Assignment 1:

```
#assignment 1

.data
        test:   .asciiz "Hello World"
.text
        li       $v0, 4
        la       $a0, test
        syscall
```

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24020004	addiu $2,$0,0x00000004	5: li $v0, 4
	0x00400004	0x3c011001	lui $1,0x00001001	6: la $a0, test
	0x00400008	0x34240000	ori $4,$1,0x00000000	
	0x0040000c	0x0000000c	syscall	7: syscall

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)
0x10010000	l l e H	o W o	\0 d l r	\0 \0 \0 \0
0x10010020	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

Mars Messages | Run I/O

Clear

```
Reset: reset completed.

Hello World
-- program is finished running (dropped off bottom) --

Hello World
-- program is finished running (dropped off bottom) --
```

Assignment 2:

```
.data
test:    .asciiz "The sum of "
test1:   .asciiz " and "
test2:   .asciiz " is "

.text
        # Gan s0,s1
        li      $s0, 8
        li      $s1, 7
        # In "The sum of"
        li      $v0, 4
        la      $a0, test
syscall
        #In s0
        li      $v0, 1
        la      $a0, 0($s0)
syscall
        #In "and"
        li      $v0, 4
        la      $a0, test1
syscall
        #In s1
        li      $v0, 1
        la      $a0, 0($s1)
syscall
        # In "is"
```

```
        li      $v0, 4
        la      $a0, test2
syscall
        #In tong
        add     $s3, $s0, $s1
        li      $v0, 1
        la      $a0, 0($s3)
syscall
```

```
The sum of 8 and 7 is 15
-- program is finished running (dropped off bottom) --
```

Assignment 3:

ass1.asm | mips2.asm | ass2.asm | mips4.asm* | ass4.asm | ass6.asm

```
1   .data
2   x: .space 32
3   y: .asciiz "Hello World 123"
4
5   .text
6   strcpy:
7           add $s0, $zero, $zero
8           la  $a1, y #Dia chi y
9           la  $a0, x #Dia chi x
10
11  Li:
12          add     $t1, $s0, $a1
13          lb      $t2, 0($t1)
14          add     $t3, $s0, $a0
15          sb      $t2, 0($t3)
16          beq     $t2, $zero, end
17          nop
18          addi    $s0, $s0, 1
19          j       Li
20          nop
21          end:
22  #in
23          li      $v0, 4
24          la      $a0, x
25          syscall
```

```
Hello World 123
-- program is finished running (dropped off bottom) --
```

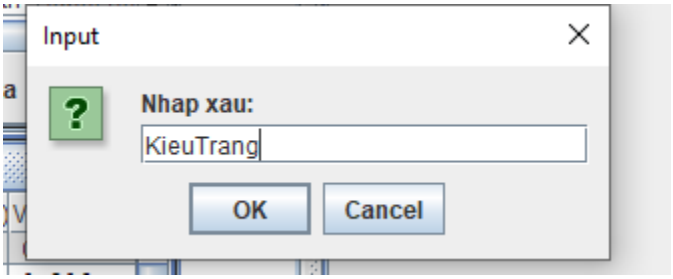
Assignment 4:

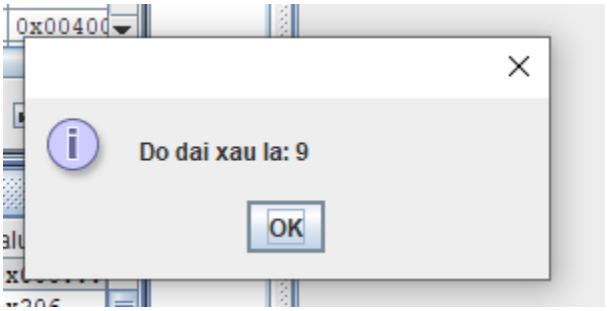
```
.data
string:  .space 50
Mess1:                  .asciiz "Nhap xau: "
Mess2:                  .asciiz "Do dai xau la: "

.text
main:
get_string:     #todo
                #Nhap xau
                li      $v0, 54
                la      $a0,Mess1
                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
                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:
                #In chieu dai
                addi    $t0, $t0, -1
                li      $v0, 56
                la      $a0, Mess2
                la      $a1, 0($t0)
                syscall
```





Assignment 5:

```
.data
string: .space 50
enter: .asciiz "\n"
Message1: .asciiz "Nhap vao xau: "
str: .space 20
.text
main:
#in thong bao nhap vao xau
    li $v0, 4
     la $a0, Message1
     la $s2,enter
     syscall

    add $s0,$zero,$zero         #$s0=i=0
    la $a3,string #a0 = address of var string
    li $s1,19     #max_string = 20, bat dau = 0.
loop:
    li $v0, 12
     syscall
     add $t3,$s0,$a3
     bge $s0,$s1,end_loop #length>20 thi endloop
     beq $v0,'\n',end_loop # is enter char?
     sb $v0,0($t3) # save entered char into string a0
    nop
        addi $s0,$s0,1 #index=length +=1
     j loop
end_loop:

start:
beq  $s0,$zero,end2
addi $s0,$s0,-1
add  $t4,$a3,$s0
lbu $a0,0($t4)
li $v0, 11
syscall
j start
end2:
```

```
Nhap vao xau: Trangefff
fffegnarT
-- program is finished running (dropped off bottom) --
```