Estructura de los computadores

Memoria de las practicas 7 a 9

Práctica 7:

Cuestión 8:

Escribe el código que calcula la suma de los elementos de la diagonal principal de una matriz 4x4 de valores enteros introducida por teclado. Muestra la suma por pantalla.

(el código está en las páginas 1 a 4):

```
.data
matriz: .word 0:16

intr_fila: .asciiz "Introduce fila "
intr_columna: .asciiz " columna "
intr_valor: .asciiz " : "
verificar: .asciiz "Has introducido bien los numeros?"
otra: .asciiz "Quieres hacer otra operacion? y/n : "
suma: .asciiz "Suma de la diagonal: "

.text
li $s1,4                                    #condicion
li $t3,121                                  #y en ascii
li $t4,110                                  #n en ascii
la $t5,verificar

main:
        jal reset

escribirFilas:
        sle $t1,$s0,$s1
        beqz $t1,fin_escribirFilas          #comprobar condicion filas
        li $s2,1                            #inicio contador columnas

escribirColumnas:
        sle $t1,$s2,$s1
        beqz $t1,fin_escribirColumnas       #comprobar condicion columnas

        la $a0,intr_fila
        li $v0,4
        syscall                             #print introduce fila

        move $a0,$s0
        li $v0,1
        syscall                             #print numero fila

        la $a0,intr_columna
        li $v0,4
```

```
        syscall                         #print columna

        move $a0,$s2
        li $v0,1
        syscall                         #print numero columna

        la $a0,intr_valor
        li $v0,4
        syscall                         #print dos puntos

        li $v0,5
        syscall                         #leer numero

        sw $v0,0($t2)                   #almacena el numero en direccion t2

        addi $t2,$t2,4                  #aumenta la direccion t2 en 4

        addi $s2,$s2,1                  #aumento contador columnas
        j escribirColumnas

fin_escribirColumnas:

        addi $s0,$s0,1                  #aumento contador filas
        j escribirFilas

fin_escribirFilas:

jal reset

imprimirFilas:
        sle $t1,$s0,$s1
        beqz $t1,fin_imprimirFilas      #comprobar condicion filas
        li $s2,1                        #inicio contador columnas

imprimirColumnas:
        sle $t1,$s2,$s1
        beqz $t1,fin_imprimirColumnas   #comprobar condicion columnas


        lw $a0,0($t2)                   #almacena el numero en direccion t2 en a0

        addi $t2,$t2,4                  #aumenta la direccion t2 en 4

        li $v0,1
        syscall                         #imprime numero

        li $a0,32
        li $v0,11
        syscall                         #imprime espacio

        addi $s2,$s2,1                  #aumento contador columnas
        j imprimirColumnas

fin_imprimirColumnas:

        li $a0,'\n'
        li $v0,11
        syscall                         #imprime salto de linea
```

```
        addi $s0,$s0,1          #aumento contador filas
        j imprimirFilas

fin_imprimirFilas:

jal reset
li $t6,0                        #reinicio valor de t6
correcto:
        beq $t6,$t3,fin_correcto    #si y
        beq $t6,$t4,escribirFilas   #si n

        move $a0,$t5
        li $v0,4
        syscall                 #imprime es correcto?

        li $a0,'\n'
        li $v0,11
        syscall                 #imprime salto de linea

        li $v0,12
        syscall                 #lee caracter

        move $t6,$v0

        li $a0,'\n'
        li $v0,11
        syscall                 #imprime salto de linea

        j correcto

fin_correcto:
jal reset
li $t6,0                        #reinicio t6 (donde se guarda y/n)
li $t7,0                        #inicio suma de la diagonal
diagonal:
        sle $t1,$s0,$s1
        beqz $t1,fin_diagonal   #comprobar condicion diagonal
        lw $a0,0($t2)           #carga numero a a0
        add $t7,$t7,$a0         #suma de cada elemento a t7
        addi $t2,$t2,20         #aumenta la posicion de la matriz en 5
        addi $s0,$s0,1          #incremento contador
        j diagonal
fin_diagonal:
        la $a0,suma
        li $v0,4
        syscall                 #print "suma diagonal:"

        move $a0,$t7            #mover el resultado de t7 a a0
        li $v0,1
        syscall                 #muestra la suma

        li $a0,'\n'
        li $v0,11
        syscall                 #imprime salto de linea

        jal reset
continuar:
        beq $t6,$t3,fin_continuar   #si y
        beq $t6,$t4,fin_main        #si n
```

```
        la $a0,otra
        li $v0,4
        syscall                         #print "quieres hacer otra operacion"

        li $a0,'\n'
        li $v0,11
        syscall                         #imprime salto de linea

        li $v0,12
        syscall                         #lee caracter
        move $t6,$v0

        li $a0,'\n'
        li $v0,11
        syscall                         #imprime salto de linea
        j continuar
fin_continuar:

        j main

fin_main:
        li $v0,10
        syscall                         #fin programa

reset:
li $s0,1                                #reinicio contador filas
la $t2,matriz                           #carga direccion de la matriz en t2
jr $ra
```

(Ejemplo de ejecución en la siguiente página)

Ejemplo de ejecución:

```
Introduce fila 1 columna 1 : 1
Introduce fila 1 columna 2 : 2
Introduce fila 1 columna 3 : 3
Introduce fila 1 columna 4 : 4
Introduce fila 2 columna 1 : 4
Introduce fila 2 columna 2 : 3
Introduce fila 2 columna 3 : 2
Introduce fila 2 columna 4 : 1
Introduce fila 3 columna 1 : 5
Introduce fila 3 columna 2 : 6
Introduce fila 3 columna 3 : 7
Introduce fila 3 columna 4 : 8
Introduce fila 4 columna 1 : 8
Introduce fila 4 columna 2 : 9
Introduce fila 4 columna 3 : 88
Introduce fila 4 columna 4 : 6
1 2 3 4
4 3 2 1
5 6 7 8
8 9 88 6
Has introducido bien los numeros?
n
Introduce fila 1 columna 1 : 1
Introduce fila 1 columna 2 : 2
Introduce fila 1 columna 3 : 3
Introduce fila 1 columna 4 : 3
Introduce fila 2 columna 1 : 3
Introduce fila 2 columna 2 : 3
Introduce fila 2 columna 3 : 3
Introduce fila 2 columna 4 : 4
Introduce fila 3 columna 1 : 5
Introduce fila 3 columna 2 : 6
Introduce fila 3 columna 3 : 456
Introduce fila 3 columna 4 : 45
Introduce fila 4 columna 1 : 3
Introduce fila 4 columna 2 : 2
Introduce fila 4 columna 3 : 1
Introduce fila 4 columna 4 : 0
1 2 3 3
3 3 3 4
5 6 456 45
3 2 1 0
Has introducido bien los numeros?
y
Suma de la diagonal: 460
Quieres hacer otra operacion? y/n :
n

-- program is finished running --
```

Práctica 8:

Cuestión 7:

Completa el siguiente código de partida que pide el radio por teclado y tiene que calcular y mostrar en la consola la longitud de la circunferencia y el área del círculo.

```
.data
demanaPi :.asciiz "Dame el valor de pi..."
pideRadio:.asciiz "Dame el radio... "
long: .asciiz "Longitud de la circunferència = "
super: .asciiz "Área del círculo = "
.text
li $v0,4
la $a0,demanaPi
syscall
li $v0,6
syscall
mov.s $f1, $f0
li $v0,4
la $a0,pideRadio
syscall
li $v0,6
syscall
li $v0,4
la $a0,long
syscall

mul.s $f2,$f0,$f1                    #f2 igual a radio por pi

li $t0,2                             #t0=2
mtc1 $t0,$f4                         #paso t0 a f4
cvt.s.w $f4,$f4                      #convierto el valor en f4 a simple precision

mul.s $f12,$f2,$f4                   #f2 por 2

li $v0,2
syscall                              #lee float en f12

mul.s $f0,$f0,$f0                    #f0 igual a radio por radio
mul.s $f12,$f1,$f0                   #f12 igual a radio^2 por pi

li $a0,'\n'
li $v0, 11
syscall                              #salto de linea

la $a0,super
li $v0,4
syscall                              #imprime cadena super

li $v0,2
syscall                              #imprime float en f12

li $v0,10
syscall                              #fin programa
```

Ejemplo de ejecución:

```
Dame el valor de pi...3.14
Dame el radio... 2.5
Longitud de la circunferència = 15.700001
Área del círculo = 19.625
-- program is finished running --
```

Cuestión 8:

Haz el código que suma los elementos del vector y calcula el valor medio en coma flotante. Muestra el resultado por la consola.

```
.data
 array: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
 long: .word 10
 suma: .word 0
 res: .asciiz "Valor medio: "
 .text
 la $t4,array                    #direccion array

 li $s0,1                        #inicio contador
 li $s1,10                       #condicion
 li $t2,0                        #suma
 for_suma:
        sle $t1,$s0,$s1
        beqz $t1,fin_suma        #comprobar condicion

        lw $t3,0($t4)            #carga numero del array
        add $t2,$t2,$t3          #se lo suma a t2

        addi $t4,$t4,4           #aumenta la direccion en 4

        addi $s0,$s0,1           #incremento contador
        j for_suma

fin_suma:
        mtc1 $s1,$f0             #cantidad
        mtc1 $t2,$f2             #suma
        div.s $f12,$f2,$f0       #suma/cantidad en $f12

        la $a0,res
        li $v0,4
        syscall                  #imprime Valor Medio:

        li $v0,2
        syscall                  #imprime float en f12

        li $v0,10
        syscall                  #fin programa
```

Ejemplo de ejecución:

```
Valor medio: 5.5
-- program is finished running --
```

Práctica 9:

Cuestión 5:

Implementar la función float pow(float x;int n) que calcula la potencia n-ésima de x. Los argumentos y los valores se pasan según convenio: x en $f12, n en $a0. El resultado se devuelve en $f0.

(el código esta en páginas 9 y 10):

```
.data
Xpide: .asciiz "X = "
Npide: .asciiz "n = "
powRes: .asciiz "X^n = "

.text
la $a0, Xpide
li $v0,4
syscall                         #imprime "x="

li $v0,6
syscall                         #pide x como float

la $a0, Npide
li $v0,4
syscall                         #imprime "n="

li $v0,5
syscall                         #pide n como entero

mov.s $f12,$f0                  #pasa x a f12
move $a0,$v0                    #pasa n a a0

jal pow                         #llama a funcion

la $a0,powRes
li $v0,4
syscall                         #imprime "x^n="

mov.s $f12,$f0
li $v0,2
syscall                         #imprime x^n

li $v0,10
syscall                         #fin programa

 pow:

li $t0, 1               #valor inicial del resultado es 1, en caso de que la potencia es 0

mtc1 $t0,$f2                    #pasar t0 a f2

cvt.s.w $f2,$f2                 #conversion a float

move $s0,$a0                    #condicion de finaliacion, contador=n

addi $s1,$s1,0                  #inicio contador

mult:
slt $t1,$s1,$s0
```

```
beqz $t1,fin_mult          #comprobar condicion

mul.s $f2,$f2,$f0          #f2(resultado) * X

addi $s1,$s1,1                  #incremento contador
j mult
fin_mult:
mov.s $f0,$f2                   #mueve el resultado a f0
jr $ra                          #volver a jal pow
```

Ejemplo de ejecución:

```
X = 4
n = 3
X^n = 64.0
-- program is finished running --
```

Cuestión 6:

Implementar la función max que nos devuelve el valor mayor de dos números en coma flotante. Los argumentos se pasan según convenio en $f12 y $f14 y el resultado se devuelve en $f0.

```
.data
Xpide: .asciiz "X = "
Ypide: .asciiz "Y = "
MaxRes: .asciiz "El mayor es "
.text
 la $a0, Xpide
 li $v0,4
 syscall                        #imprime "X="

 li $v0,6
 syscall                        #pide X en float

 mov.s $f12,$f0                 #mueve x a f12

 la $a0, Ypide
 li $v0,4
 syscall                        #imprime "Y="

 li $v0,6
 syscall                        #pide y en float

 mov.s $f14,$f0                 #mueve y a f14

 jal max                        #salto a max

 la $a0,MaxRes
 li $v0,4
 syscall                        #imprime "El mayor es"

 mov.s $f12,$f0
 li $v0,2
 syscall                        #imprime valor en f12

 li $v0,10
 syscall                        #fin programa


  max:
 c.lt.s $f12,$f14               #f12<f14 (si V entonces flag a 1,sino 0)
 bc1t Ymayor                    #si flag a 1 entonces Y mayor
 bc1f Xmayor                    #si flag a 0 entonces X mayor

Ymayor:
mov.s $f0,$f14                  #mueve Y a f0
j fin_max                       #fin funcion

Xmayor:
mov.s $f0,$f12                  #mueve X a f0
j fin_max                       #fin funcion

 fin_max:
 jr $ra                         #vuelve a jal max
```

Ejemplo de ejecución:

```
X = 4
Y = 23
El mayor es 23.0
-- program is finished running --


Reset: reset completed.

X = 23
Y = 5
El mayor es 23.0
-- program is finished running --
```