

## Computer Architecture Lab – Week 2's report

Full name	Student ID	Works
Trần Quốc Việt	1953097	1 + 6 + combine
Lý Kim Phong	1952916	2 + 4
Nguyễn Đình Khương Duy	1952207	3 + 5

### Question 1.

```
.text

        .globl main
main:
    # Print string msg1
    li    $v0,4          # print_string syscall code = 4
    la    $a0, msg1      # load the address of msg
    syscall

    # Get input from user and save
    li    $v0,5          # read_int syscall code = 5
    syscall
    move  $t0, $v0        # syscall results returned in $v0
    addi  $t1, $t0, 1

    # Print string msg2
    li    $v0,4          # print_string syscall code = 4
    la    $a0, msg2      # load the address of msg
    syscall

# Print result
    li    $v0,1          # print_int syscall code = 1
    move  $a0, $t1        # int to print must be loaded into $a0
    syscall

# Start .data segment (data!)
.data
msg1: .asciiz "Enter number :  "
msg2: .asciiz "Result after increased that number by 1 is :  "
```

### Question 2.

```
.text

        .globl main
main:
    # Print string msg1
    li    $v0,4          # print_string syscall code = 4
    la    $a0, msg1      # load the address of msg
    syscall

    # Get input A from user and save
```

```

    li    $v0,5          # read_int syscall code = 5
    syscall
    move  $t0, $v0        # syscall results returned in $v0

    # Print string msg2
    li    $v0,4          # print_string syscall code = 4
    la    $a0, msg2      # load the address of msg
    syscall

    # Get input B from user and save
    li    $v0,5          # read_int syscall code = 5
    syscall
    move  $t1, $v0        # syscall results returned in $v0

    # Add Processing
    add   $a1, $t0, $t1

    # Print string msg3
    li    $v0,4          # print_string syscall code = 4
    la    $a0, msg3      # load the address of msg
    syscall

# Print sum
    li    $v0,1          # print_int syscall code = 1
    move  $a0, $a1        # int to print must be loaded into $a0
syscall

# Start .data segment (data!)
.data
msg1: .ascii "Enter A:  "
msg2: .ascii "Enter B:  "
msg3: .ascii "Sum of A and B:  "

```

### Question 3.

```

.text

    .globl    main
main:
    # Print string msg1
    li    $v0,4          # print_string syscall code = 4
    la    $a0, msg1      # load the address of msg
    syscall

    # Get input A from user and save
    li    $v0,5          # read_int syscall code = 5
    syscall
    move  $t0,$v0        # syscall results returned in $v0

    # Print string msg2
    li    $v0,4          # print_string syscall code = 4
    la    $a0, msg2      # load the address of msg2
    syscall

    # Get input B from user and save
    li    $v0,5          # read_int syscall code = 5

```

```

syscall
move  $t1,$v0          # syscall results returned in $v0

# Print string msg3
li    $v0,4             # print_string syscall code = 4
la    $a0, msg3         # load the address of msg3
syscall

# Get input C from user and save
li    $v0,5             # read_int syscall code = 5
syscall
move  $t2,$v0           # syscall results returned in $v0

# Print string msg4
li    $v0,4             # print_string syscall code = 4
la    $a0, msg4         # load the address of msg2
syscall

# Get input D from user and save
li    $v0,5             # read_int syscall code = 5
syscall
move  $t3,$v0           # syscall results returned in $v0

# Math!
add   $t4, $t0, $t1     # t4 = A + B
sub   $t5, $t2, $t3     # t5 = C - D
addi  $t5, $t5, -2      # t5 = t5 - 2
sub   $t4, $t4, $t5     # t4 = t4 - t5

# Print string msg5
li    $v0, 4
la    $a0, msg5
syscall

# Print sum
li    $v0,1             # print_int syscall code = 1
move  $a0, $t4          # int to print must be loaded into $a0
syscall

# Print \n
li    $v0,4             # print_string syscall code = 4
la    $a0, newline
syscall

# Math!
add   $t0, $t0, $t1     # t0 = A + B
add   $t4, $t0, $t0     # t4 = t0 + t0
add   $t4, $t4, $t0     # t4 = t4 + t0
add   $t2, $t2, $t3     # t2 = C + D
add   $t2, $t2, $t2     # t2 = t2 + t2
sub   $t4, $t4, $t2

# Print string msg6
li    $v0, 4
la    $a0, msg6
syscall

```

```

        # Print sum
        li    $v0,1          # print_int syscall code = 1
        move  $a0,$t4        # int to print must be loaded into $a0
syscall

# Start .data segment (data!)
.data
msg1: .asciiz "Enter A:  "
msg2: .asciiz "Enter B:  "
msg3: .asciiz "ENTER C:  "
msg4: .asciiz "ENTER D:  "
msg5: .asciiz "f =      "
msg6: .asciiz "g =      "
newline: .asciiz "\n"

```

#### Question 4.

```

.text

.globl    main
main:
    # Print string msg1
    li    $v0,4          # print_string syscall code = 4
    la    $a0,msg1       # load the address of msg
    syscall

    # Get input A from user and save
    li    $v0,5          # read_int syscall code = 5
    syscall
    move  $t0,$v0        # syscall results returned in $v0

    # Print string msg2
    li    $v0,4          # print_string syscall code = 4
    la    $a0,msg2       # load the address of msg2
    syscall

    # Get input B from user and save
    li    $v0,5          # read_int syscall code = 5
    syscall
    move  $t1,$v0        # syscall results returned in $v0

    # Print string msg3
    li    $v0,4          # print_string syscall code = 4
    la    $a0,msg3       # load the address of msg
    syscall

    # Get input C from user and save
    li    $v0,5          # read_int syscall code = 5
    syscall
    move  $t2,$v0        # syscall results returned in $v0

    # Print string msg4
    li    $v0,4          # print_string syscall code = 4
    la    $a0,msg4       # load the address of msg2

```

```

syscall

# Get input D from user and save
li    $v0,5      # read_int syscall code = 5
syscall
move  $t3,$v0    # syscall results returned in $v0

# Print string msg5
li    $v0,4      # print_string syscall code = 4
la    $a0,msg5   # load the address of msg2
syscall

# Get input E from user and save
li    $v0,5      # read_int syscall code = 5
syscall
move  $t4,$v0    # syscall results returned in $v0


# Print sum
li    $v0,1      # print_int syscall code = 1
move  $a0,$t4    # int to print must be loaded into $a0
syscall

# Print \n
li    $v0,4      # print_string syscall code = 4
la    $a0,newline
syscall

# Print sum
li    $v0,1      # print_int syscall code = 1
move  $a0,$t3    # int to print must be loaded into $a0
syscall

# Print \n
li    $v0,4      # print_string syscall code = 4
la    $a0,newline
syscall

# Print sum
li    $v0,1      # print_int syscall code = 1
move  $a0,$t2    # int to print must be loaded into $a0
syscall

# Print \n
li    $v0,4      # print_string syscall code = 4
la    $a0,newline
syscall

# Print sum
li    $v0,1      # print_int syscall code = 1
move  $a0,$t1    # int to print must be loaded into $a0
syscall

# Print \n
li    $v0,4      # print_string syscall code = 4
la    $a0,newline

```

```

        syscall

# Print sum
li      $v0,1      # print_int syscall code = 1
move    $a0, $t0   # int to print must be loaded into $a0
syscall

# Start .data segment (data!)
.data
msg1: .asciiz      "Enter A:  "
msg2: .asciiz      "Enter B:  "
msg3: .asciiz      "ENTER C:  "
msg4: .asciiz      "ENTER D:  "
msg5: .asciiz      "ENTER E:  "
newline: .asciiz    "\n"

```

### Question 5.

```

.text
.globl main

main:
    li $t0, 30          #load immediate 30 to $t1
    add $t1, $t0, $t0   #add 2 times of 30 to $t1
    add $t2, $t1, $t1   #add 2 times of $t1 (60) to $t2
    add $t2, $t2, $t2   #add 2 times of $t2 to itself (240)
    add $t2, $t2, $t1   #add 2 times of $t1 (60) to $t2, now $t2
is 300

    addi $t2, $t2, 66000 #add immediate 66000 to $t2, now $t2
is 66300
    subi $t2, $t2, 6000  #subtract immediate 6000 from $t2,
now $t2 is 60300
    addi $a0, $t2, 25    #add immediate 25 to $t2, now $t2 is
60325

    li $v0, 1           #syscall to print result out : 60325
    syscall
    li $v0, 10          #program termination
    syscall

```

### Question 6.

```

.text
.globl main
main:
# Print string msg1
li      $v0,4      # print_string syscall code = 4
la      $a0, msg1  # load the address of msg
syscall

# Get input a from user and save
li      $v0,5      # read_int syscall code = 5

```

```

syscall
move $t0, $v0          # syscall results returned in $v0

# Print string msg2
li    $v0,4             # print_string syscall code = 4
la    $a0, msg2         # load the address of msg
syscall

# Get input b from user and save
li    $v0,5             # read_int syscall code = 5
syscall
move  $t1, $v0          # syscall results returned in $v0

# Print string msg3
li    $v0,4             # print_string syscall code = 4
la    $a0, msg3         # load the address of msg
syscall

# Get input c from user and save
li    $v0,5             # read_int syscall code = 5
syscall
move  $t2, $v0          # syscall results returned in $v0

# Print string msg4
li    $v0,4             # print_string syscall code = 4
la    $a0, msg4         # load the address of msg
syscall

# Get input d from user and save
li    $v0,5             # read_int syscall code = 5
syscall
move  $t3, $v0          # syscall results returned in $v0

# Print string msg5
li    $v0,4             # print_string syscall code = 4
la    $a0, msg5         # load the address of msg
syscall

# Get input x from user and save
li    $v0,5             # read_int syscall code = 5
syscall
move  $t4, $v0          # syscall results returned in $v0

#Math
mul $t5, $t4, $t4 # t5 = x^2
mul $t6, $t5, $t4 # t6 = x^3
mul $t7, $t0, $t6 # t7 = a*x^3
mul $t8, $t1, $t5 # t8 = b*x^2
mul $t9, $t2, $t4 # t9 = c*x
add $t7, $t7, $t8 # t7 = a*x^3 + b*x^2
add $t7, $t7, $t9 # t7 = a*x^3 + b*x^2 + c*x
add $t7, $t7, $t3 # t7 = a*x^3 + b*x^2 + c*x + d

```

```

# Print string msg6
li    $v0, 4      # print_string syscall code = 4
la    $a0, msg6   # load the address of msg
syscall

# Print sum
li    $v0, 1      # print_int syscall code = 1
move  $a0, $t7    # int to print must be loaded into $a0
syscall

# Start .data segment (data!)
.data
msg1:  .asciiz  "Enter a:  "
msg2:  .asciiz  "Enter b:  "
msg3:  .asciiz  "Enter c:  "
msg4:  .asciiz  "Enter d:  "
msg5:  .asciiz  "Enter x:  "
msg6:  .asciiz  "f =      "
newline:  .asciiz  "\n"

```