

Resolución de problemas propuestos para entregar:
Mateo Julian Pretelli Rodriguez

repositorio github:
<https://github.com/mateopretelli/MipsRepositorio.git>

1) Dado el siguiente ejemplo de programa ensamblador:

```
.data
dato: .byte 3 #inicializo una posición de
memoria a 3
.text
.globl main # debe ser global
main: lw $t0, dato($0)
```

Indica las etiquetas, directivas y comentarios que aparecen en el mismo.

Etiquetas / Directivas / Comentarios

2)

.data 0x10000000

A: .space 80

B: .space 80

3)

.data 0x10001000

palabra1: .space 4

byte: .space 1

.align 2

palabra2: .space4

4)

.data

palabra: .word 3

byte: .byte 0x10

.align 2

byte4: .space 4

byte20: .byte 20

5)

.data

texto1: .ascii "Esto es un problema"

texto2: .byte 69, 115, 116, 111, 32, 117, 110, 32, 112, 114, 111, 98, 108, 101,
109, 97

texto3: .word 69, 115, 116, 111, 32, 117, 110, 32, 112, 114, 111, 98, 108,
101, 109, 97

6)

.data

matrizF: .word 1, 2, 3

 .word 4, 5, 6

 .word 7, 8, 9

matrizC: .word 1, 4, 7

 .word 2, 5, 8

 .word 3, 6, 9

7)

.data 0x10000000

V: .word 10, 20, 25, 500, 3

.text

main: lw \$s0, V(\$0)

 lw \$s1, V+4(\$0)

 lw \$s2, V+8(\$0)

 lw \$s3, V+12(\$0)

 lw \$s4, V+16(\$0)

8)

.data 0x10000000

V: .word 10, 20, 25, 500, 3

.text

```
main:
la $t0, v($0)
li $t1, 0x10010000
li $t2, 0x10010014
loop: lw $s0, 0($t0)
sw $s0, 0($t1)
add $t0, $t0, 4
add $t1, $t1, 4
blt $t1, $t2, loop
```

```
9)
.data 0x10000000
palabra: .word 0x10203040
```

```
.text
lw $t0, palabra($0)
li $t1, 0x10010000
li $t2, 0x10010004
loop: lb $s0, 0($t0)
sb $s0, 0($t1)
add $t0, $t0, 1
add $t1, $t1, 1
blt $t1, $t2, loop
```

```
10)
.data
palabra: .word 0x10203040
```

```
.text
la $t0, palabra($0)
lh $s0, 0($t0)
lh $s1, 2($t0)
sh $s1, 0($t0)
sh $s0, 2($t0)
```

11)

.data 0x10010002

byte: .byte 0x10, 0x20, 0x30, 0x40

.data 0x10010102

space: .space 4

.text

la \$t0, byte

la \$t1, space

andi \$t2, \$t2, 0

add \$t2, \$t1, 4

loop: lb \$s0, 0(\$t0)

sb \$s0, 0(\$t1)

add \$t0, \$t0, 1

add \$t1, \$t1, 1

blt \$t1, \$t2, loop

12)

.data 0x10000000

V: .byte 10, 20

.text

la \$t0, V(\$0)

lb \$s0, 0(\$t0)

lb \$s1, 1(\$t0)

add \$s2, \$s0, \$s1

sb \$s2, 2(\$t0)

13)

.data 0x10000000

V: .word 18, -1215

.text

la \$t0, V(\$0)

li \$t1, 0x10010000

```
lw $s0, 0($t0)
lw $s1, 4($t0)
div $s0, $s0, 5
div $s1, $s1, 5
div $s3, $s0, $1
sb $s3, 0($t1)
```

```
14)
.data 0x10000000
bits: .word 0xabcd12bd
```

```
.text
lw $s0, bits($0)
andi $s0, $s0, 0x2BB
```

```
15)
.data 0x10000000
bits: .word 0xffffffff41
```

```
.text
lw $s0, bits($0)
xor $s0, $s0, 0x144
```

```
16)
.data
bits: .word 0x1237
```

```
.text
lw $s0, bits($0)
sll $s0, $s0, 5
```