.data

n:.word 4

v:.word 5, 4, 7, 6

space:.asciiz " "

.text

modifica:

subu $sp,4

sw $fp, 0($sp)

addi $fp,$sp,4

subu $sp,4

sw $ra,0($sp)

subu $sp,4

sw $s0,0($sp)

subu $sp,4

sw $s1,0($sp)

lw $s0, 0($fp) #v

lw $s1, 4($fp) #n

beqz $s1,exit\_modifica

lw $t0,0($s0)

subu $sp,4

sw $t0,0($sp) #vom incarca treptat fiecare valoare a vectorului pe stiva pentru a putea apela g

li $v0,0

#aici se va apela suma\_patrate

jal suma\_patrate

addu $sp,4

#vom inlocui valoarea fiecarui element din vector cu valoarea returnata de procedura suma\_patrate prin registrul $v0

sw $v0,0($s0)

addi $s0,4

addi $s1,-1

#apel recursiv

subu $sp,4

sw $s1,0($sp)

subu $sp,4

sw $s0,0($sp)

jal modifica

addu $sp, 8

exit\_modifica:

lw $s1,-16($fp)

lw $s0,-12($fp)

lw $ra,-8($fp)

lw $fp, -4($fp)

addu $sp,16

jr $ra

suma\_patrate:

subu $sp, 4

sw $fp, 0($sp)

addi $fp, $sp, 4

subu $sp, 4

sw $ra, 0($sp) # pentru ca fac apeluri imbricate catre proc

subu $sp, 4

sw $s0, 0($sp)

# $sp:($s0 v)($ra v)($fp v)$fp:(x)

lw $s0, 0($fp)

beqz $s0, exit # daca x = 0, atunci oprim procedura

addi $s0, -1

mul $t0,$s0,$s0

add $v0,$v0,$t0

subu $sp, 4

sw $s0, 0($sp)

jal suma\_patrate

addu $sp, 4

exit:

lw $s0, -12($fp)

lw $ra, -8($fp)

lw $fp, -4($fp)

addu $sp, 12

jr $ra

main:

lw $t0,n

subu $sp,4

sw $t0,0($sp)

la $t0,v

subu $sp,4

sw $t0,0($sp)

jal modifica

#afisez vectorul v modificat

lw $t0,n

li $t1,0

li $t2,1

for2:

bgt $t2,$t0,exit2

lw $a0,v($t1)

li $v0,1

syscall

la $a0,space

li $v0,4

syscall

addi $t1,4

addi $t2,1

j for2

exit2:

li $v0,10

syscall