.text

.globl main

main:

################### preload 3 empty arrays into memory ####################################

la $t7,arrcB

la $t6,arrcA

la $t5,arrcC

#################### getting inputs of b[i][j] and a[i][j] from user ########################

li $t1,4

li $s0,0

loop1: li $s1,0

loop2: sll $t0,$s0,2

addu $t0,$t0,$s1

sll $t0,$t0,3

addu $t0,$t7,$t0 #store array b in t7

li $v0,4 #ask user to input b

la $a0,ask1

syscall

move $a0,$s0

li $v0,1

syscall

li $v0,4

la $a0,ask2

syscall

move $a0,$s1

li $v0,1

syscall

li $v0,4

la $a0,ask3

syscall

li $v0,7 #load double

syscall

s.d $f0,0($t0)

################## now for a #################################

sll $t0,$s0,2

addu $t0,$t0,$s1

sll $t0,$t0,3

addu $t0,$t6,$t0 #store array a in t6

li $v0,4 #ask user to input b

la $a0,ask4

syscall

move $a0,$s0

li $v0,1

syscall

li $v0,4

la $a0,ask5

syscall

move $a0,$s1

li $v0,1

syscall

li $v0,4

la $a0,ask6

syscall

li $v0,7 #load double

syscall

s.d $f0,0($t0)

addiu $s1,$s1,1

bne $s1,$t1,loop2

addiu $s0,$s0,1

bne $s0,$t1,loop1

################### start computing the multiplication #########################

li $t1,4

li $s0,0

Write an assembly code to multiply two (4x4) matrices A and B and store the result in C (also a matrix obviously). Populate A and B with double precision floating point values. The multiplication should happen in column-major order.

L1: li $s1,0

L2: li $s2,0

sll $t2,$s0,2

addu $t2,$t2,$s1

sll $t2,$t2,3

addu $t2,$t5,$t2 #a0 => s5 I store c into t5

l.d $f4,0($t2)

L3: sll $t0,$s2,2

addu $t0,$t0,$s1

sll $t0,$t0,3

addu $t0,$t7,$t0 #a2 => t7

l.d $f16,0($t0)

sll $t0,$s0,2

addu $t0,$t0,$s2

sll $t0,$t0,3

addu $t0,$t6,$t0 #a1 => t6

l.d $f18,0($t0)

mul.d $f4, $f18, $f16

addiu $s2,$s2,1

bne $s2,$t1,L3

s.d $f4,0($t2)

addiu $s1,$s1,1

bne $s1,$t1,L2

addiu $s0,$s0,1

bne $s0,$t1,L1

######################## use loop for printout the c ##########################

li $t1,4

li $s0,0

outp1: li $s1,0

outp2: sll $t2,$s0,2

addu $t2,$t2,$s1

sll $t2,$t2,3

addu $t2,$t5,$t2 #a0 => s5 I store c into t5

li $v0,4

la $a0,rst1

syscall

move $a0,$s0

li $v0,1

syscall

li $v0,4

la $a0,rst2

syscall

move $a0,$s1

li $v0,1

syscall

li $v0,4

la $a0,rst3

syscall

l.d $f12,0($t2) #load c[i][j]

#mov.s $f12,$f9 #move it for output ready

li $v0,3 #output

syscall

li $v0,4 #change line

la $a0,enterlin

syscall

addiu $s1,$s1,1

bne $s1,$t1,outp2

addiu $s0,$s0,1

bne $s0,$t1,outp1

.data

ask1: .asciiz "input b["

ask2: .asciiz "]["

ask3: .asciiz "]:\n"

ask4: .asciiz "input a["

ask5: .asciiz "]["

ask6: .asciiz "]:\n"

rst1: .asciiz "output c["

rst2: .asciiz "]["

rst3: .asciiz "]"

enterlin: .asciiz "\n"

arrcA: .double 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

arrcB: .double 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

arrcC: .double 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

Write an interactive program that inputs 3 integers (say A,B and C) and does the following: (i) computes their product(A\*B\*C) and displays the result (refer to Table 2 for system call codes) (ii) computes their sum (A+B+C), divides the sum by the first input (SUM / A) and displays the quotient and the remainder. Submit your code and screenshot(s) of your simulation containing the Register values.

.text

.globl main

main:

li $v0,4 #ask user to inout 3 ints

la $a0,ask

syscall

li $v0,5 #load 1st int from user (a)

syscall

move $t0,$v0

li $v0,5 #load 2nd int from user (b)

syscall

move $t1,$v0

li $v0,5 #load 3rd int from user (c)

syscall

move $t2,$v0

mult $t0,$t1 # a \* b

mflo $t7 # store lower part of the product into t7

mult $t2,$t7 # a \* b \* c

mflo $t7 # store lower part of product into t7

mfhi $t6 # store higher part of product into t6

add $t1,$t0,$t1 # a + b in t1

add $t1,$t1,$t2 # a + b + c in t1

div $t1,$t0 # sum / a

mflo $t5 # integer result in t5

mfhi $t4 # reminder in t4

li $v0,4 #print out the product of 3 numbers

la $a0,prinmult

syscall

move $a0,$t7

li $v0,1

syscall

li $v0,4 #change line

la $a0,enterlin

syscall

move $a0,$t6

li $v0,1

syscall

li $v0,4 #change line

la $a0,enterlin

syscall

li $v0,4 #print out the result and reminder of (a+b+c)/c

la $a0,divresul

syscall

move $a0,$t5

li $v0,1

syscall

li $v0,4 #change line

la $a0,enterlin

syscall

move $a0,$t4

li $v0,1

syscall

.data

ask: .asciiz "Enter 3 values for a, b and c \n"

prinmult: .asciiz "the product of a, b, c is: \n"

divresul: .asciiz "the result of (a+b+c)/a and the reminder are: \n"

enterlin: .asciiz "\n"