**Sa se implementeze procedura cifre\_pare(x), cu x numar natural nenul, care afiseaza pe ecran cifrele pare ale numarului x (in orice ordine).**

.data

space: .byte ' '

.text

cifre\_pare:

subu $sp, 4 #salvarea lui fp

sw $fp, 0($sp) # $sp:($fp v)(x)

addi $fp, $sp, 4 # $sp:($fpv)$fp:(x)

subu $sp, 4 # $sp:()($fpv)$fp:(x)

sw $s0, 0($sp) # $sp:($s0)($fpv)$fp:(x)

subu $sp, 4 # $sp:()($s0)($fpv)$fp:(x)

sw $s1, 0($sp) # $sp:($s1)($s0)($fpv)$fp:(x)

subu $sp, 4 # $sp:()($s1)($s0)($fpv)$fp:(x)

sw $s2, 0($sp) # $sp:($s2)($s1)($s0)($fpv)$fp:(x)

lw $s0, 0($fp) # s0 = x

loop\_pare:

ble $s0, 0, exit\_suma\_pare # while suma\_pare>0

rem $s1, $s0, 10 # s1 = s0 % 10

div $s0, $s0, 10 # s0 = s0 / 10

rem $s2, $s1, 2 # s2 = s1 % 2 (verif par/impar)

beq $s2, 0, afiseaza # daca e par, afiseaza

j loop\_pare

exit\_suma\_pare:

sw $s2, -16($fp) #restaurare s2, s1, s0, fp

sw $s1, -12($fp)

sw $s0, -8($fp)

sw $fp, -4($fp)

addu $sp, 16 # elibereaza stiva

jr $ra

afiseaza:

move $a0, $s1

li $v0, 1

syscall

lb $a0, space

li $v0, 11

syscall

j loop\_pare

main:

li $v0, 5 # read int

syscall

subu $sp, 4 #aloc spatiu in stiva

sw $v0, 0($sp) #salvez x pe stiva

jal cifre\_pare

addu $sp, 4 #golim stiva

li $v0, 10

syscall