Lab Report 05

Assignment 1

Code:

```
.data
test:       .asciiz "Hello World"

.text
            li      $v0, 4
            la      $a0, test
            syscall
```

Comments:

- The beggning address of the string is 0x10010000, which also corresponds to letter “H”.
- The content of the first 4 bytes: `H e l l`
- The content of the next 4 bytes: `o  W o`
- The content of the last 4 bytes: `r l d \0`

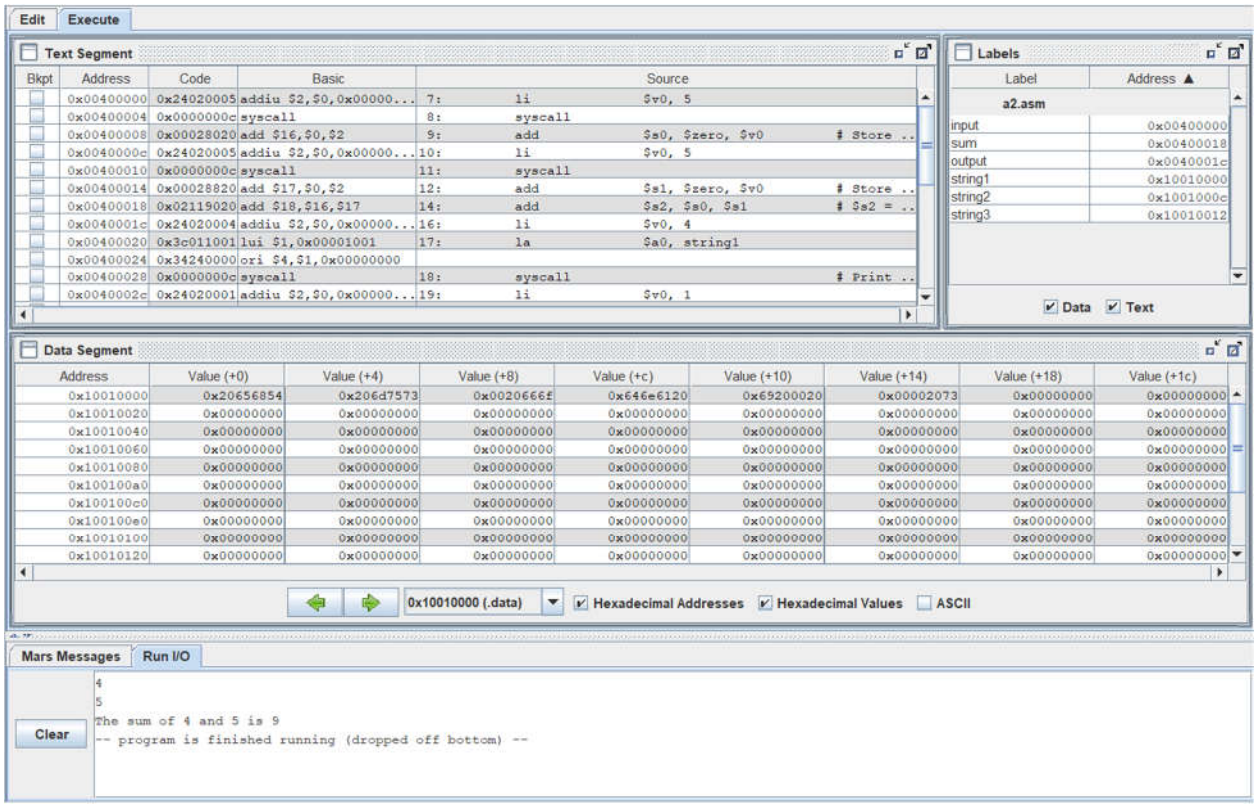
Assignment 2

Code:

```
.data
string1: .asciiz "The sum of "
string2: .asciiz " and "
string3: .asciiz " is "
.text
input:
        li          $v0, 5
        syscall
        add         $s0, $zero, $v0         # Store $s0
        li          $v0, 5
        syscall
        add         $s1, $zero, $v0         # Store $s1
sum:
```

```
        add     $s2, $s0, $s1       # $s2 = $s0 + $s1
output:
        li      $v0, 4
        la      $a0, string1
        syscall                     # Print   "The sum of "
        li      $v0, 1
        add     $a0, $zero, $s0     # $a0 = $s0
        syscall                     # Print (s0)
        li      $v0, 4
        la      $a0, string2
        syscall                     # Print " and "
        li      $v0, 1
        add     $a0, $zero, $s1     # $a0 = $s1
        syscall                     # Print (s1)
        li      $v0, 4
        la      $a0, string3
        syscall                     # Print " is "
        li      $v0, 1
        add     $a0, $zero, $s2     # $a0 = sum
        syscall                     # Print (sum)
```

Result:



Assignment 3

Code:

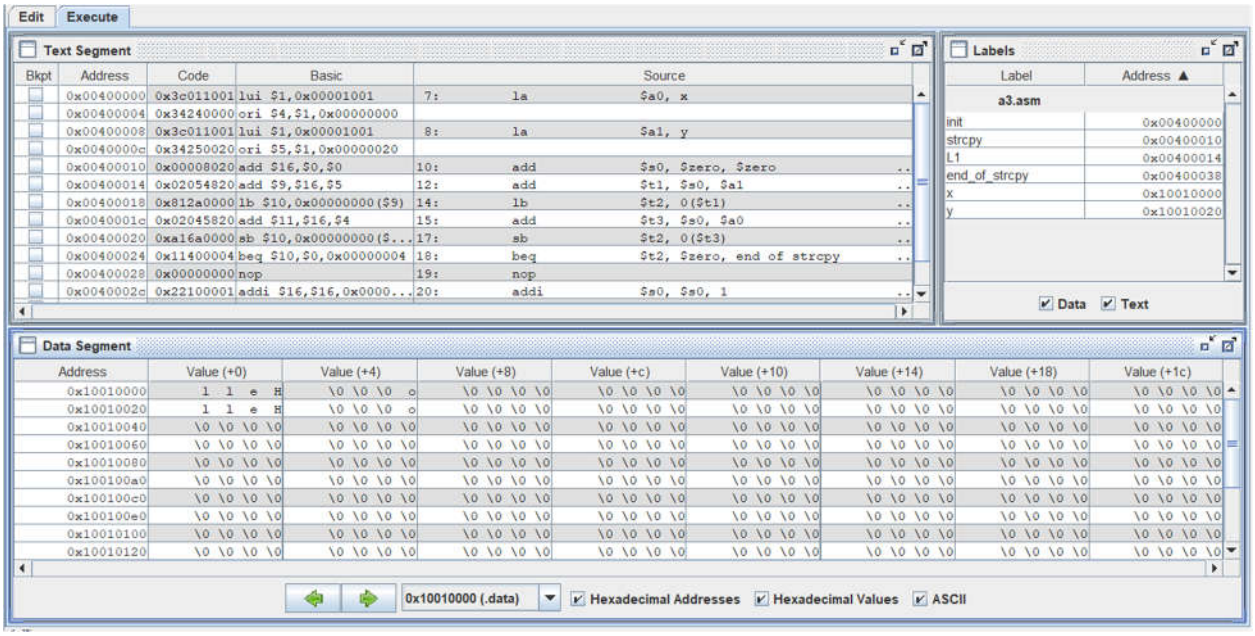
```
.data
x:      .space  32                      # destination string x, empty
y:      .asciiz "Hello"                 # source string y
```

```
.text
init:
        la      $a0, x
        la      $a1, y
strcpy:
        add     $s0, $zero, $zero       # $s0 = i = 0
L1:
        add     $t1, $s0, $a1           # $t1 = $s0 + $a1 = i + y[0]
                                        #     = address of y[i]
        lb      $t2, 0($t1)             # $t2 = value at $t1 = y[i]
        add     $t3, $s0, $a0           # $t3 = $s0 + $a0 = i + x[0]
                                        #     = address of x[i]
        sb      $t2, 0($t3)             # x[i]= $t2 = y[i]
        beq     $t2, $zero, end_of_strcpy   # if y[i] == 0, exit
        nop
        addi    $s0, $s0, 1             # $s0 = $s0 + 1 <-> i = i + 1
        j       L1                      # next character
        nop
end_of_strcpy:
```

Comments:

- The red code is where to load the address of x & y into the corresponding registers $a0, $a1.

Result:



Assignment 4

Code:

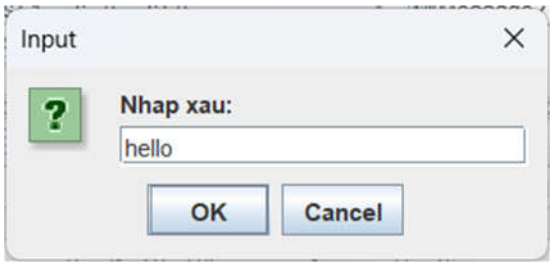
```
.data
string:     .space 50
Message1:   .asciiz "Nhap xau: "
Message2:   .asciiz "Do dai xau la: "

.text
main:
get_string:
            li      $v0, 54
            la      $a0, Message1
            la      $a1, string
            la      $a2, 50
            syscall
get_length:
            la      $a0, string             # $a0 = address(string[0])
            add     $t0, $zero, $zero       # $t0 = i = 0
```

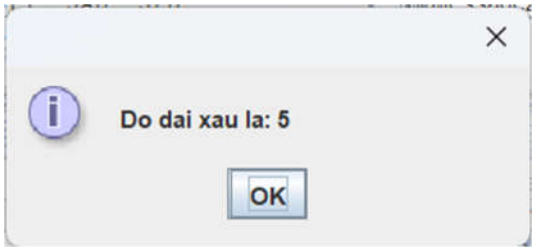
```
check_char:
        add     $t1, $a0, $t0       # $t1 = $a0 + $t0
                                    #     = address(string[i])
        lb      $t2, 0($t1)         # $t2 = string[i]
        beq     $t2, $zero, end_of_str  # is null char?
        addi    $t0, $t0, 1         # $t0 = $t0 + 1 -> i = i + 1
        j       check_char
end_of_str:
end_of_get_length:
print_length:
        li      $v0, 56
        la      $a0, Message2
        subi    $a1, $t0, 1
        syscall
```

Comments:

- The red code is the implementation of the required parts.

Result:





Assignment 5

Code:

```
.data
```

```
string:         .space 21

.text
main:
init:
                la      $a0, string             # $a0 = address(string[0])
                add     $t0, $zero, $zero       # $t0 = i = 0
                li      $s0, 20                 # Maximum size of string
scan_char:
                add     $t1, $a0, $t0           # $t1 = $a0 + $t0
                                                #     = address(string[i])

                li $v0, 12
                syscall
                sb      $v0, 0($t1)
                beq     $v0, '\n', end_of_string    # Stop if '\n' is inputted

                addi    $t0, $t0, 1
                beq     $t0, $s0, end_of_string     # Stop if maximum size is reached
                j       scan_char
end_of_string:
                add     $s1, $zero, $a0         # Store address of string in $s1
                                                # $a0 is dangerous to store in long
term!
reverse_string:

                subi    $t0, $t0, 1
                bltz    $t0, exit
```

```
        add     $t1, $s1, $t0           # $t1 = $s1 + $t0
                                        #     = address(string[i])

        li      $v0, 11
        lb      $a0, 0($t1)             # Load string[i]
        syscall                         # Print string[i]

        j       reverse_string
exit:
```

Result: