Punë Laboratori 6

**Ushtrim i 1:**

Jepet kodi në Assembly si më poshtë:

**# messages.asm .data**

**str: .asciiz "the answer = "**

**.text main:**

**li $v0, 4 # kodi i thirrjes së sistemit për print\_string**

**la $a0, str # venia ne regjister e adreses se stringës syscall # printimi i stringës**

**li $v0, 1 # kodi i thirrjes së sistemit për print\_int li $a0, 5 # numri i plotë për tu printuar syscall # printimi i numrit të plotë li $v0, 10 # kodi i thirrjes së sistemit për dalje syscall # mbarimi i programit**

Bëni ndryshime për të krijuar një program i cili merr një numër nga përdoruesi, shfaq stringën e dhënë në kod dhe më pas printon numrin

**Ushtrim i 2:**

Jepet skedari arrayCount.asm :

**# arrayCount.asm**

**.data**

**arrayA: .word 1, 0, 2, 0, 3 count: .word 999**

**.text main:**

**add $zero, $zero, $zero …………..**

**# kodi për të lexuar vlerën e ndryshores X**

**# kodi për të numëruar shumëfishat e X në vektorin arrayA**

**# kodi për të printuar rezultatin**

**# kodi për të mbaruar programin**

1. Tregoni adresat e kujtesës ku ruhen ***arrayA*** dhe ***count.***
2. Zmadhoni vektorin duke vendosur 8 elemente nga 1 deri te 8.
3. Printoni numrin e elementëve të vektorit të cilët plotëpjestohen me një numër të dhënë nga përdoruesi, *X*, ku *X* është një numër i plotë fuqi e 2-shit.
4. Shtoni rreshta në kodin e pikës C në mënyrë që elementet e vektorit ti japë vetë përdoruesi.

Ushtrimi 1

# messages.asm

.data

str: .asciiz "Vendosni një numër: " # stringa për t'u shfaqur

.text

main:

li $v0, 4 # kodi i thirrjes për print\_string

la $a0, str # adresimi i stringës për t'u printuar

syscall # thirrja për print\_string

li $v0, 5 # kodi i thirrjes për leximin e një numri të plotë

syscall # thirrja për leximin e një numri të plotë

move $s0, $v0 # ruaj numrin e lexuar në $s0

li $v0, 1 # kodi i thirrjes për print\_int

move $a0, $s0 # vendos numrin e lexuar në regjistrin për printim

syscall # thirrja për print\_int

li $v0, 10 # kodi i thirrjes për dalje

syscall # dalje nga programi

Ushtrimi 2

Adresa e arrayA ruhet në $s0 i cili ka vlerën 10010040

Adresa e count ruhet në $s1 i cili ka vlerën 10010088

Vlerat janë në formatin Hexadecimal.

1. Madhësinë e vektorit e rrisim duke inicializuar 8 vlera në vend të 5.

# arrayCount

.data

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

prompt1:.asciiz "Numri për të cilin gjejmë shumëfishat në vektor (fuqia e dytë): "

prompt2:.asciiz "Rezultati: "

count: .word 999

.text

main:

la $s0, arrayA # Ngarko adresën bazë të vargut në $s0

la $s1, count # Ngarko adresën e variablës së numrit në $s1

li $s3, 0 # Inicializo numrin për shumëfishat

li $v0, 4 # Printo kërkesën për hyrjen e përdoruesit

la $a0, prompt1

syscall

li $v0, 5 # Lexo numrin nga hyrja e përdoruesit

syscall

addi $t2, $v0, -1 # Ruaj numrin për të gjetur shumëfishat

li $t1, 0 # Inicializo numrin për iteracionet

LOOP:

slti $t0, $t1, 8 # Kontrollo nëse numri i iteracioneve është më pak se madhësia e vargut

beq $t0, $zero, FUND # Nëse jo, del nga cikli

lw $t3, 0($s0) # Ngarko vlerën nga vargu

beq $t3, $zero, SKIP # Nëse elementi i vargut është zero, shko tek iterimi tjetër

and $t0, $t3, $t2 # Kontrollo nëse elementi i vargut është shumëfish i numrit të dhënë

bne $t0, $zero, SKIP # Nëse nuk është shumëfish, shko tek iterimi tjetër

addi $s3, $s3, 1 # Inkremento numrin për shumëfishat

SKIP:

addi $s0, $s0, 4 # Shko te elementi tjetër në varg

addi $t1, $t1, 1 # Inkremento numrin për përsëritje

j LOOP # Përsërit pjesën e shumëfishave

FUND:

li $v0, 4 # Printo kërkesën për rezultatin

la $a0, prompt2

syscall

sw $s3, 0($s1) # Ruaj numrin e shumëfishave në memorie

li $v0, 1 # Printo numrin e shumëfishave

move $a0, $s3

syscall

li $v0, 10 # Dal nga programi

syscall

# arrayCount

.data

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

prompt: .asciiz "Vendos vleren ne vektor: "

prompt1: .asciiz "Numri për të cilin gjejmë shumëfishat në vektor (fuqia e dytë): "

prompt2: .asciiz "Rezultati: "

count: .word 999

.text

main:

la $s0, arrayA # Ngarko adresën bazë të vektorit në $s0

la $s1, count # Ngarko adresën e variablës së numrit në $s1

li $s3, 0 # Inicializo një numër për shumëfishat

# Përshkrimi për të futur vlerat në varg

li $t1, 0 # Inicializo numrin për iteracionin

INSERT:

slti $t0, $t1, 8 # Kontrollo nëse numri i iteracionit është më pak se madhësia e vargut

beq $t0, $zero, DIL # Nëse jo, dal nga përsëritja

li $v0, 4 # Printo kërkesën për vendosjen e vlerave prej përdoruesit

la $a0, prompt

syscall

li $v0, 5 # Lexo vleren nga përdoruesi

syscall

sw $v0, 0($s0) # Ruaj vleren në varg

addi $s0, $s0, 4 # Shko te elementi tjetër në varg

addi $t1, $t1, 1 # Inkremento numrin për iteracion

j INSERT # Përsërit

# Pas futjes së vlerave, kërko numrin për të gjetur shumëfishat

DIL:

li $v0, 4 # Printo kërkesën për hyrjen e përdoruesit

la $a0, prompt1

syscall

li $v0, 5 # Lexo numrin nga hyrja e përdoruesit

syscall

addi $t2, $v0, -1 # Ruaj numrin për të gjetur shumëfishat

# Përshkrimi për të numëruar shumëfishat e numrit të dhënë në varg

li $t1, 0 # Rifillo numrin për iteracionet

LOOP:

slti $t0, $t1, 8 # Kontrollo nëse numri i përsëritjeve është më pak se madhësia e vargut

beq $t0, $zero, FUND # Nëse jo, dal nga përsëritja

lw $t3, 0($s0) # Ngarko vlerën nga vargu

beq $t3, $zero, SKIP # Nëse elementi i vargut është zero, shko tek iterimi tjetër

and $t0, $t3, $t2 # Kontrollo nëse elementi i vargut është shumëfish i numrit të dhënë

bne $t0, $zero, SKIP # Nëse nuk është shumëfish, shko tek iterimi tjetër

addi $s3, $s3, 1 # Inkremento numrin për shumëfishat

SKIP:

addi $s0, $s0, 4 # Shko te elementi tjetër në varg

addi $t1, $t1, 1 # Inkremento numrin për përsëritje

j LOOP # Përsërit

# Printo numrin e shumëfishave

FUND:

li $v0, 4 # Printo kërkesën për rezultatin

la $a0, prompt2

syscall

sw $s3, 0($s1) # Ruaj numrin e shumëfishave në memorie

li $v0, 1 # Printo numrin e shumëfishave

move $a0, $s3

syscall

li $v0, 10 # Dal nga programi

syscall