

Nombre:

Apellido:

DNI:

Legajo:

Arquitectura de Computadoras

Parcial 2 - Fecha 1

15/11/2022

1) ¿Cuál es el CPI del siguiente programa? El mismo se ejecuta con forward, BTB y delay slot DESACTIVADOS (2p)

```
.data
numero: .double 4

.code
l.d f6, numero($0)
add.d f3, f3, f3
s.d f9, numero($0)
halt
```

CPI : _____

Ejercicio Puntos

1

2

3

4

5

6

2) ¿Qué instrucciones son necesarias para simular un PUSH a la pila del registro \$t9? (1p)

3. Siendo XX los 2 últimos dígitos de su DNI (los de más a la derecha). ¿Cuántas veces se predecirá correctamente si el salto se toma o no en el siguiente código, si se ejecuta con la opción BTB habilitada? (1p)

```
daddi $t7, $zero, 6
```

```
daddi $t7, $t7, XX
```

siguiente:

```
daddi $t7, $t7, -1
```

```
bnez $t7, siguiente
```

Cantidad : _____

4. Escribir un programa que lea desde teclado un número en punto flotante y lo compare con el valor almacenado en la celda Valor. Si el valor ingresado por teclado es mayor que el almacenado en Valor, deberá calcular $(X - \text{Valor}) * X$, donde X es el número leído por teclado. Caso contrario, deberá calcular $(\text{Valor} - X) / X$. Por último, deberá imprimir el texto "El resultado es:" junto con el valor calculado (6p)

1.

-l.d f6,numero(\$0)

IF	ID	EX	MEM	WB
----	----	----	-----	----

La primera se ejecuta normalmente y guarda el valor en el registro f6 en la etapa WB

-add.d f3,f3,f3

IF	ID	A0	A1	A2	A3	MEM	WB
----	----	----	----	----	----	-----	----

Esta instrucción ejecuta suma de punto flotante que demora 4 ciclos entonces la etapa EX se extiende en 4 ciclos

-s.d f9,numero(\$0)

IF	ID	EX	MEM	WB
----	----	----	-----	----

Esta instrucción se ejecuta normalmente guardando el dato de f9 en la etapa MEM

-halt

IF	ID	EX	MEM	WB
----	----	----	-----	----

Ejecuta todos los ciclos

-> Ciclos serán 9 e instrucciones serán 4 -> CPI = 9/4

2.

daddi \$sp,\$sp,-8	;sp - 8 para guardar los 8 bytes de \$t9
sd \$t9,0(\$sp)	;guardar lo que tiene \$t9 en \$sp

3.

XX = 98

Primero hace $98+6 = 104$

Predecirá $104 - 2 = 102$ veces con BTB

4.

```
.data
valor: .double 10.00
control: .word 0x10000
data: .word 0x10008
msj: .asciiz 'el resultado es:'

.text
##cargar control y data
ld $s6,control($0)
ld $s7,data($0)

##leer flotante(X)
daddi $t0,$0,8 ;funcion 8(leer numero)
sd $t0,0($s6) ;control = funcion 8
ld f0,0($s7) ;f0 = X

##ver si es mayor o menor a valor
#en f0 tenemos X y en f1 tendremos Valor
ld f1,valor($0) ;f1 = valor
c.lt.d f1,f0 ;si f0 > f1 => fp=1 si no fp=0
bc1t mayor ;si fp = 1 salta

##si X fue menor a valor sigue
#hace (valor-x)/x
sub.d f1,f1,f0 ;resta f1(valor)-f0(x)
div.d f1,f1,f0 ;divide f1(valor-x) / f0(x)
j print ;salta a imprimir

##si fue mayor f0(X) con f1(valor)
#hace (X-valor)*X
mayor: sub.d f1,f0,f1 ;resta f0(x)-f1(valor)
mul.d f1,f1,f0 ;multiplica f1(x-valor) * f0(X)

##imprime
print: daddi $t0,$0,msj
sd $t0,0($s7)
```

```

daddi $t0,$0,4
sd $t0,0($s6)
daddi $t0,$0,3 ;funcion 3 imprimir punto flotante
s.d f1,0($s7) ;data = numero calculado
sd $t0,0($s6) ;control = funcion 3
halt

```

5. ¿Qué valor queda guardado en result si el programa se ejecuta con Delay Slot habilitado? (2p)

```

.data
cadena: .asciiz 'ejemplo'
result: .word 0

.code
daddi $t1, $0, cadena
dadd $t2, $0, $0

```

```

loop: lbu $t3, 0($t1)
      daddi $t2, $t2, 2
      bnez $t3, loop
      daddi $t1, $t1, 1
      sd $t2, result($0)
      halt

```

6. Escribir la subrutina MIN_MAX que recibe la dirección de comienzo de una tabla y la cantidad de elementos, y devuelve el valor máximo y el mínimo de dicha tabla. Usando la subrutina, implementar un programa que obtenga el min y el max de 2 tablas. Por último, imprimir en la pantalla gráfica un punto de color Verde (0,255,0) en la coordenada (minimo_tabla2, maximo_tabla2) y otro de color Azul (0, 0, 255) en la coordenada (minimo_tabla1, maximo_tabla1). Usar la convención para nombrar a los registros (8p)

cadena: "asciiz" | ejemplo
result: "word" | 0

daddi \$t1, \$0, cadena	: t1 = dir cadena
daddi \$t2, \$0, \$0	; t2 = 0
loop: lbv \$t3, 0(\$t1)	; t3 = car actual
daddi \$t2, \$t2, 2	; t2 = t2 + 2
bnez \$t3, loop	; si t3 no es 0 salta
daddi \$t1, \$t1, 1	; t1 = t1 + 1 - sig caracter
sd \$t2, result(\$0)	; result = t2 (16) ← guarda
halt	

$e \Rightarrow \begin{cases} t_2 = 2 \\ f_2 = 10 \end{cases}$
 $i \Rightarrow \begin{cases} t_2 = 4 \\ f_2 = 12 \end{cases}$
 $c \Rightarrow \begin{cases} t_2 = 6 \\ f_2 = 14 \end{cases}$
 $m \Rightarrow \begin{cases} t_2 = 8 \\ f_2 = 16 \end{cases}$

```
.data
tabla1: .word 0,10,2,3,4,5,6,7,8,9
tabla2: .word 2,4,6,2,4,32,4,5
control: .word 0x10000
data: .word 0x10008
verde: .byte 0,255,0,0
azul: .byte 0,0,255,0
```

```
.text
##Max y min de tabla1
daddi $a0,$0,tabela1 ;$a0 = dir tabla1
daddi $a1,$0,10      ;cantida de elementos
jal min_max
dadd $s0,$0,$v0      ;$s0 = minimo(x)
dadd $s1,$0,$v1      ;$s1 = maximo(y)
```

##Max y min de tabla2

```
daddi $a0,$0,tabla2 ;$a0 = dir tabla2
daddi $a1,$0,8      ;cantida de elementos
jal min_max
dadd $s2,$0,$v0     ;$s2 = minimo(x)
dadd $s3,$0,$v1     ;$s3 = maximo(y)
```

##inicializar data y control

```
ld $s6,control($0)
ld $s7,data($0)
```

##imprimir punto del min(x) y max(y) de tabla 2 en color verde

```
lw $t0,verde($0)    ;guardar en $t0 el color
sw $t0,0($s7)        ;guardar en data el color
sb $s2,5($s7)        ;guardar en data+5 la x
sb $s3,4($s7)        ;guardar en data+4 la y
daddi $t0,$0,5       ;$t0 = funcion 5 para imprimir
sd $t0,0($s6)        ;imprime pixel
```

##imprimir punto del min(x) y max(y) de tabla 2 en color azul

```
lw $t0,azul($0)     ;guardar en $t0 el color
sw $t0,0($s7)        ;guardar en data el color
sb $s0,5($s7)        ;guardar en data+5 la x
sb $s1,4($s7)        ;guardar en data+4 la y
daddi $t0,$0,5       ;$t0 = funcion 5 para imprimir
sd $t0,0($s6)        ;imprime pixel
halt
```

##Subrutina

```
min_max: daddi $v0,$0,999 ;$v0 = sera el minimo
daddi $v1,$0,-1          ;$v1 = sera el maximo
loop: ld $t0,0($a0)       ;$t0 = numero act
      daddi $a0,$a0,8      ;siguiente numero
      daddi $a1,$a1,-1     ;-1 cantidad de caracteres
      ##preguntamos if $t0 > $v1 por el maximo
      slt $t2,$v1,$t0     ;si $t0 > $v1 => $t2 = 1 si no $t2 = 0
      #si $t2 = 0 significa que no fue mayor entonces es menor y vamos a actualizar el min
      bnez $t2,max        ;si $t2 = 1 salta
      ##preguntamos if $t0 < $v0 por el minimo
      slt $t2,$t0,$v0     ;si $t0 < $v0 => $t2 = 1 si no $t2 = 0
      bnez $t2,min        ;si $t2 = 0 salta
      j seguir
#actualizar maximo
max: dadd $v1,$0,$t0
```

```
j seguir
#actualizar minimo
min: dadd $v0,$0,$t0
#siguiente caracter
seguir: bnez $a1,loop
jr $ra
```