

TC1017 – Solución de Problemas con Programación

Examen 2

Nombre: _____

Matrícula: _____

Fecha: 21.10.19

Lee cuidadosamente y contesta lo que se te pide. Este examen es individual.

Al momento de contestar, intenta ser lo más explícito posible: se calificará con base en lo que esté escrito.

Recuerda que puedes revisar material de la clase, el libro de texto o tus notas. Administra bien tu tiempo.

Buena suerte.

El tipo de tu examen dependerá de tu matrícula.

$$student = \begin{cases} Torr Olgosso & = [2, 0.5, 1, 1, 0.7] \text{ si el último dígito de tu matrícula mod } 3 = 0 \\ Antalie Naporá & = [0.1, 3, 2, 1, 0.2] \text{ si el último dígito de tu matrícula mod } 3 = 1 \\ Xandro Blooddrinker & = [3, 3, 0.5, 2, 3] \text{ si el último dígito de tu matrícula mod } 3 = 2 \end{cases}$$

Mi matrícula es _____, así que el último dígito de mi ID mod 3 es _____.

Por tanto me toca usar a _____ (2 %)

1. Técnicas de primer año (60 %)

Analiza cada uno de los siguientes códigos que sirven para calcular la fuerza de las técnicas que los estudiantes deben aprender el primer año de su estancia en el instituto.

1.1. Split Arrow

```
1 function output = split_arrow(x)
2     steps = [1 2 3];
3     difficulty = steps + 2;
4     cost = sum(difficulty);
5     output = cost * x;
6 end
```

■ split_arrow(student(1))

1.2. Riposte

```
1 function output = riposte(x)
2     output = x * sum((1:2:20) * 2) - 7;
3 end
    ■ riposte(student(2))
```

1.3. Heavy Strike

```
1 function output = heavy_strike(x)
2     steps = [1 2 3 2 2];
3     output = x * sum(steps) - 2;
4 end
    ■ heavy_strike(student(3))
```

1.4. Static Strike

```
1 function output = static_strike(x)
2     ease = 1:10;
3     steps = 11:20;
4     difficulty = ease + steps;
5     cost = sum(difficulty);
6     output = cost / x;
7 end
    ■ static_strike(student(4))
```

1.5. Frenzy

```
1 function output = frenzy(x)
2     difficulty = zeros(5,1);
3     steps = 3:8;
4     for i = 1:length(steps)
5         difficulty(i) = 1.5 * steps(i);
6     end
7     cost = difficulty * x;
8     output = sum(cost) - 4;
9 end
    ■ frenzy(student(5))
```

2. Técnicas de segundo año (30 %)

Los códigos anteriores generan los costos de cada una de las técnicas que *student* debe aprender en el tronco común de su estancia en la escuela de combate. Sin embargo, a partir del segundo año pueden escoger **uno de los siguientes dos cursos, o ambos**:

2.1. Tornado Shot

El tiro tornado—como se le conoce en español—es una de las técnicas más eficientes para atacar a múltiples oponentes al mismo tiempo. Para dominar esta técnica, se analizan múltiples factores como la distancia a la que debe hacerse el tiro inicial, la fuerza aplicada a la flecha y el número de proyectiles y oponentes cercanos. Por ello, el instituto te ha facilitado la siguiente tabla:

DTI	FA	NP	OC	DE
2	78	3	1	x
2	85	3	2	x
3	70	3	3	x
3	85	5	2	x
1	50	5	1	x

Sabiendo que el daño esperado **DE** se calcula de la siguiente manera:

$$DE = \frac{OC^2}{NP} \left(FA + \frac{FA + 1.06}{DTI} \right)$$

Escribe **dos funciones**:

1. Una función que reciba los datos necesarios de un tiro y que devuelva el daño esperado del mismo (15 %)
2. Una función que reciba la matriz mostrada y devuelva la misma matriz con la columna **DE** correctamente calculada (15 %)
 - **Reto**: Utiliza operaciones vectoriales en esta función en lugar de un `for` (+2 ★)

2.2. Power Siphon

A diferencia de *Tornado Shot*, que sufre una fuerte penalización entre menos oponentes cercanos encuentre, *Power Siphon* es una mejor opción para el combate uno a uno a distancia. Sin embargo, esta técnica gana poder sacrificando cada una de tus cargas de poder de concentración (**PC**) y por cada punto de equilibrio (**Qu**) de la mente, al igual que por cada nivel de abstracción espiritual (**Lv**).

¿Cuál es la clave para maximizar el daño final (**FD**)? Nadie está seguro, pero te han proporcionado las siguientes observaciones:

Lv	Qu	PC	FD
1	0	0	125
2	0	0	126
3	0	0	127
4	1	0	129.28
5	1	0	130.29
6	1	0	131.3
1	0	1	137.5
1	0	2	150
1	0	3	162.5
6	5	3	175.5
21	23	7	279.85

Sabiendo que el daño base (sin modificaciones) es siempre 125, contesta las siguientes preguntas y finalmente calcula lo que se pide:

- ¿Cuánto daño se obtiene por un nivel de abstracción espiritual *Lv*? (1 %) _____
- ¿Cuánto daño se obtiene por cada punto de equilibrio *Qu*? (2 %) _____
- ¿Cuánto daño se obtiene por cada carga de poder de concentración *PC*? (2 %) _____

Escribe **la función** matemática que modela el daño final de *Power Siphon* (10 %) y posteriormente escribe el código en MATLAB necesario para implementarla (15 %)

3. Feedback (8 %)

- Dibuja al estudiante que te tocó y ponle nombre a su instituto (2 %)
- Escribe una opinión breve sobre el modelo de examen. Si se pueden atacar situaciones fantásticas con programación, ¿crees que sirva para situaciones reales? ¿Por qué? (6 %)

4. Reto (+8 ★)

Escribe el Diagrama de flujo de cualquiera de las técnicas de segundo año.

De acuerdo con el Código de Ética del Tecnológico de Monterrey, mi desempeño en esta actividad estará guiado por la integridad académica.