

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).

```
.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
```

Nro. RAWs: _____ BTS: _____ CPI: _____/_____

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: _____ 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
```

5. Completar las dos instrucciones necesarias para convertir a punto flotante un valor entero ubicado en el registro r5 y dejarlo en el registro t9 (0,5 pts c/u).

6. Completar las tres instrucciones que faltan para leer un caracter desde el teclado, y que el código ASCII del caracter 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_t ; compara F_d con F_t , dejando flag FP=1 si F_d es menor que F_t (en punto flotante)
Bc1t OFFSET ; salta a la dirección rotulada OFFSET si flag FP=1 (ó 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) LD F2, MAX(\$zero) DADD \$t3, \$zero, \$zero _____</pre>	<pre>LOOP: DADD \$t5, \$zero, \$zero LD F3, TABLA1(\$t4) C.LT.D F3, F1 _____ C.LT.D 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</pre>
---	---

2

8. Escribir un programa para Wimmips 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
.code	