**#Lab5: 1**

.data

sir: .asciiz "anaaremere"

revsir: .space 100

space: .asciiz "\n"

.text

main:

adaugareInStiva:

lb $a0,sir($t2)

subu $sp, $sp, 4

sw $a0, 0($sp)

addiu $t2,$t2,1

bnez $a0,adaugareInStiva

addu $sp, $sp, 4

subu $t0,$t2,1

li $t2 0

extragereDinStiva:

lw $a0, 0($sp) #load in registru a valorii de la adresa din varful stivei

sb $a0,revsir($t2)

addu $sp, $sp, 4

addu $t2,$t2,1

blt $t2,$t0,extragereDinStiva

li $a0,0

sb $a0,revsir($t2)#caracterul null

la $a0,revsir

li $v0,4

syscall

li $v0,10 syscall

**#Lab5: 2**

.data

numar: .word 432234

space: .asciiz "\n"

nu: .asciiz "NU"

da: .asciiz "DA"

.text

main:

lw $t0 , numar

move $t3,$t0

li $t1,10

li $t2,0

loop:

add $t2,$t2,1

div $t0,$t1

mflo $t0

mfhi $a0

jal adaugareInStiva

bnez $t0,loop

numberRev:

jal preluareDinStiva

#a0->cifra din stiva

#t3->numaru

div $t3,$t1

mflo $t3

mfhi $t4

#t4->cifra acc

bne $a0,$t4,nonPal

add $t2,$t2,-1

bnez $t2,numberRev

j Pal

#apelaeaza PAL

li $v0,10

syscall

adaugareInStiva:

subu $sp, $sp, 4

sw $a0 ,0($sp)

jr $ra

preluareDinStiva:

lw $a0 ,0($sp)

addu $sp, $sp, 4

jr $ra

nonPal:

la $a0,nu

li $v0,4

syscall

li$v0,10

syscall

Pal:

la $a0,da

li $v0,4

syscall

li$v0,10

syscall

**#Lab5: 3**

.data

.text

main:

li $a0 ,1

li $a1 ,2

li $t0 ,5

loop:

jal proc

add $t0 ,$t0 ,-1

bnez $t0,loop

li$v0,10

syscall

proc:

# $a0 -> n-2

# $a1 -> n-1

# $a2 -> n

mulo $a0,$a0,3

add $a2,$a0,$a1

move $a0,$a1

move $a1,$a2

jr $ra

**#Lab5: 4**

.data

sir: .asciiz "azscsffq"

.text

# $s0 ->contorMAX

# $s1 ->lieraMAX

# $s2 ->contorACC

main:

li $s2,1

#citire pr char

lb $s3,sir($t2)

addiu $t2,$t2,1

mainLoop:

lb $s4,sir($t2)

beq $s4,$s3,incrC

bne $s4,$s3,notEq

continuareLoop:

addiu $t2,$t2,1

move $s3,$s4

bnez $s3,mainLoop

li $v0,10

syscall

incrC:

add $s2,$s2,1

j continuareLoop

notEq:

blt $s2,$s0,notAct

move $s0,$s2

notAct:

li $s2,1

j continuareLoop

**#Lab5: 5**

.data

das: .asciiz "da"

nus: .asciiz "nu"

.text

main:

li $t0,10

li $v0,5

syscall

move $s0,$v0

li $v0,5

syscall

move $s1,$v0

loop:

div $s0,$t0

mflo $s0

mfhi $t1

beq $t1,$s1,da

bnez $s0,loop

j nu

nu:

la $a0,nus

li $v0,4

syscall

li $v0,10

syscall

da:

la $a0,das

li $v0,4

syscall

li $v0,10

syscall

**#Lab5: 6**

.data

n: .word 3

mat: .word 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 , 9

das: .asciiz "da"

nus: .asciiz "nu"

.text

main:

li $t2,2

lw $s0,n

loop:

mulo $t1,$t0,$s0

add $t1,$t1,$t0

mulo $t1,$t1,4

lw $s1,mat($t1)

div $s1,$t2

mfhi $t3

beqz $t3,nu

add $t0,$t0,1

blt $t0,$s0,loop

j da

da:

la $a0,das

li $v0,4

syscall

li $v0,10

syscall

nu:

la $a0,nus

li $v0,4

syscall li $v0,10 syscall

**#Lab5: 7**

.data

n: .word 5

vector: .word 5 ,1 ,3 ,2 ,7

.text

#Legenda:

#s0 ->nr comp

#t0 ->i

#t1 ->j

#a0 -> vetor+i

#a1 -> vetor+j

#t1 ->s1-s2

main:

lw $s0,n #incarca in registrul s0 numarul de elemente

add $s1 ,$s0 , -1 #incarca in registrul s0 numarul de elemente-1

loop1:

mulo $t2,$t0,4 #se obtine adresa in memorie a urmatoarei valori

lw $a0,vector($t2) #incarca in a0 elementul pe care se afla pointerul din vector

add $t1,$t0,1

loop2:

mulo $t3,$t1,4 #simuleaza j-ul, care merge de la i+1 la n

lw $a1,vector($t3)

bgt $a1,$a0, notI #daca a1 < a0 se interschimba

#interschimbare in mem

sw $a1,vector($t2)

sw $a0,vector($t3)

#interschimbare in reg

lw $a0,vector($t2)

lw $a1,vector($t3)

notI:

add $t1,$t1,1 #j++

blt $t1,$s0,loop2 #while j<n-1

add $t0,$t0,1 #i++

blt $t0,$s1,loop1 #$s0-1 #$s0-1, n-1 adica t0 < n-1 , t0 =i

li $v0,10

syscall

**#Lab6:1(c)**

.data

string1: .asciiz "Dati numarul de elemente al matricei.\n"

string3: .asciiz "Dati elementele ,matricei.\n"

string2: .asciiz "matricea este:\n"

newline: .asciiz "\n"

n: .word 0

l1: .word 1

l2: .word 3

elem: .space 64

.text

main:

li $v0, 4

la $a0,string1

syscall

li $v0, 5

syscall

sw $v0, n

lw $t0,n #memorare nr de elem n in reg t0

lw $t3,n

lw $t4,n

mulo $t0,$t0,$t0

li $t1,0

la $t2, elem #initializare t2 cu adresa la care se memoreaza primul element

li $v0, 4 #afisare mesaj de introdus elemente

la $a0,string3

syscall

#introducere elemente

loop:

beq $t0,$t1,end\_loop

addi $t1,1

li $v0, 5 #in bucla se citeste fiecare element

syscall

sw $v0, ($t2) #se pune elementul la urmatoarea adresa in zona de date;

addi $t2,4 # adresarea se face din 4 in 4

b loop

end\_loop:

# citire l1

li $v0, 5

syscall

sw $v0, l1

lw $t8,l1

# citire l2

li $v0, 5

syscall

sw $v0, l2

lw $t9,l2

li $t7,0

add $t8,$t8,-1

mulo $t7,$t8,$t4 #ajung la pr elem dp linia1

mulo $t7,$t7,4

la $s1,elem #ma pozitionez la adresa primului element

add $s1,$t7,$s1

li $t7,0

add $t9,$t9,-1

mulo $t7,$t9,$t4

mulo $t7,$t7,4

la $s2,elem

add $s2,$t7,$s2

li $t1,0

parcurgere:

lw $t5,0($s1)

lw $t6, 0($s2)

sw $t5,0($s2)

sw $t6,0($s1)

addi $s1,4

addi $s2,4

addi $t1,1

beq $t1,$t4,iesire

b parcurgere

iesire:

#de aici incep afisarea

li $v0, 4

la $a0,string2

syscall

li $t1,0

la $t2, elem

loop\_afisare: #afisarea valorilor

beq $t0,$t1,end\_loop\_afisare

beq $t4,$t1 new\_line

lw $a0,($t2) #afisarea valorii de la

adresa memorata in t2

li $v0,1

syscall

addi $t2,4

addi $t1,1

b loop\_afisare

new\_line:

add $t4,$t4,$t3

li $v0, 4

la $a0, newline

syscall

b loop\_afisare

end\_loop\_afisare:

li $v0,10 #oprire executie program syscall

**#Lab6:3(c)**

.data

string1: .asciiz "Dati numarul de elemente al sirului .\n"

string3: .asciiz "Dati sirul.\n"

string2: .asciiz "sirul este:\n"

string4: .asciiz "\n"

n: .word 0

elem: .space 64

.text

proc:

li $t1,0

li $t3,'a'

li $t4,'A'

la $t2,elem

subprogram:

beq $t1,$t0,sfarsit

addi $t1,1

lb $t6,($t2)

beq $t6,$t3,mic

beq $t6,$t4 ,mare

addi $t2,1

j subprogram

mic:

sb $t4,($t2)

addi $t2,1

j subprogram

mare:

sb $t3,($t2)

addi $t2,1

j subprogram

sfarsit:

jr $ra

main:

li $v0, 4

la $a0,string1

syscall

li $v0, 5

syscall

sw $v0, n

lw $t0,n

li $t1,0

la $t2, elem

li $v0, 4

la $a0,string3

syscall

loop:

beq $t0,$t1,end\_loop

addi $t1,1

li $v0, 12

syscall

sb $v0, ($t2)

addi$t2,1

b loop

end\_loop:

li $v0, 4

la $a0,string4

syscall

li $t1,0

jal proc

li $v0, 4

la $a0,string2

syscall

li $t1,0

la $t2, elem

loop\_afisare:

beq $t0,$t1,end\_loop\_afisare

addi $t1,1

lb $a0,($t2)

li $v0,11

syscall

addi $t2,1

b loop\_afisare

end\_loop\_afisare:

li $v0,10

syscall

**#Lab6:2 (c)**

.data

string1: .asciiz "Dati numarul de elemente al vectorului.\n"

string3: .asciiz "Dati elementele ,vectorului.\n"

string2: .asciiz "vectorul este este:\n"

newline: .asciiz "\n"

n: .word 0

p: .word 0

elem: .space 64

.text

putere:

li $t8,1

move $t6,$t4

calcul:

beq $t8,$t9,gata

mulo $t4,$t4,$t6

addi $t8,1

b calcul

gata:

add $t5,$t5,$t4

jr $ra

main:

li $v0, 4

la $a0,string1

syscall

li $v0, 5

syscall

sw $v0, n

lw $t0,n

li $v0, 5

syscall

sw $v0, p #p este puterea

lw $t9,p

li $t1,0

la $t2, elem

li $v0, 4

la $a0,string3

syscall

loop:

beq $t0,$t1,end\_loop

addi $t1,1

li $v0, 5

syscall

sw $v0, ($t2)

addi$t2,4

b loop

end\_loop:

li $t1,0

la $t2,elem

parcurgere:

beq $t1,$t0, sfarsit

lw $t4,($t2)

jal putere

addi $t2,4

addi $t1,1

b parcurgere

sfarsit:

li $v0,1

move $a0,$t5

syscall

li $v0,10

syscal