1) 133 - 1

```
.data
v: .word 1, 2, 4, 5
n: .word 4
x: .word 2
t: .word 16

.text
j main

este_putere:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addi $fp, $sp, 4
	subu $sp, $sp, 4
	sw $s0, 0($sp)
	subu $sp, $sp, 4
	sw $s1, 0($sp)

	lw $s0, 4($fp) #t
	lw $s1, 0($fp) #v[i]

	beq $s0, 1, exit_while
	beq $s1, 1, cont

	li $t0, 1
	while:
		bge $t0, $s0, exit_while
		mul $t0, $t0, $s1
		j while

	exit_while:
		bne $t0, $s0, cont
		li $v0, 1
		j cont

	cont:
		lw $s1, -12($fp)
		lw $s0, -8($fp)
		lw $fp, -4($fp)
		addu $sp, $sp, 12
		jr $ra
```

```
eval:
        subu $sp, $sp, 4
        sw $fp 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)

        lw $s0, 12($fp) #adr vector
        lw $s1, 8($fp) #n
        lw $s2, 4($fp) #x
        lw $s3, 0($fp) #t

        beqz $s1, exit

        lw $t0, 0($s0) #v[i]
        subu $sp, $sp, 4
        sw $s3, 0($sp)
        subu $sp, $sp, 4
        sw $t0, 0($sp)
        li $v0, 0
        jal este_putere
        addu $sp, $sp, 8

        li $t1, 1
        subu $t1, $t1, $v0
        lw $t0, 0($s0)
        mul $t0, $t0, $s2
        addi $t0, $t0, 1
        mul $t1, $t1, $t0
        add $v1, $v1, $t1

        addi $s0, $s0, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subi $s1, $s1, 1
```

```
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)
        jal eval
        addu $sp, $sp, 16

        exit:
                lw $s3, -24($fp)
                lw $s2, -20($fp)
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 24
                jr $ra

main:
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, x
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, t
subu $sp, $sp, 4
sw $t0, 0($sp)
jal eval
addu $sp, $sp, 16
move $a0, $v1
li $v0, 1
syscall
li $v0, 10
syscall
```

2) 133 - 2

```
.data
v: .word 10, 11, 12, 13
```

```
w: .word 2, 3, 5, 6
n: .word 4
p: .word 3

.text
j main

cel_putin_p_div:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addi $fp, $sp, 4
	subu $sp, $sp, 4
	sw $s0, 0($sp)
	subu $sp, $sp, 4
	sw $s1, 0($sp)

	lw $s0, 4($fp) #v[i] + w[i]
	lw $s1, 0($fp) #p

	li $t1, 2 #nr div
	li $t2, 2
	div $t3, $s0, 2

	for1:
		bgt $t2, $t3, exit_for1
		rem $t4, $s0, $t2
		bnez $t4, cont1
		addi $t1, $t1, 1
		j cont1

		cont1:
			addi $t2, $t2, 1
			j for1

	exit_for1:
		blt $t1, $s2, exit1
		li $v0, 1
		j exit1

		exit1:
			lw $s1, -12($fp)
			lw $s0, -8($fp)
			lw $fp, -4($fp)
```

```
                addu $sp, $sp, 12
                jr $ra

eval:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)

        lw $s0, 12($fp) #adr v
        lw $s1, 8($fp) #adr w
        lw $s2, 4($fp) #n
        lw $s3, 0($fp) #p

        li $t0, 0 #contor

        for:
                beq $t0, $s2, exit_for
                lw $t1, 0($s0) #v[i]
                lw $t2, 0($s1) #w[i]
                add $t1, $t1, $t2

                subu $sp, $sp, 4
                sw $t1, 0($sp)
                subu $sp, $sp, 4
                sw $s3, 0($sp)
                li $v0, 0
                jal cel_putin_p_div
                addu $sp, $sp, 8

                lw $t1, 0($s0)
                lw $t2, 0($s1)
                rem $t1, $t1, $t2
                mul $t1, $t1, $v0
```

```
                add $v1, $v1, $t1

                addi $s0, $s0, 4
                addi $s1, $s1, 4
                addi $t0, $t0, 1
                j for

        exit_for:
                lw $s3, -24($fp)
                lw $s2, -20($fp)
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 24
                jr $ra

main:
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, w
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, p
subu $sp, $sp, 4
sw $t0, 0($sp)
jal eval
addu $sp, $sp, 16
move $a0, $v1
li $v0, 1
syscall
li $v0, 10
syscall
```

3) 135 - 1

```
.data
v: .word 1, 2, 4, 6
w: .word 2, 7, 1, 3
n: .word 4
```

```
x: .word 2
y: .word 3

.text:
j main

putere:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addi $fp, $sp, 4
	subu $sp, $sp, 4
	sw $s0, 0($sp)
	subu $sp, $sp, 4
	sw $s1, 0($sp)

	lw $s0, 4($fp) #v[i]
	lw $s2, 0($fp) #w[i]

	li $v0, 1 #putere
	li $t0, 0 #contor

	for:
		beq $t0, $s2, exit
		mul $v0, $v0, $s0
		addi $t0, $t0, 1
		j for

	exit:
		lw $s1, -12($fp)
		lw $s0, -8($fp)
		lw $fp, -4($fp)
		addu $sp, $sp, 12
		jr $ra

modif:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addi $fp, $sp, 4
	subu $sp, $sp, 4
	sw $ra, 0($sp)
	subu $sp, $sp, 4
	sw $s0, 0($sp)
	subu $sp, $sp, 4
```

```
sw $s1, 0($sp)
subu $sp, $sp, 4
sw $s2, 0($sp)
subu $sp, $sp, 4
sw $s3, 0($sp)
subu $sp, $sp, 4
sw $s4, 0($sp)

lw $s0, 16($fp) #adr v
lw $s1, 12($fp) #adr w
lw $s2, 8($fp) #n
lw $s3, 4($fp) #x
lw $s4, 0($fp) #y

beqz $s2, final

lw $t0, 0($s0) #v[i]
mul $t1, $s3, $s4 #xy
rem $t1, $t0, $t1 #restul imp la xy
bnez $t1, cont

subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, 0($s1) #w[i]
subu $sp, $sp, 4
sw $t0, 0($sp)
jal putere
addu $sp, $sp, 8

sw $v0, 0($s0)
j cont

cont:
        addi $s0, $s0, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        addi $s1, $s1, 4
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subi $s2, $s2, 1
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
```

```
                sw $s3, 0($sp)
                subu $sp, $sp, 4
                sw $s4, 0($sp)
                jal modif
                addu $sp, $sp, 20

        final:
                lw $s4, -28($fp)
                lw $s3, -24($fp)
                lw $s2, -20($fp)
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 28
                jr $ra

main:
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, w
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, x
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, y
subu $sp, $sp, 4
sw $t0, 0($sp)
jal modif
addu $sp, $sp, 20

lw $t0, n #contor
li $t1, 0 #adr mem
for1:
        beqz $t0, exit1
        lw $a0, v($t1)
        li $v0, 1
        syscall
```

```
        la $a0, ' '
        li $v0, 11
        syscall
        subi $t0, $t0, 1
        addi $t1, $t1, 4
        j for1

exit1:
        li $v0, 10
        syscall
```

4) 135 - 2

```
.data
v: .word 2, 4, 1, 7, 3
n: .word 5
a: .word 2

.text
j main

nr_f:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)

        lw $s0, 0($fp) #v[i]

        beq $s0, 1, e1

        rem $t0, $s0, 2
        beqz $t0, edivcu2

        mul $v0, $s0, 3
        addi $v0, $v0, 1
        j cont1

        e1:
                li $v0, 1
                j cont1

        edivcu2:
```

```
            div $v0, $s0, 2
            j cont1

        cont1:
            lw $s0, -8($fp)
            lw $fp, -4($fp)
            addu $sp, $sp, 8
            jr $ra

modif:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)

        lw $s0, 8($fp) #adr v
        lw $s1, 4($fp) #n
        lw $s2, 0($fp) #a

        for1:
            beqz $s1, final

            lw $t0, 0($s0) #v[i]
            subu $sp, $sp, 4
            sw $t0, 0($sp)
            jal nr_f
            addu $sp, $sp, 4

            rem $t1, $v0, 2
            bnez $t1, cont
            add $v0, $v0, $s2
            sw $v0, 0($s0)
            j cont

            cont:
                addi $s0, $s0, 4
```

```
                subi $s1, $s1, 1
                j for1

        final:
                lw $s2, -20($fp)
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 20
                jr $ra

main:
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, a
subu $sp, $sp, 4
sw $t0, 0($sp)
jal modif
addu $sp, $sp, 12

lw $t0, n #contor
li $t1, 0 #adr mem

for:
        beqz $t0, exit
        lw $a0, v($t1)
        li $v0, 1
        syscall
        la $a0, ' '
        li $v0, 11
        syscall
        subi $t0, $t0, 1
        addi $t1, $t1, 4
        j for

exit:
        li $v0, 10
        syscall
```

5) 141 - 1

```
.data
v: .word 2, 3, 7, 4
n: .word 4
x: .word 2
y: .word 2
z: .word 1

.text
j main

exactXDivizori:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addi $fp, $sp, 4
	subu $sp, $sp, 4
	sw $s0,0($sp)
	subu $sp, $sp, 4
	sw $s1, 0($sp)

	lw $s0, 4($fp) #x
	lw $s1, 0($fp) #v[i] + y

	li $t0, 2 #nrdiv
	li $t1, 2 #div
	div $t2, $s1, 2

	for:
		bgt $t1, $t2, exit
		rem $t3, $s1, $t1
		bnez $t3, cont
		addi $t0, $t0, 1
		j cont

		cont:
			addi $t1, $t1, 1
			j for

	exit:
		bne $t0, $s0, exit1
		li $v0, 1
		j exit1
```

```
        exit1:
                lw $s1, -12($fp)
                lw $s0, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 12
                jr $ra

eval:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)
        subu $sp, $sp, 4
        sw $s4, 0($sp)

        lw $s0, 16($fp) #adr v
        lw $s1, 12($fp) #n
        lw $s2, 8($fp) #x
        lw $s3, 4($fp) #y
        lw $s4, 0($fp) #z

        beqz $s1, final

        lw $t0, 0($s0) #v[i]
        add $t0, $t0, $s3 #v[i] + y
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $t0, 0($sp)
        li $v0, 0
        jal exactXDivizori
        addu $sp, $sp, 8
```

```
        lw $t0, 0($s0)
        sub $t0, $t0, $s4 #v[i] - z
        mul $v0, $v0, $t0
        add $v1, $v1, $v0

        addi $s0, $s0, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subi $s1, $s1, 1
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)
        subu $sp, $sp, 4
        sw $s4, 0($sp)
        jal eval
        addu $sp, $sp, 20

        final:
                lw $s4, -28($fp)
                lw $s3, -24($fp)
                lw $s2, -20($fp)
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 28
                jr $ra

main:
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, x
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, y
subu $sp, $sp, 4
```

```
sw $t0, 0($sp)
lw $t0, z
subu $sp, $sp, 4
sw $t0, 0($sp)
jal eval
addu $sp, $sp, 20

move $a0, $v1
li $v0, 1
syscall

li $v0, 10
syscall
```

6) 141 - 2

```
.data
v: .word 2, 3, 5, 8
n: .word 4
a: .word 2
c: .word 5
t: .word 5

.text
j main

f:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)

        lw $s0, 8($fp) #a
        lw $s1, 4($fp) #c
        lw $s2, 0($fp) #v[i]

        mul $v0, $s0, $s2
        rem $v0, $v0, $s1
```

```
        lw $s2, -16($fp)
        lw $s1, -12($fp)
        lw $s0, -8($fp)
        lw $fp, -4($fp)
        addu $sp, $sp, 16
        jr $ra

modif:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)
        subu $sp, $sp, 4
        sw $s4, 0($sp)

        lw $s0, 16($fp) #adr v
        lw $s1, 12($fp) #n
        lw $s2, 8($fp) #a
        lw $s3, 4($fp) #c
        lw $s4, 0($fp) #t

        for1:
                beqz $s1, final
                lw $t0, 0($s0) #v[i]
                bge $t0, $s4, cont
                subu $sp, $sp, 4
                sw $s2, 0($sp)
                subu $sp, $sp, 4
                sw $s3, 0($sp)
                subu $sp, $sp, 4
                sw $t0, 0($sp)
                jal f
                addu $sp, $sp, 12
                sw $v0, 0($s0)
```

```
                j cont

                cont:
                        subi $s1, $s1, 1
                        addi $s0, $s0, 4
                        j for1

        final:
                lw $s4, -28($fp)
                lw $s3, -24($fp)
                lw $s2, -20($fp)
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 28
                jr $ra

main:
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, a
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, c
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, t
subu $sp, $sp, 4
sw $t0, 0($sp)
jal modif
addu $sp, $sp, 20

lw $t0, n
li $1, 0 #adr mem
for:
        beqz $t0, exit
        lw $a0, v($t1)
        li $v0, 1
```

```
        syscall
        la $a0, ' '
        li $v0, 11
        syscall
        addi $t1, $t1, 4
        subi $t0, $t0, 1
        j for

exit:
        li $v0, 10
        syscall
```

7) 131 - 1

```
.data
str : .asciiz "abcdef"
vocale: "aeiou"
s: .asciiz "Sirul nu contine vocale"

.text
j main

este_vocala:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)

        lb $s0, 0($fp)
        li $t1, 0
        lb $t2, vocale($t1)

        for:
                beqz $t2, nuvocala
                beq $s0, $t2, vocala
                addi $t1, $t1, 1
                lb $t2, vocale($t1)
                j for

        nuvocala:
                li $v1, 0
                j exit1
```

```
        vocala:
                li $v1, 1
                j exit1

        exit1:
                lw $s0, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 8
                jr $ra

modif:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)

        lw $s0, 0($fp) #adr str
        lb $t0, 0($s0) #caract

        beqz $t0, exit

        subu $sp, $sp, 4
        sb $t0, 0($sp)
        jal este_vocala
        addu $sp, $sp, 4

        beqz $v1, cont
        li $v0, 1
        addi $t0, $t0, 1
        sb $t0, 0($s0)
        j cont

        cont:
                addi $s0, $s0, 1
                subu $sp, $sp, 4
                sw $s0, 0($sp)
                jal modif
                addu $sp, $sp, 4

        exit:
```

```
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 12
                jr $ra

main:
la $t0, str
subu $sp, $sp, 4
sw $t0, 0($sp)
jal modif
addu $sp, $sp, 4

beqz $v0, cont1
la $a0, str
li $v0, 4
syscall
j final

cont1:
        la $a0, s
        li $v0, 4
        syscall
        j final

final:
        li $v0, 10
        syscall
```

8) 131 - 2

```
.data
str1: .asciiz "ABCDZ"
ch: .asciiz "Z"
vocale: .asciiz "AEIOU"

.text
j main

este_consoana:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
```

```
        sw $s0, 0($sp)

        lb $s0, 0($fp) #caract
        li $t3, 0
        lb $t4, vocale($t3)

        for1:
                beqz $t4, consoana
                beq $s0, $t4, vocala
                addi $t3, $t3, 1
                lb $t4, vocale($t3)
                j for1

        consoana:
                li $v1, 1
                j exit

        vocala:
                li $v1, 0
                j exit

        exit:
                lw $s0, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 8
                jr $ra

strchr_const:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)

        lw $s0, 0($fp) #adr str1
        lb $s1, 4($fp) #ch
        li $t0, 0 #contor
        lb $t1, 0($s0) #caract sir
        li $v0, -1
```

```
        for:
                beqz $t1, final
                beq $t1, $s1, apare
                j cont

                apare:
                        lb $t2, 1($s0)
                        beqz $t2, da
                        subu $sp, $sp, 4
                        sb $t2, 0($sp)
                        jal este_consoana
                        addu $sp, $sp, 4
                        beq $v1, 1, da
                        j cont

                        da:
                                move $v0, $t0
                                j final

                        cont:
                                addi $t0, $t0, 1
                                addi $s0, $s0, 1
                                lb $t1, 0($s0)
                                j for

        final:
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 16
                jr $ra

main:
lb $t0, ch
subu $sp, $sp, 4
sb $t0, 0($sp)
la $t0, str1
subu $sp, $sp, 4
sw $t0, 0($sp)
jal strchr_const
addu $sp, $sp, 8
```

```
move $a0, $v0
li $v0, 1
syscall
li $v0, 10
syscall
```

9) 132 - 1

```
.data
v: .word 1, 2, 3, 4, 5
n: .word 5
r: .word 2
c: .word 3

.text
j main

putere_mod:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addi $fp, $sp, 4
	subu $sp, $sp, 4
	sw $s0, 0($sp)
	subu $sp, $sp, 4
	sw $s1, 0($sp)
	subu $sp, $sp, 4
	sw $s2, 0($sp)

	lw $s0, 0($fp) #r
	lw $s1, 4($fp) #c + v[i]
	lw $s2, 8($fp) #v[i]
	li $v0, 1 #putere
	li $t3, 0 #contor

	for1:
		beq $t3, $s1, exit1
		mul $v0, $v0, $s0
		addi $t3, $t3, 1
		j for1

	exit1:
		mul $v0, $v0, $s2
		sub $s1, $s1, $s2
		rem $v0, $v0, $s1
```

```
                lw $s2, -16($fp)
                lw $s1, -12($fp)
                lw $s0, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 16
                jr $ra

modif:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)

        lw $s0, 0($fp) #adr v
        lw $s1, 4($fp) #n
        lw $s2, 8($fp) #r
        lw $s3, 12($fp) #c

        beqz $s1, exit

        lw $t0, 0($s0) #v[i]
        rem $t1, $t0, 2
        beqz $t1, vocala
        j cont

        vocala:
                subu $sp, $sp, 4
                sw $t0, 0($sp)
                add $t2, $s3, $t0
                subu $sp, $sp, 4
                sw $t2, 0($sp)
                subu $sp, $sp, 4
                sw $s2, 0($sp)
```

```
            jal putere_mod
            addu $sp, $sp, 12
            sw $v0, 0($s0)
            j cont

        cont:
            subu $sp, $sp, 4
            sw $s3, 0($sp)
            subu $sp, $sp, 4
            sw $s2, 0($sp)
            subi $s1, $s1, 1
            subu $sp, $sp, 4
            sw $s1, 0($sp)
            addi $s0, $s0, 4
            subu $sp, $sp, 4
            sw $s0, 0($sp)
            jal modif
            addu $sp, $sp, 16

        exit:
            lw $s3, -24($fp)
            lw $s2, -20($fp)
            lw $s1, -16($fp)
            lw $s0, -12($fp)
            lw $ra, -8($fp)
            lw $fp, -4($fp)
            addu $sp, $sp, 24
            jr $ra

main:
lw $t0, c
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, r
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
jal modif
```

```
addu $sp, $sp, 16
lw $t0, n
mul $t0, $t0, 4
li $t1, 0 #contor

for:
	beq $t1, $t0, final
	lw $a0, v($t1)
	li $v0, 1
	syscall
	la $a0, ' '
	li $v0, 11
	syscall
	addi $t1, $t1, 4
	j for

final:
	li $v0, 10
	syscall
```

10) 132 - 2

```
.data
v: .word 1, 3, 5, 7, 8
w: .space 400
n: .word 5

.text
j main

media_aritmetica:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addi $fp, $sp, 4
	subu $sp, $sp, 4
	sw $s0, 0($sp)
	subu $sp, $sp, 4
	sw $s1, 0($sp)

	lw $s0, 0($fp) #v[i+1]
	lw $s1, 4($fp) #v[i]

	add $v0, $s0, $s1
	div $v0, $v0, 2
```

```
        lw $s1, -12($fp)
        lw $s0, -8($fp)
        lw $fp, -4($fp)
        addu $sp, $sp, 12
        jr $ra

creare:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)

        lw $s0, 0($fp) #adr v
        lw $s1, 4($fp) #adr w
        lw $s2, 8($fp) #n
        subi $s2, $s2, 1 #n - 1
        li $t0, 0 #contor v

        for1:
                beq $t0, $s2, exit
                lw $t1, 0($s0) #v[i]
                sw $t1, 0($s1) #pun v[i] in w[[i]
                addi $s1, $s1, 4
                addi $s0, $s0, 4
                lw $t2, 0($s0) #v[i+1]

                subu $sp, $sp, 4
                sw $t1, 0($sp)
                subu $sp, $sp, 4
                sw $t2, 0($sp)
                jal media_aritmetica
                addu $sp, $sp, 8

                sw $v0, 0($s1) #pun ma in w[i]
                addi $s1, $s1, 4
```

```
            addi $t0, $t0, 1
            j for1

        exit:
            sw $t2, 0($s1) #pun v[n] in w

            lw $s2, -20($fp)
            lw $s1, -16($fp)
            lw $s0, -12($fp)
            lw $ra, -8($fp)
            lw $fp, -4($fp)
            addu $sp, $sp, 20
            jr $ra

main:
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, w
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
jal creare
addu $sp, $sp, 12
lw $t0, n
add $t0, $t0, $t0
subi $t0, $t0, 1 #m = 2n - 1
mul $t0, $t0, 4
li $t1, 0 #contor

for:
        beq $t1, $t0, final
        lw $a0, w($t1)
        li $v0, 1
        syscall
        la $a0, ' '
        li $v0, 11
        syscall
        addi $t1, $t1, 4
        j for
```

```
final:
	li $v0, 10
	syscall
```

11) 142 - 1 MARE FLEX PE CAPUL MEU

```
.data
v: .word 16, 14, 10, 8, 7, 9, 3, 2, 4, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1 #indexare de la 1
n: .word 21

.text
j main

suma_drum_dreapta:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addi $fp, $sp, 4
	subu $sp, $sp, 4
	sw $s0, 0($sp)
	subu $sp, $sp, 4
	sw $s1, 0($sp)

	lw $s0, 0($fp) #adr v
	lw $s1, 4($fp) #i
	li $t1, 1 #contor

	for1: #mut adr lui v pe poz i-1
		beq $t1, $s1, exit1
		addi $s0, $s0, 4
		addi $t1, $t1, 1
		j for1

	exit1:
		lw $t2, 0($s0)
		while:
			beq $t2, -1, exit2
			add $v1, $v1, $t2
			mul $s1, $s1, 2
			addi $s1, $s1, 1

			for2:
				beq $t1, $s1, exit3
				addi $s0, $s0, 4
				addi $t1, $t1, 1
```

```
                j for2

            exit3:
                lw $t2, 0($s0)
                j while

        exit2:
            lw $s1, -12($fp)
            lw $s0, -8($fp)
            lw $fp, -4($fp)
            addu $sp, $sp, 12
            jr $ra

eval:
    subu $sp, $sp, 4
    sw $fp, 0($sp)
    addi $fp, $sp, 4
    subu $sp, $sp, 4
    sw $ra, 0($sp)
    subu $sp, $sp, 4
    sw $s0, 0($sp)
    subu $sp, $sp, 0
    sw $s1, 0($sp)

    lw $s0, 0($fp) #adr v
    lw $s1, 4($fp) #n
    li $t0, 1 #contor

    for:
        bgt $t0, $s1, final

        subu $sp, $sp, 4
        sw $t0, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        li $v1, 0
        jal suma_drum_dreapta
        addu $sp, $sp, 8

        add $v0, $v0, $v1
        addi $t0, $t0, 1
        j for
```

```
        final:
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 16
                jr $ra

main:
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
li $v0, 0
jal eval
addu $sp, $sp, 8
move $a0, $v0
li $v0, 1
syscall
li $v0, 10
syscall

12) 142 - 2
.data
v: .word 2, 4, 20, 3, 28
n: .word 5
m: .word 2
p: .word 2

.text
j main

exact_p_desc:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
```

```
        sw $s2, 0($sp)

        lw $s0, 0($fp) #v[i]
        lw $s1, 4($fp) #m
        lw $s2, 8($fp) #p
        li $t2, 0 #nr aparitii m in descomp

        while:
                rem $t3, $s0, $s1
                bnez $t3, exit2
                addi $t2, $t2, 1
                div $s0, $s0, $s1
                j while

        exit2:
                seq $v1, $t2, $s2

                lw $s2, -16($fp)
                lw $s1, -12($fp)
                lw $s0, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 16
                jr $ra

eval:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)

        lw $s0, 0($fp) #adr v
        lw $s1, 4($fp) #n
        lw $s2, 8($fp) #m
        lw $s3, 12($fp) #p
```

```
        li $t0, 0 #contor

        for:
                beq $t0, $s1, exit
                lw $t1, 0($s0) #v[i]

                subu $sp, $sp, 4
                sw $s3, 0($sp)
                subu $sp, $sp, 4
                sw $s2, 0($sp)
                subu $sp, $sp, 4
                sw $t1, 0($sp)
                jal exact_p_desc
                addu $sp, $sp, 12

                sub $t1, $t1, $s2 #v[i] - m
                mul $v1, $v1, $t1 #(v[i] - m) * v1
                add $v0, $v0, $v1 #adaug la suma
                addi $s0, $s0, 4
                addi $t0, $t0, 1
                j for

        exit:
                lw $s3, -24($fp)
                lw $s2, -20($fp)
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 24
                jr $ra

main:
lw $t0, p
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, m
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, v
```

```
subu $sp, $sp, 4
sw $t0, 0($sp)
li $v0, 0
jal eval
addu $sp, $sp, 16
move $a0, $v0
li $v0, 1
syscall
li $v0, 10
syscall
```

13) 134 - 1

```
.data
v: .word 7, 14, 4, 2, 19
n: .word 5
z: .word 5
p: .word 3

.text
j main

exact_p_binar:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)

        lw $s0, 0($fp) #v[i]
        lw $s1, 4($fp) #p
        li $t1, 0 #nr de 1

        while:
                beqz $s0, exit2
                rem $t2, $s0, 2
                seq $t3, $t2, 1
                add $t1, $t1, $t3
                div $s0, $s0, 2
                j while

        exit2:
```

```
                seq $v0, $t1, $s1

                lw $s1, -12($fp)
                lw $s0, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 12
                jr $ra

modif:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)

        lw $s0, 0($fp) #adr v
        lw $s1, 4($fp) #n
        lw $s2, 8($fp) #z
        lw $s3, 12($fp) #p

        beqz $s1, final
        lw $t0, 0($s0) #v[i]

        subu $sp, $sp, 4
        sw $s3, 0($sp)
        subu $sp, $sp, 4
        sw $t0, 0($sp)
        jal exact_p_binar
        addu $sp, $sp, 8

        beqz $v0, cont
        mul $t1, $s2, -1 #-z
        add $t1, $t1, 2 #2-z
        mul $t0, $t0, $t1
        sw $t0, 0($s0)
```

```
        j cont

        cont:
                subu $sp, $sp, 4
                sw $s3, 0($sp)
                subu $sp, $sp, 4
                sw $s2, 0($sp)
                subi $s1, $s1, 1
                subu $sp, $sp, 4
                sw $s1, 0($sp)
                add $s0, $s0, 4
                subu $sp, $sp, 4
                sw $s0, 0($sp)
                jal modif
                addu $sp, $sp, 16

        final:
                lw $s3, -24($fp)
                lw $s2, -20($fp)
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 24
                jr $ra

main:
lw $t0, p
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, z
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
jal modif
addu $sp, $sp, 16

lw $t0, n
```

```
mul $t0, $t0, 4 #n*4
li $t1, 0 #contor

for:
	beq $t1, $t0, exit
	lw $a0, v($t1)
	li $v0, 1
	syscall
	la $a0, ' '
	li $v0, 11
	syscall
	addi $t1, $t1, 4
	j for

exit:
	li $v0, 10
	syscall
```

14) 134 - 2

```
.data
v: .word 73, 12, 5
n: .word 3
z: .word 5
p: .word 3
r: .word 1

.text
j main

exact_p_b8:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addi $fp, $sp, 4
	subu $sp, $sp, 4
	sw $s0, 0($sp)
	subu $sp, $sp, 4
	sw $s1, 0($sp)
	subu $sp, $sp, 4
	sw $s2, 0($sp)

	lw $s0, 0($fp) #v[i]
	lw $s1, 4($fp) #p
	lw $s2, 8($fp) #r
```

```
        li $t1, 0 #nr de r

        while:
                beqz $s0, exit2
                rem $t2, $s0, 8
                seq $t3, $t2, $s2
                add $t1, $t1, $t3
                div $s0, $s0, 8
                j while

        exit2:
                seq $v0, $t1, $s1

                lw $s2, -16($fp)
                lw $s1, -12($fp)
                lw $s0, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 16
                jr $ra

modif:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $ra, 0($sp)
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)
        subu $sp, $sp, 4
        sw $s4, 0($sp)

        lw $s0, 0($fp) #adr v
        lw $s1, 4($fp) #n
        lw $s2, 8($fp) #z
        lw $s3, 12($fp) #p
        lw $s4, 16($fp) #r
```

```
beqz $s1, final
lw $t0, 0($s0) #v[i]

subu $sp, $sp, 4
sw $s4, 0($sp)
subu $sp, $sp, 4
sw $s3, 0($sp)
subu $sp, $sp, 4
sw $t0, 0($sp)
jal exact_p_b8
addu $sp, $sp, 12

beqz $v0, cont
mul $t0, $t0, 3 #v[i]*3
mul $t1, $s2, 2 #z*2
sub $t0, $t0, $t1
sw $t0, 0($s0)
j cont

cont:
        subu $sp, $sp, 4
        sw $s4, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subi $s1, $s1, 1
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        add $s0, $s0, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        jal modif
        addu $sp, $sp, 20

final:
        lw $s4, -28($sp)
        lw $s3, -24($fp)
        lw $s2, -20($fp)
        lw $s1, -16($fp)
        lw $s0, -12($fp)
        lw $ra, -8($fp)
        lw $fp, -4($fp)
```

```
        addu $sp, $sp, 28
        jr $ra

main:
lw $t0, r
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, p
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, z
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
jal modif
addu $sp, $sp, 16

lw $t0, n
mul $t0, $t0, 4 #n*4
li $t1, 0 #contor

for:
        beq $t1, $t0, exit
        lw $a0, v($t1)
        li $v0, 1
        syscall
        la $a0, ' '
        li $v0, 11
        syscall
        addi $t1, $t1, 4
        j for

exit:
        li $v0, 10
        syscall
```

15) 144 - 1

```
.data
```

```
v: .word 123, 231, 4556, 26, 3
n: .word 5
x: .word 10
t: .word 100

.text
j main

prima_cifra_para:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addu $fp, $sp, 4
	subu $sp, $sp, 4
	sw $s0, 0($sp)

	lw $s0, 0($fp) #v[i] + t

	while:
		beqz $s0, exit
		rem $t1, $s0, 10 #ult cif
		div $s0, $s0, 10
		j while

	exit:
		rem $t1, $t1, 2
		seq $v1, $t1, 0

		lw $s0, -8($fp)
		lw $fp, -4($fp)
		addu $sp, $sp, 8
		jr $ra

eval:
	subu $sp, $sp, 4
	sw $fp, 0($sp)
	addu $fp, $sp, 4
	subu $sp, $sp, 4
	sw $ra, 0($sp)
	subu $sp, $sp, 4
	sw $s0, 0($sp)
	subu $sp, $sp, 4
	sw $s1, 0($sp)
	subu $sp, $sp, 4
```

```
        sw $s2, 0($sp)
        subu $sp, $sp, 4
        sw $s3, 0($sp)

        lw $s0, 0($fp) #adr v
        lw $s1, 4($fp) #n
        lw $s2, 8($fp) #x
        lw $s3, 12($fp) #t

        for:
                beqz $s1, final
                lw $t0, 0($s0) #v[i]
                add $t0, $t0, $s3 #v[i] + t

                subu $sp, $sp, 4
                sw $t0, 0($sp)
                jal prima_cifra_para
                addu $sp, $sp, 4

                mul $v1, $v1, -1 #-v1
                addi $v1, $v1, 1 #1 - v1
                sub $t0, $t0, $s3
                sub $t0, $t0, $s2 #v[i] - x
                mul $v1, $v1, $t0 #(1-v1) * (v[i]-x)
                add $v0, $v0, $v1 #adaug la suma

                addi $s0, $s0, 4
                subi $s1, $s1, 1
                j for

        final:
                lw $s3, -24($fp)
                lw $s2, -20($fp)
                lw $s1, -16($fp)
                lw $s0, -12($fp)
                lw $ra, -8($fp)
                lw $fp, -4($fp)
                addu $sp, $sp, 24
                jr $ra

main:
lw $t0, t
subu $sp, $sp, 4
```

```
sw $t0, 0($sp)
lw $t0, x
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
li $v0, 0
jal eval
addu $sp, $sp, 16
move $a0, $v0
li $v0, 1
syscall
li $v0, 10
syscall
```

16) 144 - 2

```
.data
v: .word 231, 3442, 18111
n: .word 3
y: .word 10

.text
j main

cifra_mijloc_impara:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)

        lw $s0, 0($fp) #v[i]-y
        move $t2, $s0 #copie v[i]-y
        li $t3, 0 #nrcifre

        while1:
                beqz $t2, exit1
                addi $t3, $t3, 1
                div $t2, $t2, 10
```

```
            j while1

        exit1:
            rem $t4, $t3, 2
            div $t5, $t3, 2 #nrcifre /2
            beqz $t4, par

            addi $t5, $t5, 1
            while2:
                beqz $t5, exit2
                rem $t6, $s0, 10
                div $s0, $s0, 10
                subi $t5, $t5, 1
                j while2


            par:
                while3:
                    beqz $t5, cont1
                    rem $t6, $s0, 10
                    div $s0, $s0, 10
                    subi $t5, $t5, 1
                    j while2

                cont1:
                    rem $t7, $s0, 10
                    add $t6, $t6, $t7
                    j exit2

        exit2:
            rem $t6, $t6, 2
            seq $v0, $t6, 1

            lw $s0, -8($fp)
            lw $fp, -4($fp)
            addu $sp, $sp, 8
            jr $ra

modif:
        subu $sp, $sp, 4
        sw $fp, 0($sp)
        addi $fp, $sp, 4
        subu $sp, $sp, 4
```

```
sw $ra, 0($sp)
subu $sp, $sp, 4
sw $s0, 0($sp)
subu $sp, $sp, 4
sw $s1, 0($sp)
subu $sp, $sp, 4
sw $s2, 0($sp)

lw $s0, 0($fp) #adr v
lw $s1, 4($fp) #n
lw $s2, 8($fp) #y

beqz $s1, final
lw $t0, 0($s0) #v[i]
sub $t1, $t0, $s2 #v[i]-y

subu $sp, $sp, 4
sw $t1, 0($sp)
jal cifra_mijloc_impara
addu $sp, $sp, 4

beqz $v0, cont
mul $t0, $t0, 2
sw $t0, 0($s0)
j cont

cont:
        subu $sp, $sp, 4
        sw $s2, 0($sp)
        subi $s1, $s1, 1
        subu $sp, $sp, 4
        sw $s1, 0($sp)
        add $s0, $s0, 4
        subu $sp, $sp, 4
        sw $s0, 0($sp)
        jal modif
        addu $sp, $sp, 12

final:
        lw $s2, -20($fp)
        lw $s1, -16($fp)
        lw $s0, -12($fp)
        lw $ra, -8($fp)
```

```
        lw $fp, -4($fp)
        addu $sp, $sp, 20
        jr $ra

main:
lw $t0, y
subu $sp, $sp, 4
sw $t0, 0($sp)
lw $t0, n
subu $sp, $sp, 4
sw $t0, 0($sp)
la $t0, v
subu $sp, $sp, 4
sw $t0, 0($sp)
jal modif
addu $sp, $sp, 12

lw $t0, n
mul $t0, $t0, 4 #n*4
li $t1, 0 #contor

for:
        beq $t1, $t0, exit
        lw $a0, v($t1)
        li $v0, 1
        syscall
        la $a0, ' '
        li $v0, 11
        syscall
        addi $t1, $t1, 4
        j for

exit:
        li $v0, 10
        syscall
```