

Observaciones: Escribir las respuestas con tinta. Cada ejercicio indica su valor en puntos. SE APRUEBA CON 10 PUNTOS. NOTA MAXIMA: 20.

1. Dado el siguiente programa, indicar la cantidad de atascos tipo RAW que se producen, la cantidad de Branch Taken Stalls y el número de CPI, si la ejecución se efectúa sólo con la opción Forwarding habilitada (0,5 pts c/u).

Nro. RAWs: 6 BTS: 2 CPE: 33 / 21

```

.data          LOOP: LD r3, tabla1(r1)
tabla1: word 15, 11, 24    DADDI r3, r3, 1
tabla2: word 0, 0, 0      SD r3, tabla2(r1)
                        DADDI r1, r1, 2
.code          DADDI r2, r2, -1
DADDI r1, r0, 0           BNEZ r2, LOOP
DADDI r2, r0, 3           HALT

```

- 2 Indique la nueva cantidad de atascos RAW, BTS y Branch Misprediction si el programa anterior se ejecuta ahora con la opción Branch Target Buffer (BTB) habilitada (0,5 pts c/u).

No. RAWs: 6 BTS: 2 Branch Misprediction: 2

3. El valor del CPI con BTB habilitado es mayor, igual o menor que en el ejercicio 12. Justifique brevemente (0,5 pts).

EN ESTE PROGRAMA QUE SON POCOS SALIR AL MUNDO DE EL CER

4. Reordenar las instrucciones del lazo del programa del ejercicio 1 de manera que, ejecutándolo con la opción Delay Slot habilitada, el resultado generado por el programa sea igual al original (1 punto).

```

    LOOP: LD R3, TABLE(R1)
        DA001 R3, R3, 1
        SD R3, TABLE(R2)
        DA001 R2, R2, -1
        RSNEZ R2, LOOP
        DA001 R1, R2, 8
    HALT

```

5. Completar las dos instrucciones necesarias para convertir a entero un número de punto flotante ubicado en el registro F9 y dejarlo en el registro r5 (0.5 pts c/u)

CvT. L. d. f₀, f₉
 $\eta_{fC} \approx \rho_s, f_0$

6. Completar las tres instrucciones para imprimir en la pantalla alfanumérica del simulador un número en punto flotante, ya almacenado en el registro R (0,5 pts c/u).

lw \$s6, CONTROL(\$0)
lw \$s7, DATA(\$0)

7. El siguiente programa recorre un arreglo de números en punto flotante (TABLA1) y genera otro arreglo (TABLA2) con los números que están dentro del rango establecido por MIN y MAX. Además, guarda en RES la cantidad de elementos de TABLA2. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente (1 pto. c/instrucción). NOTA (del set de instrucciones): c.lt.d F₄, F₅ ; compara F₄ con F₅, dejando flag FP=1 si F₄ es menor que F₅ (en punto flotante) ; bult OFFSET : salta a la dirección rotulada OFFSET si flag FP=1 (ó true) (en punto flotante)

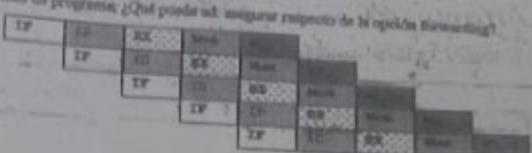
TABLA1:	double	12.0, 15.5, 31.2, 56.4, 44.3, 78.1	
MIN:	double	20.0	L0 Fz, o(%T1)
MAX:	double	50.0	
CANT:	word	6	
RES:	word	0	
TABLA2:	double	0.0	
text			
	LD	\$10, CANT(\$zero) 6	
	LD	F1, MIN(\$zero) 20	
		LD F1, MAX(%Z0R0)	
	DADD	\$10, \$zero, \$zero	
	DADD	\$10, \$zero, \$zero	
LOOP:			
	DADD	\$15, \$zero, \$zero	
	CLTD	F3, F1	
	BC1T	FUERA	
	CLTD	F2, F1	
	BC1T	FUERA	
	DADD	\$10, \$13, 1	
	SD	F3, TABLA2 (\$15)	
FUERA:			
	DADDI	\$M, \$14, 8	
	DADDI	\$10, \$10, -1	
	BNEZ	\$10, LOOP	
	SD	\$13, RES(\$zero)	
	HALT		

8. Dado el siguiente fragmento de programa; ¿Qué puede sd. asegurarse respecto de la opción Branching?

```

LD    R1, 45(R2)      LO
DADO R5, R6, R7      DADO
DCTB 18, R2, 17      DCTB
OR    R3, R1, R7      OR
HALT

```



7/5

- A. Está habilitada
B. No está habilitada.
C. No sé, porque no hay dependencia de salto en el programa
D. NS/NC

9. Marcar con una cruz cuales de las siguientes instrucciones son incorrectas

X	RHEE	IREQ
X	LD	R4, res
X	DADDI	R5, 0, 7
✓	DIV.D	R4, R5, R7

10. ¿En qué dirección se genera la pila en el MIPS?

- A. A partir de la dirección 8000h B. El MIPS no posee pila
C. En la dirección que indica el registro R31 D. NS/NC

9/5

11. Al terminar de ejecutar la siguiente secuencia de instrucciones, el número de CPI será _____

```

LD    R4, dato(R0)
DADDI R1, R14, R6
DADDI R2, R2, 1
SD    R3, dato(R12)
HALT

```

12. En un programa que se ejecuta con la opción Branch Target Buffer habilitada, ¿cuál será el comportamiento al encontrar una instrucción de salto?

- A. Predice que el salto siempre se toma B. Predice si el salto se toma o no dependiendo del código de operación
C. Predice si el salto se toma o no en base a una tabla de historia de saltos D. NS/NC

13. ¿En qué etapa del ciclo del MIPS (IF, EX, MEM, WB) se decide si un salto es tomado o no?

CON B+B ENGENERATL

14. Se cuenta con 1 arreglo llamado TABLA que contiene 15 elementos. Escribir un programa para el winmips64 que genere un nuevo arreglo, a partir de la dirección NUEVO, que contenga sólo aquellos elementos del arreglo TABLA que sean mayores a MAYOR. También se debe establecer la cantidad de elementos de este nuevo arreglo en la dirección CANTIDAD.

.data

TABLA: .word 4,11,18,6,17,28,9,0,11,23,15,6,37,29,14

MENOR: .word 20

MAYOR: .word 10

CANTIDAD: .word 15

TOTAL: .word 0

NUEVO: .word 0

R1 ES SUELTO

NOTICO EN GIT HUB

9/5

Observaciones: Escribir las respuestas con tinta. Cada ejercicio indica su valor en puntos. SE APRUEBA CON 10 PUNTOS. NOTA MAXIMA: 20.

1. Dado el siguiente programa, indicar la cantidad de atascos tipo RAW que se producen, la cantidad de Branch Taken Stalls y el número de CPI, si la ejecución se efectúa sólo con la opción Forwarding habilitada (0,5 pts c/u).

Nro. RAWs: 4 BTS: 1 CPI: 1

```
.data
tabla1: .word 15, 11, 24
tabla2: .word 0, 0, 0

.code
DADDI r1, r0, 0
DADDI r2, r0, 3

LOOP: LD r3, tabla1(r1)
      DADDI r3, r3, 1
      SD r3, tabla2(r1)
      DADDI r1, r1, 8
      DADDI r2, r2, -1
      BNEZ r2, LOOP
      HALT
```

- 2 Indique la nueva cantidad de atascos RAW, BTS y Branch Misprediction si el programa anterior se ejecuta ahora con la opción Branch Target Buffer (BTB) habilitada (0,5 pts c/u).

Nro. RAWs: 0 BTS: 0 Branch Misprediction: 0

3. El valor del CPI con BTB habilitado es mayor, igual o menor que en el ejercicio 1? Justifique brevemente (0,5 pts)

El CPI es menor porque el BTB evita los Branch Target Buffer

4. Reordenar las instrucciones del lazo del programa del ejercicio 1 de manera que, ejecutándolo con la opción Delay Slot habilitada, el resultado generado por el programa sea igual al original (1 pto).

LOOP: _____

 HALT

BAJO EL DADDI M, R1, 8
 ANTES DEL HALT

5. Completar las dos instrucciones necesarias para convertir a entero un número de punto flotante ubicado en el registro f9 y dejarlo en el registro r5 (0,5 pts c/u).

code
 lwu \$s6, CONTROL(\$0)
 lwu \$s7, DATA(\$0)

6. Completar las tres instrucciones para imprimir en la pantalla alfanumérica del simulador un número en punto flotante, ya almacenado en el registro f2 (0,5 pts c/u).

7. El siguiente programa recorre un arreglo de números en punto flotante (TABLA1) y genera otro arreglo (TABLA2) con los números que están dentro del rango establecido por MIN y MAX. Además, guarda en RES la cantidad de elementos de TABLA2. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente (1 pto c/instrucción)
 NOTA (del set de instrucciones): c.lt.d F_a, F_b ; compara F_a con F_b, dejando flag FP=1 si F_a es menor que F_b (en punto flotante)
 belt OFFSET ; salta a la dirección rotulada OFFSET si flag FP=1 (0 true) (en punto flotante)

<pre>.data TABLA1: .double 12.0, 15.5, 31.2, 56.4, 44.3, 78.1 MIN: .double 20.0 MAX: .double 50.0 CANT: .word 6 RES: .word 0 TABLA2: .double 0.0 .text LD \$t0, CANT(\$zero) LD \$f1, MIN(\$zero) DADD \$t3, \$zero, \$zero DADD \$t4, \$zero, \$zero</pre>	<pre>LOOP: DADD \$t5, \$zero, \$zero CLTD F3, F1 BCIT FUERA CLTD F2, F3 BCIT FUERA DADDI \$t0, \$t0, 1 SD F3, TABLA2(\$t5) BNEZ \$t4, LOOP HALT</pre>
	<pre>DADDI \$t0, \$t0, 1 SD F3, TABLA2(\$t5) BNEZ \$t4, LOOP SD \$t0, RES(\$zero)</pre>

TEMA I

8. Escribir un programa para Winmigs que lea 3 números enteros (A, B y C) ingresados por el usuario desde el teclado, resuelva el cálculo $(A - B)^C$ y almacene el resultado en la memoria en la variable RES. El cálculo debe resolverse en una subrutina que reciba como parámetros los 3 operandos y retorne el valor del resultado. Finalmente, el valor calculado debe mostrarse en la pantalla alfanumérica. Debe utilizarse la convención para nombrar los registros que se empleen durante el programa.(10 pts).

```
.data
At: .word 0
B: .word 0
C: .word 0
REST: .word 0
CONTROL: .word32 0x10000
DATA: .word32 0x10008
```

1. Dado el siguiente programa, indicar la cantidad de atascos tipo RAW que se producen, la cantidad de Branch Taken Stalls y el número de CPI, si la ejecución se efectúa sólo con la opción Forwarding habilitada (0,5 pts c/u).

Nro. RAWs: ✓ BTS: ✓ CPI: ✓

```

.data
tabla1: .word 15, 11, 24
tabla2: .word 0, 0, 0

.code
DADDI r1,r0,0
DADDI r2,r0,3

LOOP: LD r3,tabla1(r1)
        DADDI r3,r3,1
        SD r3,tabla2(r1)
        DADDI r1,r1,8
        DADDI r2,r2,-1
        BNEZ r2,LOOP
        HALT

```

2. Indique la nueva cantidad de atascos RAW, BTS y Branch Misprediction si el programa anterior se ejecuta ahora con la opción Branch Target Buffer (BTB) habilitada (0,5 pts c/u). 15

Nro. RAWs: BTS: Branch Misprediction:

3. El valor del CPI con BTB habilitado es mayor, igual o menor que en el ejercicio 1? Justifique brevemente (0,5 pts)

4. Reordenar las instrucciones del lazo del programa del ejercicio 1 de manera que, ejecutándolo con la opción Delay Slot habilitada, el resultado generado por el programa sea igual al original (1 pto).

```
LOOP:    ADD R1, R1, R1      ; R1 = R1 + R1  
        SUB R1, R1, R2      ; R1 = R1 - R2  
        MUL R1, R1, R3      ; R1 = R1 * R3  
        DIV R1, R1, R4      ; R1 = R1 / R4  
        SQR R1, R1           ; R1 = R1^2  
        HALT
```

5. Completar las dos instrucciones necesarias para convertir a punto flotante un valor entero ubicado en el registro r5 y dejarlo en el registro f9 (0,5 pts c/u).

6. Completar las tres instrucciones que faltan para leer un carácter desde el teclado, y que el código ASCII del carácter leído se almacene en el registro \$13 (0.5 pts c/u).

```
.code  
lwu $s6, CONTROL($0)  
lwu $s7, DATA($0)
```

7. El siguiente programa recorre un arreglo de números en punto flotante (TABLA1) y genera otro arreglo (TABLA2) con los números que están dentro del rango establecido por MIN y MAX. Además, guarda en RES la cantidad de elementos de TABLA2. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente (1 pto c/instrucción).

NOTA (del set de instrucciones): **c.lt.d F_d, F_t** ; compara F_d con F_t, dejando flag FP=1 si F_d es menor que F_t (en punto flotante)

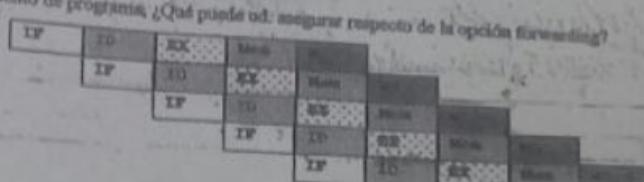
<pre> .data TABLA1: double 12.0, 15.5, 31.2, 56.4, 44.3, 78.1 MIN: double 20.0 MAX: double 50.0 CANT: .word 6 RES: .word 0 TABLA2: double 0.0 .text LD \$10, CANT(\$zero) L.D F1, MIN(\$zero) L.D F2, MAX(\$zero) DADD \$13, \$zero, \$zero </pre>	<pre> LOOP: DADD \$15, \$zero, \$zero LD F3, TABLA1(\$14) CLTD F3, F1 CLTD F2, F3 BCIT FUERA DADDI \$13, \$13, 1 S.D F3, TABLA2(\$15) DADDI \$15, \$15, 8 FUERA: DADDI \$10, \$10, -1 BNEZ \$10, LOOP SD \$13, RES(\$zero) HALT </pre>
---	--

8. Dado el siguiente fragmento de programa, ¿Qué puede oír asegurar respecto de la opción forwarding?

```

LD    R1, 45(R2)
DADD R5, R6, R7
DSUB R8, R2, R7
OR    R9, R1, R7
HALT

```



9/5

- A. Está habilitada
C. No sé, porque no hay dependencia de datos en el programa
B. No está habilitada
D. NS/NC

9. Marcar con una cruz cuáles de las siguientes instrucciones son incorrectas

<input checked="" type="checkbox"/> BNEZ Lazo
<input checked="" type="checkbox"/> LD R4, res
<input checked="" type="checkbox"/> DADDI R5, 0, 7
<input checked="" type="checkbox"/> DIV.D F4, F5, F7

10. ¿En qué dirección se genera la pila en el MIPS?
A. A partir de la dirección 8000h
B. El MIPS no posee pila
C. En la dirección que indica el registro R31
D. NS/NC

11. Al terminar de ejecutar la siguiente secuencia de instrucciones, el número de CPI será ____.
 LD R4, dato(R0)
 DADD R1, R14, R6
 DADDI R2, R2, 1
 SD R5, dato(R12)
 HALT

12. En un programa que se ejecuta con la opción Branch Target Buffer habilitada, ¿cuál será el comportamiento al encontrar una instrucción de salto?

- A. Predice que el salto siempre se toma
B. Predice si el salto se toma o no dependiendo del código de operación
C. Predice si el salto se toma o no en base a una tabla de historia de saltos
D. NS/NC

13. ¿En qué etapa del cauce del MIPS (IF, ID, EX, MEM, WB) se decide si un salto se toma o no? _____

14. Se cuenta con 1 arreglo llamado TABLA que contiene 15 elementos. Escribir un programa para el winmips64 que genere un nuevo arreglo, a partir de la dirección NUEVO, que contenga sólo aquellos elementos del arreglo TABLA que sean mayores a MAYOR. También se debe establecer la cantidad de elementos de este nuevo arreglo en la dirección CANTIDAD.

```
.data
```

```
TABLA: .word 4,11,18,6,17,28,9,0,11,23,15,6,37,29,14
```

```
MENOR: .word 20
```

```
MAYOR: .word 10
```

```
CANTIDAD: .word 15
```

```
TOTAL: .word 0
```

```
NUEVO: .word 0
```

Nro. de Legajo: _____
 NOTA MAXIMA: 20.

1. Dado el siguiente programa, indicar la cantidad de stacos tipo RAW que se producen, la cantidad de Branch Taken Stalls, Cantidad de ciclos que demanda y la cantidad de instrucciones ejecutadas, considerando que la ejecución se efectúa con todas las opciones (Forwarding, BTB, Delay Slot) deshabilitadas. (0.75 pts c/u)

Cantidad de RAWs: 2 Cantidad de BTS: 2

Cantidad de Ciclos: 24 Cantidad de Instrucciones: 11

```
.data .code
cant: .word 2 dadd r1, r0, r0
datos: .word 1, 2 ld r5, cant(r0)
res: .word 0 loops: ld r3, datos(r1)
                                daddi r5, "r5, -1
                                dally r3, r3, r3->Z
                                sd r3, res(r1)-1
                                daddi r1, r1, 1
                                bnez r5, loop
                                halt
```

2. Al finalizar la ejecución del programa previo, ¿qué valor quedará registrado en memoria de datos en la posición 20h ? (1 pto)

3. Dado el siguiente fragmento de ejecución del programa del ejercicio 1 junto a los ciclos resultantes, complete la siguiente frase que puede inferirse: (1 pto)

bnez r5, loop	<table border="1"> <tr> <td>IF</td><td>ID</td><td><u>0</u></td><td>EX</td><td>MEM</td><td>WB</td></tr> </table>	IF	ID	<u>0</u>	EX	MEM	WB
IF	ID	<u>0</u>	EX	MEM	WB		
halt	<table border="1"> <tr> <td>IF</td><td></td><td></td><td></td><td></td><td></td></tr> </table>	IF					
IF							
ld r3, datos(r1)	<table border="1"> <tr> <td>IF</td><td>ID</td><td></td><td>EX</td><td>MEM</td><td>WB</td></tr> </table>	IF	ID		EX	MEM	WB
IF	ID		EX	MEM	WB		

La opción de configuración _____ se encuentra _____

4. ¿Cómo deberían reordenarse las instrucciones del programa del ejercicio 1 de manera que, ejecutándolo con la opción Delay Slot habilitada, la lógica del mismo se mantenga inalterada? (1 pto)

Se debe intercambiar posición entre la instrucción _____ y la instrucción _____

5. Complete los ciclos de la siguiente secuencia de ejecución de instrucciones indicando claramente la etapa del pipeline o el stasco que corresponda. (2 ptos)

add.d f3, f2, f1	<table border="1"> <tr> <td>IF</td><td>ID</td><td><u>0</u></td><td><u>0</u></td><td><u>4</u></td><td>A3</td><td>ME</td><td>WB</td><td><u>0</u></td></tr> </table>	IF	ID	<u>0</u>	<u>0</u>	<u>4</u>	A3	ME	WB	<u>0</u>
IF	ID	<u>0</u>	<u>0</u>	<u>4</u>	A3	ME	WB	<u>0</u>		
mul.d f4, f2, f1	<table border="1"> <tr> <td>IF</td><td><u>0</u></td><td><u>0</u></td><td><u>0</u></td><td><u>0</u></td><td></td><td></td><td></td><td><u>0</u></td></tr> </table>	IF	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>				<u>0</u>
IF	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>				<u>0</u>		
ld r2, 0(r0)	<table border="1"> <tr> <td>IF</td><td><u>0</u></td><td><u>0</u></td><td><u>0</u></td><td><u>0</u></td><td>M0</td><td></td><td></td><td><u>0</u></td></tr> </table>	IF	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	M0			<u>0</u>
IF	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	M0			<u>0</u>		
ld r4, 0(r0)	<table border="1"> <tr> <td>IF</td><td><u>0</u></td><td><u>0</u></td><td><u>0</u></td><td><u>0</u></td><td></td><td></td><td></td><td><u>0</u></td></tr> </table>	IF	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>				<u>0</u>
IF	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>				<u>0</u>		
s.d f3, res1(r0)	<table border="1"> <tr> <td>IF</td><td><u>0</u></td><td><u>0</u></td><td><u>0</u></td><td><u>0</u></td><td></td><td>RA</td><td></td><td><u>0</u></td></tr> </table>	IF	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>		RA		<u>0</u>
IF	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>		RA		<u>0</u>		

6. En la secuencia de instrucciones del ej. 5, además del RAW ¿se produce otro atasco? Indique su tipo y el motivo del mismo (1 pto)

STR

7. Explique brevemente qué realiza el siguiente programa (1 pto)

.data CONTROL: .word32 0x10000 DATA: .word32 0x10008 VALOR: .byte 255,0,0,0,5,10 X Y	.code lwu \$s0, DATA(\$0) lwu \$s1, CONTROL(\$0) ld \$t1, VALOR(\$0) sd \$t1, 0(\$s0) daddi \$t0, \$s0, 5 sd \$t0, 0(\$s1) halt
--	--

Observaciones: Escribir las respuestas con tinta. Cada ejercicio indica su valor en puntos. **SE APRUEBA CON 70 PUNTOS**
MÁXIMA: 20.

1. Dado el siguiente programa, indicar la cantidad de atascos tipo RAW que se producen, la cantidad de Branch Taken Stalls y el número de CPI, si la ejecución se efectúa sólo con la opción Forwarding habilitada (0,3 pts c/u).

Nro. RAWs: _____ BTS: _____ CPI: _____

.data	LOOP: LD r2, tabla1(r1)
tabla1: .word 15, 11, 24	DADDI r5, r5, 1
tabla2: .word 0, 0, 0	SD r5, tabla2(r1)
.code	DADDI r1, r1, 8
DADDI r1, r6, 0	DADDI r2, r2, -1
DADDI r2, r6, 3	BNEZ r2, LOOP
	HALT

2. Indique la nueva cantidad de atascos RAW, BTS y Branch Misprediction si el programa anterior se ejecuta ahora con la opción Branch Target Buffer (BTB) habilitada (0,3 pts c/u).

Nro. RAWs: _____ BTS: _____ Branch Misprediction: _____

3. El valor del CPI con BTB habilitado es mayor, igual o menor que en el ejercicio 1? Justifique brevemente (0,5 pts)

4. Reordenar las instrucciones del lazo del programa del ejercicio 1 de manera que, ejecutándolo con la opción Delay Slot habilitada, el resultado generado por el programa sea igual al original (1 pto).

LOOP: _____

 HALT

H E U f G

5. Completar las dos instrucciones necesarias para convertir a punto flotante un valor entero ubicado en el registro r5 y dejarlo en el registro f9 (0,5 pts c/u).

6. Completar las tres instrucciones que faltan para leer un carácter desde el teclado, y que el código ASCII del carácter leído se almacene en el registro \$t3 (0,5 pts c/u).

.code
 lwu \$s6, CONTROL(\$0) _____
 lwu \$s7, DATA(\$0) _____

7. El siguiente programa recorre un arreglo de números en punto flotante (TABLA1) y genera otro arreglo (TABLA2) con los números que están dentro del rango establecido por MIN y MAX. Además, guarda en RES la cantidad de elementos de TABLA2. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente (1 pto c/instrucción).

NOTA (del set de instrucciones): c.lt.d F_d, F_f; compara F_d con F_f, dejando flag FP=1 si F_d es menor que F_f (en punto flotante)

Belt OFFSET ; salta a la dirección rotulada OFFSET si flag FP=1 (ó true) (en punto flotante)

.data	
TABLA1: double	12.0, 15.5, 31.2, 56.4, 44.3, 78.1
MIN: double	20.0
MAX: double	50.0
CANT: word	6
RES: word	0
TABLA2: double	0.0
.text	
LD	\$t0, CANT(\$zero)
LD	F1, MIN(\$zero)
LD	F2, MAX(\$zero)
DADD	\$t3, \$zero, \$zero

LOOP:	DADD \$t5, \$zero, \$zero
	LD F3, TABLA1(\$t4)
	CLTD F3, F1
	C.LTD F2, F3
	BC1T FUERA
	DADDI \$t3, \$t3, 1
	SD F3, TABLA2(\$t5)
	DADDI \$t5, \$t5, 8
FUERA:	DADDI \$t0, \$t0, -1
	BNEZ \$t0, LOOP
	SD \$t3, RES(\$zero)
	HALT

TEMA 1

8. Escribir un programa para Winmips que lea 3 números enteros (A, B y C) ingresados por el usuario desde el teclado, resuelva el cálculo $(A - B)^C$ y almacene el resultado en la memoria en la variable RES. El cálculo debe resolverse en una subrutina que reciba como parámetros los 3 operandos y retorne el valor del resultado. Finalmente, el valor calculado debe mostrarse en la pantalla alfanumérica. Debe utilizarse la convención para nombrar los registros que se empleen durante el programa (10 pts).

```
.data
A:    .word 0
B:    .word 0
C:    .word 0
RES:   .word 0
CONTROL: .word32 0x10000
DATA:   .word32 0x10008
```

H ECHO

```
.code
```

Arquitectura de Computadoras
Repaso 2012 – Segundo Parcial

Ejercicios 1 a 7 basados en el siguiente programa:

.data	.code
TABLA: .word 1,2,3,4,5 084632	DADDI R14, R0, 5 DADD R15, R0, R0 LOOP: SD R15, TABLA(R15) S 4 3 2 1 DADDI R14, R14, -1 DADDI R15, R15, 8 BNEZ R14, LOOP 4 3 2 1 0 HALT
	i f i d E X M M W R


```

        .code
DADDI R14, R0, 5
DADD R15, R0, R0
LOOP: DADDI R14, R14, -1
      SD R15, TABLA(R15)
      bneq R14, loop
      addi R15, r15, 8
      HALT

```

- A. 7. Con la opción Delay Slot habilitada, ¿cuántas veces se ejecutará la instrucción escrita a continuación de ENEZ R14, LOOP?

8. Dado el siguiente fragmento de programa, ¿Qué puede ud. asegurar respecto de la opción forwarding?

LD R1, 45(R2)	IF	ID	EX	Mem	WB				
DADD R5, R6, R7	IF	ID	EX	Mem	WB				
DSUB R8, R2, R7	IF	ID	EX	Mem	WB				
OR R9, R1, R7	IF	ID	EX	Mem	WB				
HALT	IF	ID	EX	Mem	WB				

- A. Está habilitada
 C. No sé, porque no hay dependencia de datos en el programa

- B. No está habilitada
 D. NS/NC

9. Marcar con una cruz cuales de las siguientes instrucciones son incorrectas

<input checked="" type="checkbox"/>	BNEZ lazo
<input checked="" type="checkbox"/>	LD R4, res
<input checked="" type="checkbox"/>	DADDI R5, 0, 7
<input checked="" type="checkbox"/>	DIV.D F4, F5, F7

10. ¿En qué dirección se genera la pila en el MIPS?

- A. A partir de la dirección 8000h
 B. El MIPS no posee pila
 C. En la dirección que indica el registro R31
 D. NS/NC

$$5+4+0=9$$

11. Al terminar de ejecutar la siguiente secuencia de instrucciones, el número de CPI será _____.

LD R4, dato(R0)	if id ex mm wb
DADD R1, R14, R6	if id ex mm wb
DADDI R2, R2, 1	if id ex mm wb
SD R5, dato(R12)	if id ex mm wb
HALT	if id ex mm wb

cpi = 9/5
 no hay atascos/dependencias

12. En un programa que se ejecuta con la opción Branch Target Buffer habilitada, ¿cuál será el comportamiento al encontrar una instrucción de salto?

- A. Predice que el salto siempre se toma
 B. Predice si el salto se toma o no dependiendo del código de operación
 C. Predice si el salto se toma o no en base a una tabla de historia de saltos
 D. NS/NC

13. ¿En qué etapa del cauce del MIPS (IF, ID, EX, MEM, WB) se decide si un salto se toma o no? _____

14. Se cuenta con 1 arreglos llamado TABLA que contiene 15 elementos. Escribir un programa para el winnips64 que genere un nuevo arreglo, a partir de la dirección NUEVO, que contenga sólo aquellos elementos del arreglo TABLA que sean mayores a MAYOR. También se debe establecer la cantidad de elementos de este nuevo arreglo en la dirección CANTIDAD.

data

TABLA: .word 4,11,18,6,17,28,9,0,11,23,15,6,37,29,14

MENOR: .word 20

MAYOR: .word 10

CANTIDAD: .word 15

TOTAL: .word 0

NUEVO: .word 0



- Dado el siguiente programa, indicar la cantidad de atascos tipo RAW que se introducen, la cantidad de Branch Taken Stalls y el número de CPI, si la ejecución se efectúa con la opción Forwarding habilitada. (0.5 pts c/u)

Nro. RAWs: 3 BTS: 2 CPI: 23 / 14

```

.data
A: .word 3      lazo: daddd r3, r0, r0
B: .word 5      daddd r3, r3, r2
C: .word 0      dadddi r1, r1, -1
                bnez r4, lazo
                ld r1, A(r0)    sd r3, C(r0)
                ld r2, B(r0)   halt
                .code

```

- En el programa anterior ¿qué instrucciones intercambiaría de posición para que el número de atascos RAW sea cero? (1 pto)

sd r3, C(r0)

un lugar antes que lazo

- Si se eliminara el lazo del programa del ejercicio 1 y en lugar de ello se escribieran en forma repetida las instrucciones que correspondan, de manera que el resultado final no se vea alterado, indicar la nueva cantidad de atascos RAW y BTS que se introducirían, manteniendo habilitada la opción Forwarding. (0.5 pts c/u)

Nro. RAWs: 0 BTS: 0

- Reordenar las instrucciones del lazo del programa del ejercicio 1 de manera que, ejecutándolo con la opción Delay Slot habilitada, el valor final en r3 sea el mismo. (1 pto)

```

lazo:    dadd r3, r3, r2
          sd r3, C(r0)
          bnez r1, lazo
          addi r1, r1, -1

```

- Describir brevemente qué funciones cumple la etapa IF del pipeline del Winnips. (1 pto)

- Completar las tres instrucciones que faltan para leer un carácter desde el teclado, y que el código ASCII del carácter leído se almacene en el registro \$t1. (0.5 pts c/u)

```

.code
lwu $s6, CONTROL($0)    addi $t0, $zero, 9
lwu $s7, DATA($0)        sd $t0, 0($s6)
                        ld $t1, 0($s7)

```

- El siguiente programa lee una serie de datos almacenados en la memoria, correspondientes a la temperatura mensual promedio (°C) en la ciudad de La Plata durante 2012, y genera un gráfico de barras en base a dichos datos. Cada °C se representa gráficamente por un punto de color negro. Las barras son verticales y se dibujan una junto a otra, sin espacios blancos entre ellas. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente. (1 pto c/instrucción)

<pre> .data CONTROL: .word32 0x10000 DATA: .word32 0x10008 coordX: .byte 20 coordY: .byte 10 color: .byte 0,0,0 datos: .byte 25,24,21,18,16,12,10,13,15,18,22,24 .text lwu \$s6, CONTROL(\$0) lwu \$s7, DATA(\$0) longitud: addi \$t1,\$0,12 lwu \$s2, color(\$0) sw \$s2, 0(\$s7) </pre>	<pre> lazo1: lb \$s0, coordX(\$0) addi \$t0, \$0, 5 lb \$t3, datos(\$t2) lb \$s1, coordY(\$0) sb \$s0, 5(\$s7) sb \$s1, 4(\$s7) sd \$t0, 0(\$s6) addi \$t3, \$t3,-1 addi \$s1, \$s1,1 bnez \$t3, lazo2 addi \$s0, \$s0, 1 addi \$t2, \$t2,1 bnez \$t1, lazo1 halt </pre>
--	---

TEMA 1

8. Escribir un programa para Winnips que lea un arreglo (tabla1) de N datos en punto flotante almacenados en la memoria, y genere un segundo arreglo (tabla2) cuyos N-1 elementos sean el promedio de cada par de datos consecutivos del arreglo original. Es decir, el primer elemento de tabla2 deberá ser el promedio entre los elementos 1 y 2 de tabla1, el segundo elemento de tabla2 deberá ser el promedio entre los elementos 2 y 3 de tabla1, etc. El cálculo del promedio se debe realizar en una subrutina. El arreglo tabla2 debe quedar almacenado en memoria a continuación de tabla1. Una vez generado el arreglo tabla2, todos sus elementos deben visualizarse en la pantalla terminal del simulador. Esto debe implementarse a través de otra subrutina, que reciba como parámetro la dirección inicial del arreglo. Considerar N=12. (10 pts)

① COMPLETE LOS CICLOS CONSIDERANDO TODAS LAS OPCIONES DE CONFIGURACIÓN DESHABILITADAS (FORWARDING, DELAY SLOT, BRANCH TARGET BUFFER) (CICLOS POSIBLES: IF, ID, EX, MEM, WB, RAW).

LD RA,A (R0)
DADD R2,R4,R3
HALT

IF	ID	EX	MM	WB
IF	RAW	RAW	ID	EX
-	-	IF	ID	EX MM WB

$\frac{9}{3}$

② ¿Cuál es el valor resultante de ciclos por instrucción del programa anterior? CPI =

③ AHORA CONSIDERANDO LA OPCIÓN DE FORWARDING HABILITADA COMPLETE LOS VALORES QUE RESULTARÁN EN LA EJECUCIÓN DE LAS INSTRUCCIONES DEL EJERCICIO 4:

CANT. CICLOS: 8 CANT. RAWSTALL: 1

$\frac{8}{3}$

④ ¿Cuál es el valor de los registros al finalizar el siguiente programa?

.DATA
A: WORD 1
B: WORD 6
.CODE
DADD R1, R0, R0
DADDI R1, R0, 8

LD RA,A (R0)
LD R2,A (R15)
LD R3,B (R0)
DSUB R5,R2,R3
HALT

KJS = 8
R1 = 1
R2 = 6
R3 = 6
RS = 0

⑤ COMPLETE LAS OPCIONES DE CONFIGURACIÓN QUE NO SE PUEDEN HABILITAR EN SIMULTÁNEO:

• LA OPCIÓN DELAY SLOT NO PUEDE HABILITARSE JUNTO CON LA OPCIÓN BRANCH TARGET BUFFER

EJERCICIOS 6 AL 8 BASADOS EN LA EJECUCIÓN DE LAS SIGUIENTES INSTRUCCIONES

.CODE
DADDI R1,R0,1 1
DADDI R2,R0,2 2
DADD R3,R0,R0 0
BEQ R1,R2,LISTO R1=R2 → Listo
BNE R1,R2,FIN R1 > R2 → FIN
J FIN —→ 4,1 MF

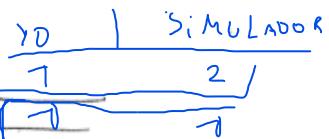
LISTO : DADDI R3,R0,1
FIN : HALT

⑥ COMPLETE LA CANTIDAD DE BRANCH TAKEN STALLS QUE SE COMPUTARÁN EN LA EJECUCIÓN DEL PROGRAMA ANTERIOR SEGÚN LA OPCIÓN DE CONFIGURACIÓN DEL BRANCH TARGET BUFFER:

BTS = 1 PARA N;

EL SIMULADOR DICE 2

ES 1 BTS
Y
1 MIS PREDICTION
SEGUN Y



CON BRANCH TARGET BUFFER HABILITADA SE COMPUTARAN
" " DESHABILITADA " "

- 14) EN BASE A LA SIGUIENTE SECUENCIA DE EJECUCIÓN Y CICLOS DEL PROGRAMA ANTERIOR, ¿QUE OPCIÓN DE CONFIGURACIÓN PUEDE ARMAR QUE SE ENCUENTRA HABILITADA

BED R1, R2, LISTO	IF	ID	EX	MEM	WB
BNE R1, R2, FIN	IF	ID	EX	MEM	WB
J FIN	IF	—	—	—	—
HALT	IF	—	—	—	—

IF ID EX MEM WB

- 15) SI SE DESEA HABILITAR LA OPCIÓN ENABLE DELAY SLOT GARANTIZANDO QUE LA LÓGICA DEL PROGRAMA ANTERIOR SE MANTIENE INALTERADA COMPLETE LA SIGUIENTE TABLA SE DEBE AGREGAR LA INSTRUCCIÓN _____ PREVIO A LA INSTRUCCIÓN _____

- 16) COMPLETAR LA COLUMNA CC CON LA CANTIDAD DE CICLOS DE RELOJ QUE REQUIERE CADA INSTRUCCIÓN PARA SU EJECUCIÓN. LA CONFIGURACIÓN ES LA EXISTENTE POR DEFECTO
- | INSTRUCCIÓN | CC | CICLOS POR |
|-------------|----|----------------|
| LD | | RELOJ POSIBLES |
| MUL.D | | 8 |
| DIV.D | | 5 |
| DADD | | 28 |
| S.D | | 11 |
| ADD.D | | |

- 17) DADA LA ABSENCIA DEL CONCEPTO DE PILA, Y ANTE LA INVOCACIÓN A UNA SUBRUTINA MEDIANTE LA INSTRUCCIÓN JAL (AL DEBERÍA SER LA ÚLTIMA INSTRUCCIÓN A EJECUTARSE EN LA SUBRUTINA PARA GARANTIZAR EL CORRECTO RETORNO?) _____ JR \$RA

- 18) SE CUENTA CON UN ARREGLO LLAMADO TABLA QUE CONTIENE 8 ELEMENTOS. ESCRIBIR UN PROGRAMA QUE GENERE UN NUEVO ARREGLO, A PARTIR DE LA DIRECCIÓN NUEVA QUE CONTENGA SOLO AQUELLOS ELEMENTOS QUE SEAN MAYORES A MAYOR (INCLUYE SUBRUTINA) QUE COMPARE LOS ELEMENTOS DE LA TABLA.

Preguntas de Parcial – Winmips64

1) Ejercicios A hasta G están basados en el siguiente programa:

```

.data
TABLA: .word 1,2,3,4,5

.code
    DADDI R14, R0, 5
    DADDI R15, R0, R0
LOOP: SD    R15, TABLA(R15)      R15 = 0, 8,
    DADDI R14, R14, -1
    DADDI R15, R15, 8
    BNEZ R14, LOOP
    HALT

```

a- ¿Qué tarea realiza el programa anterior?

- 1 – Lee una tabla de 5 elementos, les suma 8 a cada uno y los vuelve a almacenar en memoria.
 - 2 – Almacena los números del 5 al 1 en orden decreciente a partir de la dirección TABLA.
 - 3 – Genera una tabla de 5 elementos con los múltiplos de 8 comenzando desde cero.
 - 4 – NS/NC.

b- ¿Qué valor queda almacenado en la dirección de memoria TABLA+1 una vez finalizada la ejecución?

- 1 – 2 (dos).
 - 2 – 0 (cero).
 - 3 – 8 (ocho).
 - 4 – NS/NC.

c- Solamente con la opción Forwarding habilitada el programa demanda 31 ciclos de reloj para ejecutarse completamente. ¿Cuántos ciclos se están malgastando producto de los diferentes atascos que se producen?

- 1 – 4 ciclos.**
 - 2 – 8 ciclos.**
 - 3 – Ninguno.**
 - 4 – NS/NC.**

d- Con la opción de Forwarding inhabilitada se generan 7 atascos tipo RAW adicionales durante la ejecución del programa. ¿En qué consiste la técnica del Forwarding utilizada para reducir el número de atascos?

- 1 – El Forwarding** reduce la cantidad de atascos estructurales a través de un reordenamiento de las instrucciones.

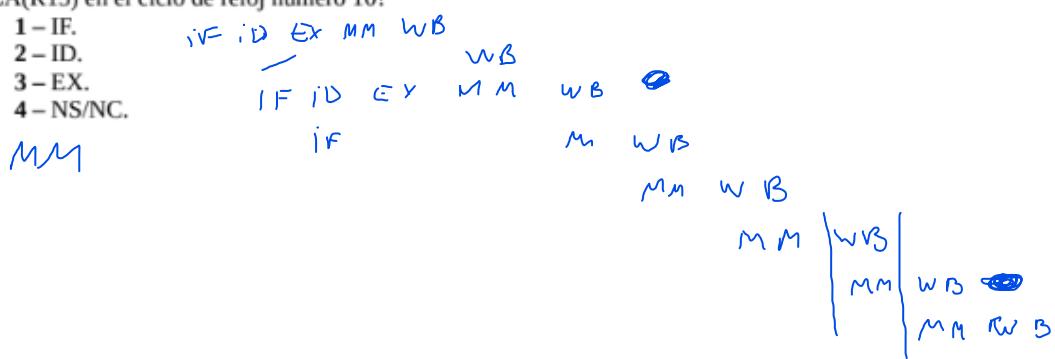
2 – El Forwarding permite utilizar resultados temporales evitando esperar a que sean escritos.

3 – El Forwarding predice si un salto condicional va a ser tomado o no evitando el ingreso al pipeline de la instrucción que no debe ejecutarse.

4 – NS/NC.

e- Asumiendo que no se produce ningún atasco por dependencia de datos al ejecutarse el programa sólo con la opción Forwarding habilitada ¿en qué etapa del pipeline se encontrará la instrucción SD R15, TABLA(R15) en el ciclo de reloj número 10?

- 1 – IF.**
 - 2 – ID.**
 - 3 – EX.**
 - 4 – NS/NC.**



f- ¿Cómo reordenaría el programa para ejecutarlo con la opción Delay Slot habilitada de manera que no se produzcan atascos de ningún tipo?

```

    .code
        DADDI R14, R0, 5      7 2 3 4
        DADD R15, R0, R0
    LOOP: sd r15, tabla(r15) 5 4 3 2 1
          addi r14, r14, -1
          bneq r14, loop
          addi r15, r15, 8      9 3 2 1 & NO SALT
          HALT
  
```

g- Con la opción Delay Slot habilitada, ¿cuántas veces se ejecutará la instrucción escrita a continuación de BNEZ R14, LOOP?

- 1 - 4 (cuatro).
- 2 - 5 (cinco).
- 3 - 1 (uno).
- 4 - NS/NC.

2) Dado el siguiente fragmento de programa. ¿Qué puede usted asegurar respecto de la opción forwarding?

LD R1, 45(R2)	IF	ID	EX	Mem	WB	
DADD R5, R6, R7	IF	ID	EX	Mem	WB	
DSUB R8, R2, R7	IF	ID	EX	Mem	WB	
OR R9, R1, R7	IF	ID	EX	Mem	WB	
HALT	IF	ID	EX	Mem	WB	

- a- Está habilitada.
- b- No está habilitada.
- c- No se, porque no hay dependencia de datos en el programa.
- d- NS/NC.

3) Marcar con una cruz cuales de las siguientes instrucciones son incorrectas.

✗ BNEZ lazo
✗ LD R4, res
✗ DADDI R5, 0, 7
✓ DIV.D F4, F5, F7

4) ¿En qué dirección se genera la pila en el MIPS?

- a- A partir de la dirección 8000h.
- b- El MIPS no posee pila.
- c- En la dirección que indica el registro R31.
- d- NS/NC.

5) Al terminar de ejecutar la siguiente secuencia de instrucciones, el número de CPI será _____. 9/5

```
LD      R4, dato(R0)    if id ex mm wb
DADD   R1, R14, R6     if id ex mm wb
DADDI  R2, R2, 1        if id ex mm wb
SD      R5, dato(R12)   if id ex mm wb
HALT
```

6) En un programa que se ejecuta con la opción Branch Target Buffer habilitada ¿cuál será el comportamiento al encontrar una instrucción de salto?

- a- Predice que el salto siempre se toma.
- b- Predice si el salto se toma o no dependiendo del código de operación.
- c- Predice si el salto se toma o no en base a una tabla de historia de saltos.
- d- NS/NC.

7) ¿En qué etapa del cauce del MIPS (IF, ID, EX, MEM, WB) se decide si un salto se toma o no? ____.

8) Se cuenta con 1 arreglo llamado TABLA que contiene 15 elementos. Escribir un programa para el winmips64 que genere un nuevo arreglo, a partir de la dirección NUEVO, que contenga sólo aquellos elementos del arreglo TABLA que sean mayores a MAYOR. También se debe establecer la cantidad de elementos de éste nuevo arreglo en la dirección CANTIDAD.

```
.data
TABLA: .word 4,11,18,6,17,28,9,0,11,23,15,6,37,29,14
MENOR: .word 20
MAYOR: .word 10
CANTIDAD: .word 15
TOTAL: .word 0
NUEVO: .word 0
```

9) Dado el siguiente programa, indicar la cantidad de atascos tipo RAW que se producen, la cantidad de Branch Taken Stalls y el número de CPI, si la ejecución se efectúa con la opción Forwarding habilitada.

```
.data
A: .word 3
B: .word 5
C: .word 0

.code

ld R1, A(R0)
ld R2, B(R0)
dadd R3, R0, R0
lazo: dadd R3, R3, R2
       addi R1, R1, -1
       bnez R1, lazo
sd R3, C(R0)
HALT.
```

Nro. RAWs: 3 BTS: 2 CPI: 23/14

MFV, 1
BTS
MS
TICK, 1P₀

- 10) En el programa anterior ¿qué instrucciones intercambiaría de posición para que el número de atascos RAW sea cero?

S0 R3 ↪

- 11) Si se eliminara el lazo del programa del ejercicio 9 y en lugar de ello se escribieran en forma repetida las instrucciones que correspondan, de manera que el resultado final no se vea alterado, indicar la nueva cantidad de atascos RAW y BTS que se producirían, manteniendo habilitada la opción Forwarding.

Nro. RAWs: 0 BTS: 0

- 12) Reordenar las instrucciones del lazo del programa del ejercicio 9 de manera que, ejecutándolo con la opción Delay Slot habilitada, el valor final en R3 sea el mismo.

lazo: DADDI \$S, R1, -1
BNE NZ R2, L12
DADD R3, R3, R2

- 13) Describir brevemente qué funciones cumple la etapa IF del pipeline del Winmips.

En la etapa IF se accede a memoria en busca de la instrucción

- 14) Completar las tres instrucciones que faltan para leer un carácter desde el teclado, y que el código ASCII del carácter leído se almacene en el registro \$t1.

```
.code
lwu $s6, CONTROL($0)
lwu $s7, DATA($0)
daddi $t0, $zero, 9
sd $t0, 0($s6)
lbu $t1, 0($s1)
```

- 15) El siguiente programa lee una serie de datos almacenados en la memoria, correspondientes a la temperatura mensual promedio ($^{\circ}$ C) en la ciudad de La Plata durante 2012, y genera un gráfico de barras en base a dichos datos. Cada $^{\circ}$ C se representa gráficamente por un punto de color negro. Las barras son verticales y se dibujan una junto a otra, sin espacios blancos entre ellas. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente.

```
.data
CONTROL:.word32 0x10000
DATA: .word32 0x10008
coordX: .byte 20
coordY: .byte 10
color: .byte 0,0,0,0
datos: .byte 25,24,21,18,16,12,10,13,15,18,22,24

.text
    lwu $s6, CONTROL ($0)
    lwu $s7, DATA ($0)
    addi $t1, $0, 12
    addi $t2, $zero, 0
    lwu $s2, color($0)
    sw $s2, 0($s7)
```

```
lb $s0, coordX($0)
daddi $t0, $0, 5
lazo1: lb $t3, datos($t2)
        lb $s1, coordY($0)
        sb $s0, 5($s7)
lazo2: sb $s1, 4($s7)
        sd $t0, 0($s0)
        daddi $t3, $t3, -1
        daddi $s1, $s1, 1
        bnez $t3, lazo2
        daddi $s0, $a0, 1
        daddi $t2, $t2, 1
        daddi $t1, $t1, -1
        bnez $t1, lazo1
halt
```

- 16) Dado el siguiente fragmento de programa, determinar el contenido de los registros después de su completa ejecución (opción Forwarding habilitada).

```

    .data
    dato: .word 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

    .code
        DADDI R4, R0, 0
        LD R6, dato(R4)
lazo: DADD R4, R4, R6      4      7      9      10
        DSLL R7, R4, 2      16     21     27     40
        DADDI R6, R6, -1      3      2      1
        BNEZ R6, lazo
        DADDI R6, R6, -2
        HALT

```

lo tengo explicado en el pdf myboard

*R4 = 9 1
R6 = 0
R7 = 0*

*16 20 36 40
10, 16, 24, 28*

R4 10; R6 -2; R7 40

- 17) Si el programa anterior se ejecuta con la opción Delay Slot habilitada.

- a- La instrucción DSLL se completa 2 veces.
- b- La instrucción DSLL se completa 3 veces.
- c- La instrucción DSLL se completa 1 vez.
- d- NS/NC.

- 18) Completar la columna CC con la cantidad de ciclos de reloj que requiere cada instrucción para su ejecución.

Instrucción	CC	Ciclos de Reloj
LD	5	5
MUL.D	7	8
DIV.D	28	11
DADD	5	28
S.D	5	
ADD.D	8	

IF ID A0 M1 M2 R3 M4 V6

- 19) Un programa que se ejecuta sin adelantamiento de Operandos o Forwarding genera un número N de RAW. Si ese mismo programa ahora se ejecuta con forwarding:

- a- Se eliminan completamente los RAW.
- b- El número de RAW será igual a N - 1.
- c- No se puede determinar (el número de RAW dependerá del programa).

- 20) Implemente un programa en Winmips64 que recorra el vector VECTOR de 5 elementos numéricos y los almacene a partir de la dirección NUEVO con el valor incrementado en 1.

```

    .data
    VECTOR: .word 1234, 2345, 3456, 4567, 5678
    NUEVO: .word 0

```



- 21) Describir brevemente qué funciones cumple la etapa ID del pipeline del Winmips.

Se decodifica la instrucción, se accede el banco de registros por los operandos, se calcula el valor del operando inmediato. Si es un salto, se calcula el destino y si se toma o no

22) Escribir un programa Winnips que genere un arreglo (tabla1) de N números en punto flotante ingresados por el usuario desde el teclado, y otro arreglo (tabla2) con los elementos de tabla1 elevados al cuadrado. Ambos arreglos deben almacenarse en memoria uno a continuación de otro. El ingreso de datos por el teclado se debe implementar en una subrutina. Una vez ingresados los N valores y almacenados en memoria se debe generar el segundo arreglo. Esto debe implementarse a través de otra subrutina, que reciba como parámetros las direcciones de comienzo de tabla1 y tabla2. Considerar N=12.

23) Complete los ciclos considerando todas las opciones de configuración deshabilitadas (forwarding, delay slot, branch target buffer). (Ciclos posibles: IF, ID, EX, MEM, WB, RAW).

LD R4, A (R0)	IF	id	ex	mm	wb					if id ex mm wb
DADD R2, R4, R3	IF	raw	RAW	id	EX	mm	WB			if id raw ex mm wb
HALT	-	-	if	ID	ex	mm	WB			if - id ex mm wb

24) ¿Cuál es el valor resultante de ciclos por instrucción del programa anterior? CPI: 9/3

25) Ahora considerando la opción de forwarding habilitada complete los valores que resultarán en la ejecución de las instrucciones del ejercicio 23.

Cantidad de ciclos: 8/3 Cantidad Raw Stall: 1

26) ¿Cuál es el valor de los registros al finalizar el siguiente programa?

```

.data
A: .word 1
B: .word 6

.code
DADD R15, R0, R0      r15= 8
DADDI R15, R0, 8        r1= 1
LD   R1, A(R0)          r2= 6
LD   R2, A(R15)         r3= 6
LD   R3, B(R0)          r5= 0
DSUB R5, R2, R3
HALT

```

27) Complete las opciones de configuración que no se pueden habilitar en simultáneo:
La opción delay slot no puede habilitarse junto con la opción branch target buffer

28) Los ejercicios A, B y C están basados en la ejecución de las siguientes instrucciones:

```

.code
DADDI R1, R0, 1
DADDI R2, R0, 2
DADD R3, R0, R0
BEQ R1, R2, LISTO
BNE R1, R2, FIN
J FIN
LISTO: DADDI R3, R0, 1
FIN: HALT

```

- a-** Complete la cantidad de branch taken stalls que se computarán en la ejecución del programa previo, según la opción de configuración del branch target buffer:
- Con branch target buffer habilitada se computarán 1 pero la computadora creo que me tiraba 2 nose porq
 - Con branch target buffer deshabilitada se computarán 1.
- b-** En base a la siguiente secuencia de ejecución y ciclos del programa anterior ¿qué opción de configuración puede afirmar que se encuentra habilitada?

BEQ R1, R2, LISTO	IF	IF	EX	MEM	WB	nose				
BNE R1, R2, FIN		IF	ID			EX	MEM	WB		
J FIN			IF							
HALT						IF	ID	EX	MEM	WB

- c-** Si se desea habilitar la opción Enable Delay Slot garantizando que la lógica del programa anterior se mantiene inalterada, complete la siguiente frase:

Se debe agregar la instrucción _____ previo a la instrucción _____, muy rebuscado

29) Dada la ausencia del concepto de pila, y ante la invocación a una subrutina mediante la instrucción JAL, ¿cuál debería ser la última instrucción a ejecutarse en la subrutina para garantizar el correcto retorno? jr \$ra

30) Los ejercicios A hasta E están basados en las siguientes instrucciones y su detalle de ejecución:

LD R1, A(R0)	IF	ID	EX	Mem	WB	EX	Mem	WB
DADD R2, R1, R3	IF	ID	Raw	Raw		EX	Mem	WB
HALT	IF				ID	EX	Mem	WB

A- ¿A qué se debe la presencia de Raw Stalls?

- 1 – El mismo componente de pipeline no puede atender varios pedidos simultáneos.
- 2 – La segunda instrucción usa registros alterados por la instrucción previa.
- 3 – La etapa Decode de la instrucción DADD utiliza 3 ciclos.
- 4 – NS/NC.

B- ¿Cómo minimizaría el problema del Raw stall mediante la configuración del Winmips?

- 1 – Habilitando Forwarding.
- 2 – Habilitando Branch Target Buffer.
- 3 – Desabilitando Forwarding.
- 4 – NS/NC.

C- ¿Cuántos ciclos de ejecución demandó la ejecución del programa?

- 1 – 9 ciclos.
- 2 – 19 ciclos.
- 3 – 7 ciclos.
- 4 – NS/NC.

D- ¿Cuál sería el estado del Flag Z al finalizar la ejecución DADD R2, R1, R3?

- 1 – Z=0.
- 2 – No se puede determinar por el valor desconocido de R3.
- 3 – Esta arquitectura no utiliza flags de estado.
- 4 – NS/NC.

E- En un escenario ideal sin atascos (stalls) y maximizando el paralelismo del Pipeline, ¿cuántos ciclos hubiera demandado la ejecución del programa previo?

1 – 5 ciclos.

2 – 7 ciclos.

3 – 9 ciclos.

4 – NS/NC.

31) Los ejercicios F al J están basados en la ejecución de las siguientes instrucciones:

```

.data
A: .word 10

.code
LD R1, A(R0)
LOOP: DADDI R1, R1, -1
      DADDI R0, R0, 1
      BNEZ R1, LOOP
      NOP
      HALT
  
```

F- Si se habilita la opción de configuración Delay Slot, el programa demandará:

1 – Igual cantidad de ciclos para completarse.

2 – Mas cantidad de ciclos para completarse.

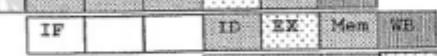
3 – Menos cantidad de ciclos para completarse.

4 – NS/NC.

G- En base al siguiente orden de ejecución ¿cómo se encuentra la opción de configuración Enable Delay Slot?

```

LD    R1, A(R0)
DADDI R1, R1, -1
DADDI R0, R0, 1
BNEZ R1, loop
NOP
DADDI R1, R1, -1
  
```



Sin Forward

y Sin Delay slot

1 – Deshabilitada.

2 – Habilitada.

3 – No se puede determinar.

4 – NS/NC.

H- Si se activa la opción Enable Delay Slot, y se elimina la instrucción NOP del programa, el mismo funcionará correctamente? ¿por qué?

1 – Si, porque la instrucción NOP es redundante en el programa.

2 – No, porque se ejecutará el Halt en la primer iteración del loop abortando el programa.

3 – No, porque el programa se quedaría ejecutando indefinidamente.

4 – NS/NC.

I – Si se ejecutó el programa con Forwarding, Branch Target Buffer (BTB) y Delay Slot deshabilitados los tres, obteniendo un promedio de Ciclos por Instrucción de 1,758 CPI y luego se ejecutó nuevamente el programa pero habilitando BTB obteniendo 1,606 CPI. Indique cuál afirmación es correcta.

1 – BTB mejoró en este caso el tiempo de ejecución del programa por la cantidad de saltos en el Loop.

2 - BTB empeoró en este caso el tiempo de ejecución del programa por la cantidad de saltos en el Loop.

3 - Habilitar Forwarding no mejorará los tiempos de ejecución del programa por no tener dependencias de datos entre instrucciones vecinas.

4 - NS/NC.

J- ¿Cómo cambiaría el programa en la cercanía a la instrucción BNEZ para eliminar el NOP y mantener la misma lógica consistente con opción Delay Slot Habilitado?

- 1- LOOP: DADDI R1, R1, -1
 BNEZ R1, LOOP
 DADDI R0, R0, 1
 HALT
- 2- LOOP: DADDI R0, R0, 1
 BNEZ R1, LOOP
 DADDI R1, R1, -1
 HALT
- 3- LOOP: DADDI R1, R1, -1
 BNEZ R1, LOOP
 HALT
 DADDI R0, R0, 1
- 4- NS/NC

32) Decir cual de las tres opciones es verdadera:

- a- En operaciones aritméticas con enteros sólo el resultado puede estar referenciado como una dirección de memoria.
- b- En operaciones en punto flotante sólo uno de los dos operandos puede estar referenciado como una dirección de memoria.
- c- El acceso a memoria está limitado sólo a dos instrucciones.
- d- NS/NC.

33) Escribir un programa que multiplique dos números enteros almacenados en memoria, mediante sumas sucesivas y almacene el resultado en memoria.

34) Pipeline aprovecha el paralelismo:

- a- A nivel de instrucciones.
- b- A nivel del procesador.
- c- Por tener dos ALUs.
- d- NS/NC.

35) La instrucción DADD R2, R3, R1 se ejecuta sin forwarding. El resultado de la operación está disponible para una instrucción posterior:

- a- Al finalizar la etapa WB.
- b- Al finalizar la etapa EX.
- c- Al finalizar la etapa MEM.
- d- NS/NC.

36) La instrucción DSUB R2, R3, R1 se ejecuta con forwarding. El resultado de la operación está disponible para una instrucción posterior:

- a- Al finalizar la etapa WB.
- b- Al finalizar la etapa EX.**
- c- Al finalizar la etapa MEM.
- d- NS/NC.

37) Con referencia a un programa que ejecuta operaciones sobre números enteros y no hay instrucciones de punto flotante, decir cual de las siguientes opciones es verdadera:

- a- No se puede producir atascos por dependencia de datos WAR.
- b- No se puede producir atascos por dependencias de datos RAW.
- c- Se puede producir atascos por dependencias de datos WAW.**
- d- NS/NC.



38) ¿En qué etapa de la instrucción BNEZ R2, LOOP se conoce que es un salto, si la misma está incorporada a la tabla BTB?

- a- En la etapa ID.**
- b- En la etapa IF.
- c- En la etapa EX.
- d- NS/NC.

39) En un lazo que se ejecuta de forma repetida 100 veces (como un FOR), con la opción BTB habilitada, ¿cuántas veces funciona la estrategia en forma acertada?

- a- 98 veces.**
- b- 100 veces.
- c- Nunca.
- d- NS/NC.

40) ¿En base a qué condición se producen los saltos?

- a- Verificar el estado de los flags.
- b- Verificar el valor de un registro.**
- c- No se verifica ninguna condición porque todos los saltos son incondicionales.
- d- NS/NC.

41) ¿Cuántos ciclos tomará la ejecución completa de la siguiente instrucción? (suponga que no se producen stalls de ningún tipo): ADD.D F3, F2, F1

- a- 5 ciclos.
- b- 8 ciclos.
- c- 11 ciclos.**
- d- NS/NC.

iv = i 0 M 1 2 3 4 5 6 7 M M | W R S

42) ¿Bajo qué condiciones se puede producir un stall del tipo Branch Misprediction?

- a- Se puede producir en cualquier configuración.
- b- Habilitando forwarding.
- c- Habilitando Branch Target Buffer.**
- d- NS/NC.

Observaciones: Escribir las respuestas con tinta. Cada ejercicio indica su valor en puntos. **SE APRUEBA CON 10 PUNTOS. NOTA MAXIMA: 20.**

1. Dado el siguiente programa, indicar el contenido del registro r1 al finalizar la ejecución, considerando:

Opción Delay Slot NO habilitada \rightarrow (r1) = 2 H (\downarrow ph)

Opción Delay Slot habilitada \rightarrow (r1) = 32 H (\downarrow ph)

```
.data
A: .byte 1
B: .byte 5
loop: daddi r2, r2, -1
       bnez r2, loop
       dsll r1, r1, 1
       halt
```

2. Indique con tilde (~) ó cruz (X) cuáles de las siguientes instrucciones son correctas o incorrectas (+0,5 pts c/u).

[~] [X] daddi \$a2, \$a3, 75
[V] [~] lb \$t1, variable(\$zero)

[~] [X] /l.d \$t0, variable(\$zero)
[~] [X] beq \$a1, etiqueta

3. Bajo la convención de nombres de registros del WinMips, marque con cruz ¿qué uso tienen los registros \$a0 a \$a3? (1 pto correcta, -0,5 pts incorrecta, 0 si Ns/Nc)

[~] Valores de retorno de la subrutina llamada
[~] Registros salvados durante el llamado a subrutina

[X] Argumentos pasados a la subrutina llamada
[~] Ns/Nc

4. ¿Qué característica permite la ejecución efectiva de una instrucción sin tener en cuenta los efectos de la instrucción de salto precedente? (1 pto correcta, -0,5 pts incorrecta, 0 si Ns/Nc)

[~] Forwarding
[X] Delay Slot

[~] Branch Target Buffer
[~] Ns/Nc

5. Completar las dos instrucciones necesarias para convertir a punto flotante un valor entero ubicado en el registro r5 y dejarlo en el registro f9 (0,5 pts c/u).

mtc1 r5, f2
cvt,d,l f9, f2

6. Completar las tres instrucciones que faltan para leer un carácter desde el teclado, y que el código ASCII del carácter leído se almacene en el registro \$t3 (0,5 pts c/u).

```
.code
lwu $s6, CONTROL($0)
lwu $s7, DATA($0)
[REDACTED]
daddi $t0, $zero, 9
sd $t0, 0($s6)
lbu $t3, 0($s7)
```

7. El siguiente programa recorre un arreglo de números en punto flotante (TABLA1) y genera otro arreglo (TABLA2) con los números que están dentro del rango establecido por MIN y MAX. Además, guarda en RES la cantidad de elementos de TABLA2. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente (0,5 pts c/instrucción).

NOTA (del set de instrucciones): c.lt.d F_a, F_b; compara F_a con F_b, dejando flag FP=1 si F_a es menor que F_b (en punto flotante).
Bejt OFFSET; salta a la dirección rotulada OFFSET si flag FP=1 (ó true) (en punto flotante)

```
.data
TABLA1: .double 12.0, 15.5, 31.2, 56.4, 44.3, 78.1
MIN: .double 20.0
MAX: .double 50.0
CANT: .word 6
RES: .word 0
TABLA2: .double 0.0

.text
LD    $t0, CANT($zero)
L.D   F1, MIN($zero)
L.D   F2, MAX($zero)
DADD $t3, $zero, $zero
[REDACTED]
daddi $t0, $zero, 6
```

```
LOOP: DADD $t5, $zero, $zero
      L.D   F3, TABLA1($t4)
      C.LT.D F3, F1
      [REDACTED]
      C.LT.D F2, F3
      BC1T FUERA
      DADDI $t3, $t3, 1
      S.D   F3, TABLA2($t5)
      DADDI $t5, $t5, 8
FUERA: (daddi $t4, $t4, 8)
      DADDI $t0, $t0, -1
      BNEZ $t0, LOOP
      SD   $t3, RES($zero)
      HALT
```

8. Escribir un programa para Winmips que lea 3 números enteros (A, B y C) ingresados por el usuario desde el teclado, resuelva el cálculo $(A - B)^2$ y almacene el resultado en la memoria en la variable RES. El cálculo debe resolverse en una subrutina que reciba como parámetros los 3 operandos y retorne el valor del resultado. Finalmente, el valor calculado debe mostrarse en la pantalla alfanumérica. Debe utilizarse la convención para nombrar los registros que se empleen durante el programa (10 pts.).

```
.data
a: .word 0
b: .word 0
c: .word 0
control: .word32 0x10000
data: .word32 0x10008
res: .word 0
.code
lwu $s0, control($zero)
lwu $s1, data($zero)
jal ingreso
ld $a0, a($zero)
ld $a1, b($zero)
ld $a2, c($zero)
jal calculo
halt
```

ingreso: daddi \$t0, \$zero, 8
daddi \$t1, \$zero, 3
daddi \$t3, \$zero, 0
loop: daddi \$t1, \$t1, -1
sd \$t0, 0(\$s0)
ld \$t4, 0(\$s1)
sd \$t4, a(\$t3)
daddi \$t3, \$t3, 8
bnez \$t1, loop
jr \$ra

calculo: dsub \$t0, \$a0, \$a1 #(a-b) al cuadrado 2
daddi \$t1, \$zero, 1
loop1: dmul \$t1, \$t1, \$t0
daddi \$a2, \$a2, -1
bnez \$a2, loop1
sd \$t1, res(\$zero)
daddi \$t3, \$zero, 2
sd \$t1, 0(\$s0)
sd \$t3, 0(\$s0)
jr \$ra

7. Bajo la convención para el uso de registros, indique qué representan los siguientes tipos. 2 Puntos
- Sa0 - Sa3:
 - Sv0 - Sv1:
 - St0 - St7:
 - Sr0:

pasar argumentos a la subrutina
argumentos devueltos de la subrutina
registros temporales
contiene la dirección de retorno

8. Escribir un programa para WinMips que genere un arreglo llamado res, cuyos elementos sean el resultado de calcular la función $f = (a+b^2)/c$, siendo a, b y c tres arreglos de datos en punto flotante de 5 elementos cada uno, almacenados en memoria. Es decir, se deben hacer 5 operaciones (una por cada posición). El cálculo se debe realizar en una subrutina. El arreglo res debe almacenarse a continuación del arreglo c. ESCRIBA EL PROGRAMA EN ESTA MISMA HOJA. NO USAR OTRA HOJA. 8 Puntos.

```
.data
a: .double 1.3, 3.5, 5.8, 1.9, 2.7
b: .double 7.1, 3.4, 9.0, 1.0, 5.8
c: .double 2.3, 8.2, 0.5, 1.1, 2.9
```

INSTRUCCIÓN	CORRECTA	INCORRECTA
DADDI R3,R0,RI	✓	✗
LD R2, TEXTO (R1)		✗
LD F1, VALOR (F0)		✗
ADD.D F1,F0,2.0		✗

6. ¿Cuántas opciones para generar salidas en la pantalla alfanumérica tiene WinMIPS64? (1 pto)
- 5

enteros con y sin signo 2, punto flotante1,
, cadena ascii1, limpiar pantalla alfanumerica1

7. El siguiente programa recorre un arreglo de números en punto flotante (TABLA1) y genera otro arreglo (TABLA2) con los números que están dentro del rango establecido por MIN y MAX. Además, guarda en RES la cantidad de elementos de TABLA2. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente (1 pto c/instrucción).

```
.data
TABLA1: double 12.0, 15.5, 31.2, 56.4, 44.3, 78.1
MIN: .double 20.0
MAX: .double 50.0
CANT: .word 6
RES: .word 0
TABLA2: double 0.0
```

LOOP:	DADD \$t5, \$zero, \$zero LD \$t3, TABLA1(\$t4) CLTD \$t3, \$t1, \$t1
	CLTD \$t2, \$t3 BCIT \$t2, FUERA DADDI \$t3, \$t3, 1

4. Completar el siguiente código con las dos instrucciones que faltan: (1 pto c/u)

<pre>.text jal subrutina1 halt</pre>	<pre>subrutina2: addi \$t1, \$t0, 7 addi \$t5, \$t1, \$t0 addi \$t6, \$t5, \$t5</pre>
<pre>subrutina1: addi \$t21, \$t31, \$t0 jal subrutina2 ld \$ra, 0(\$sp) addi \$sp, \$sp, 8</pre>	<pre>jr \$ra</pre>

interpreto que alguien ya pusheo
la primera dirección en la pila

5. Complete con la cantidad de ciclos que tarda cada una de las siguientes instrucciones teniendo en cuenta que no se producen atascos. (0.25 pto c/u)

LWU R1, ETIQUETA (R0) 5

ADD.D F5, F6, F7 8 if id a0 a1 a2 a3 mm wb

5. Complete con la cantidad de ciclos que tarda cada una de las siguientes instrucciones teniendo en cuenta que no se producen saltos. (0.25 pto c/u)

LWUR1, ETIQUETA (R0) **S**

LD F1, NUMERO (R0) **S**

AJRD D \$5, \$6, \$7 **8**

MUL.D F2, \$9, \$10 **11**

if id m0 m1 m2 m3 m4 m5 m6 mm wb

6. Bajo la convención para el uso de registros, indiquen qué representan los siguientes tipos. (0.5 pto c/u)

- * \$t0 - \$t1 pasar argumentos a la subrutina
- * \$t0 - \$t7 argumentos devueltos de la subrutina
- * \$t0 - \$t1 registros temporales
- * \$t8 contiene la dirección de retorno

7. El siguiente programa recorre un arreglo de números en punto flotante (TABLA1) y genera otro arreglo (TABLA2) con los números que están dentro del rango establecido por MIN y MAX. Además, guarda en RES la cantidad de elementos de TABLA2. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente (1 pto c/instrucción).

```

TABLA1: .data
        .double 12.0, 15.5, 31.2, 56.4, 44.3, 78.1
MIN:    .double 20.0
MAX:    .double 50.0
CANT:   .word 6
RES:    .word 0
TABLA2: .double 0.0

.text
daddi $t0, $zero, 6      #longitud
ld   f1, MIN($zero)
ld   f2, MAX($zero)
daddi $t3, $zero, $zero
daddi $t4, $zero, 0       #desplazamiento

LOOP:   DADD  $t5, $zero, $zero
        LD    F3, TABLA1($t4)
        CLTD F3, F1
        bc1t fuera
        CLTD F2, F3
        BCIT FUERA
        DADDI $t3, $t3, 1
        SD    F3, TABLA2($t5)

FUERA:  DADDI $t5, $t5, 8
        daddi $t4, $t4, 8
        DADDI $t0, $t0, -1
        BNEZ $t0, LOOP
        SD    $t3, RES($zero)
        HALT

```

1. El siguiente programa dibuja un pixel azul en la ubicación (15,30) de la terminal gráfica de Winniips. Complete los espacios en blanco. (0.5 ptos c/u)

x y

```

.data
coorx: .byte 15
coory: .byte 30
color: .byte 0,0,255,0
CONTROL: .word32 0x10000
DATA: .word32 0x10008

.text
lwu $s6, CONTROL($0)
lwu $s7, DATA($0)
lbu $s0, coorX($zero)
sb $s0, 5($s7)
lbu $s1, coorY($zero)
sb $s1, 4($s7)
lwu $s2, color($zero)
sw $s2, 0($s7)
daddi $t0, $0, 5
sd $t0, 0($s6)

```

:cargar en \$s0 el valor de coorx
:cargar en \$s1 el valor de coory
:almacenar \$s0 en la ubicación correspondiente de DATA (Coordenada X a pintar)
:cargar en \$s1 el valor de coory
:almacenar \$s1 en la ubicación correspondiente de DATA (Coordenada Y a pintar)
:cargar en \$s2 el valor de color
:almacenar \$s2 en la ubicación correspondiente de DATA (Color a pintar)

1. Indicar el CPI de un programa con 10 instrucciones, 3 atascos por dependencia de datos y 1 atasco por dependencia de control. (0.5 pts)

CPI: $10+3+1+4/10$ 18/10 if id ex mm wb x10

2. ¿Qué mecanismo puedo usar para disminuir los atascos por dependencia de datos? (0.5 pts)
forwarding

3. Nombre y explique cómo funcionan los mecanismos que ofrece WinMIPS para reducir los atascos de control. (1 pto c/u)

branch target buffer: posee un historial de saltos, para predecir saltos
delay slot: ejecuta siempre una instrucción después del salto

4. ¿Qué tipo de instrucciones provocan los atascos estructurales en WinMIPS? (1 pto)
los atascos estructurales se deben por escases de recursos, se debe cuando 2 instrucciones quieren acceder al mismo tiempo a la etapa mm, suelen ser las instrucciones en p.f las que provocan estos atascos

5. Indicar con una X cuáles instrucciones son correctas y cuáles incorrectas. (0.5 pts c/u)

INSTRUCCIÓN	CORRECTA	INCORRECTA
DADDI R3,R0,R1		X
LD R2, TEXTO (R1)		

4. ¿Qué tipo de instrucciones provocan los atascos estructurales en WinMIPS? (1 pto)

5. Indicar con una X cuáles instrucciones son correctas y cuáles incorrectas. (0.5 pts c/u)

INSTRUCCIÓN	CORRECTA	INCORRECTA
DADDI R3,R0,R1		X
LD R2, TEXTO (R1)	V	
LD F1, VALOR (F0)		X
ADD.D F1,F0,2.0		X

6. ¿Cuántas opciones para generar salidas en la pantalla alfanumérica tiene WinMIPS64? (1 pto)
5

7. El siguiente programa recorre un arreglo de números en punto flotante (TABLA1) y genera otro arreglo (TABLA2) con los números que están dentro del rango establecido por MIN y MAX. Además, guarda en RES la cantidad de elementos de TABLA2. Completar el programa con las tres instrucciones faltantes de manera que funcione correctamente (1 pto c/instrucción).
ya lo hice, esta arriba

```
#Escriba un programa que recorra un vector de números almacenados en punto flotante, y calcule el valor promedio de los mismos. El valor promedio debe convertirse a entero.
```

```
#El calculo del promedio debe realizarse en una subrutina que reciba como parámetros la dirección de comienzo del vector y la cantidad de elementos, y devuelva el promedio (ya convertido a entero).
```

```
#El programa debe llamar a la subrutina para calcular el promedio, y luego pintar el pixel (0,0) de la pantalla con el color RGBA (P,0,0,0), donde P es el promedio calculado anteriormente.
```

Hecho, lo tengo que poner en el pdf o github

7. Escribir un programa para Winmips que solicite el ingreso de una cadena de caracteres por teclado y que finalice al leer el carácter 0. Luego, el programa debe imprimir la cadena ingresada al derecho y al revés. Escribir el programa en esta hoja. (10 pts)

<p>DNI: _____ Legajo: _____</p> <p>29/11/2022</p> <p>1. Dado el siguiente programa, indicar el contenido del registro \$t2 al finalizar su ejecución. (2pts)</p> <pre> .data num1: .byte 4 num2: .byte 3 .code lb \$t2, num1(\$zero) lb \$t3, num2(\$zero) </pre>	<pre> loop: daddi \$t3, \$t3, -1 bnez \$t3, loop dadd \$t2, \$t2, 1 halt </pre> <p>Delay Slot NO habilitada: (\$t2) = <u>8</u> (1pto) ✓</p> <p>Delay Slot habilitada: (\$t2) = <u>32</u> (1pto) ✓</p>
<p>2. ¿Qué instrucción falta en la subrutina Calculo para preservar la dirección de retorno al llamar a subrutina verificar? (1pto)</p> <p>Calculo: daddi \$sp, \$sp, -8 _____ ✓</p> <p>jal verificar</p> <p>3. Indique con un tilde <input checked="" type="checkbox"/> o una cruz <input type="checkbox"/> cuáles de las siguientes instrucciones son correctas o incorrectas. (0.5 ptos c/u)</p> <p><input checked="" type="checkbox"/> La instrucción es válida? <input checked="" type="checkbox"/> daddi \$a0, \$a1, 10 ✓ <input type="checkbox"/> xor \$t1, \$t2, \$t3 ✓</p> <p><input checked="" type="checkbox"/> l.d t0, valor(\$s0) X <input checked="" type="checkbox"/> beq \$a1, loop ✓</p> <p>4. El siguiente programa recorre una tabla de números (tablal) en PF y genera otra (tabla2) con los números de tablal que son mayores que el valor min. Además guarda en res la cantidad de elementos de tabla2. Completar instrucciones faltantes (3ptos.)</p>	
<pre> .data tablal: .double 12.0, 15.5, 31.2, 56.4, 44.3 min: .double 20.0 cant: .word 5 res: .word 0 tabla2: .double 0.0 .code ld \$t0, cant(\$zero) ld f1, min(\$zero) (1pto) ✓ dadd \$t3, \$zero, \$zero 0 dadd \$t4, \$zero, \$zero 0 dadd \$t5, \$zero, \$zero 0 </pre>	<p>lazo: <u>ld f2, tabla1(\$t4)</u> (1pto) ✓</p> <pre> c.le.d f2, f1 bclt no_mayor daddi \$t3, \$t3, 1 s.d f2, tabla2(\$t5) daddi \$t5, \$t5, 1 (1pto) ✓ </pre> <p>no_mayor: daddi \$t4, \$t4, 8 daddi \$t0, \$t0, -1 bnez \$t0, lazo sd \$t3, res(\$zero) halt</p>
<p>5. Escribir un programa que permite ingresar 5 caracteres por teclado, mostrando el mensaje "Ingrese 5 caracteres a buscar en la cadena"; por cada uno de ellos debe invocar a una subrutina CONTAR que recibe la dirección de comienzo de la cadena y el carácter ingresado, la misma debe devolver la cantidad de ocurrencias del carácter en la cadena. Los valores retornados se deben almacenar en una tabla en memoria. (5 pts)</p> <p>hecho</p> <pre> data Cadena: ascii "En esta cadena busco ocurrencias de un carácter" Mensaje: ascii "Ingrese 5 caracteres a buscar en la cadena" </pre>	
<p>6. Escribir un programa que recorra una tabla de 5 números cargada en memoria y pinte la pantalla gráfica en base a la misma. Se debe pintar de color azul las primeras 5 filas (las situadas en la parte superior) de la pantalla gráfica. Para cada fila se pintan de izquierda a derecha las primeras N_i columnas, donde N_i es el número i-ésimo contenido en la tabla, y el resto se dejan sin pintar. Si la tabla contiene los valores "3,6,2,5,1", se deben pintar los primeros 3 pixeles de la fila 49, los primeros 6 de la fila 48, etc. Suponer que el valor máximo cargado en Tabla no supera a 49. Usar la convención para nombrar a los registros. (7 ptos).</p> <p>hecho</p> <pre> data Tabla: word 3, 6, 2, 5, 1 </pre>	

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

```
.data  
numero: .double 2  
.code  
S l.d f1, numero($zero)  
S add.d f2, f2, f2  
S s.d f4, numero($zero)  
halt
```

CANTIDAD DE INSTRUCCIONES + 9 = 14 instrucciones

CPI : 1.7

Ejercicio	Puntos
1	0
2	1
3	0
4	5
5	0
6	0

2. ¿Qué instrucciones son necesarias para simular un PUSH a la pila del registro \$t6? (1p)

~~lbu \$t6, 0(\$t1)~~ ✓ ~~add \$t6, \$t1, \$t6~~

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

```
daddi $t7, $zero, 5  
daddi $t7, $t7, XX  
siguiente: daddi $t7, $t7, -1  
bnez $t7, siguiente
```

Cantidad : 1 ✗

4. Escribir un programa que lea desde teclado un número en punto flotante y lo compare con el valor almacenado en la celda llamada 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)

FL FO T → VALOR ActivaL

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

<pre>.data cadena: .asciiz 'ejemplo' result: .word 0 .code DIRECCIÓN DADDI \$T1, \$ZERO, CADENA T2:0 DADD \$T2, \$ZERO, \$ZERO</pre>	<p>loop: lbu \$t3, 0(\$t1) daddi \$t2, \$t2, 1 bnez \$t3, loop daddi \$t1, \$t1, 1 sd \$t2, result(\$zero) halt</p>
--	---

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 Rojo (255,0,0) en la coordenada (mínimo_tabla1, máximo_tabla2) y otro de color Verde (0, 255, 0) en la coordenada (mínimo_tabla2, máximo_tabla1). Usar la convención para nombrar a los registros. (8p)

```
tabla_1: .word 21, 12, 5, 16, 8, 39, 10, 41, 4, 33  
tabla_2: .word 32, 24, 15, 32, 17, 28, 11, 20, 44, 21
```