                 beq $t3, $zero, end_loop_
110                 sw $t2, 4($sp)
111                 sub $t4, $fp, $t0
112                 sw $t4, 0($sp)
113 end_loop_:
114                 j loop_
115 end_func_:
116                 lw $v0, 0($sp)
117                 lw $v1, 4($sp)
118                 jr $ra
119
```

Kết quả thu được

```
Largest: 6, 16
Smallest: 3, -8
```