

### Indicaciones específicas:

- Esta evaluación contiene 10 páginas (incluyendo esta página) con 4 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Crea la carpeta de nombre PC2.
- Crea el proyecto utilizando Pycharm
- Adiciona uno a uno los programas que dan respuesta a cada pregunta planteada
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta. Por ejemplo:
  1. p1.py
  2. p2.py
  3. p3.py
  4. p4.py
- Recuerda que el Gradescope solo conserva el último envío que se realiza, por lo tanto una vez que tengas las 4 preguntas resueltas, **deberás arrastrar los 4 archivos de manera simultánea y subirlos al Gradescope.**  
[www.gradescope.com](http://www.gradescope.com)
- **Para asignar el puntaje total a cada pregunta, es indispensable que en la solución se utilice las estructuras y/o conceptos que se indican en la rúbrica.**

### Criterios de desempeño:

- Para los alumnos de las carreras de **Ciencia de la Computación y Ciencia de Datos**:
  - 1.3 (nivel 1): Aplicar conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa.
  - 3.2 (nivel 1): Diseñar, implementar y evaluar soluciones a problemas complejos de computación.

- 4.1 (nivel 1): Crear, seleccionar, adaptar y aplicar técnicas, recursos y herramientas modernas para la práctica de la computación y comprende sus limitaciones.
  - Para los alumnos de las carreras de **Ingeniería:**
    - 1.3 (nivel 1): Aplica conocimientos de ingeniería en la solución de problemas complejos de ingeniería.
    - 3.2 (nivel 1): Diseña soluciones relacionadas a problemas complejos de ingeniería .
    - 5.1 (nivel 1): Crea, selecciona y utiliza técnicas, habilidades, recursos y herramientas modernas de la ingeniería y las tecnologías de la información, incluyendo la predicción y el modelamiento, con la comprensión de sus limitaciones.
  - Para los alumnos de la carrera de **Administración y Negocios Digitales**
    - 1.1 (nivel 1): Analizar información verbal y/o lógica proveniente de distintas fuentes, encontrando relaciones y presentándola de manera clara y concisa.
    - 2.3 (nivel 1): Resolver problemas pensando computacionalmente y empleando herramientas de programación
    - 4.5 (nivel 1): Integrar habilidades analíticas, digitales e interpersonales para el diseño de soluciones a problemas relevantes de personas y organizaciones.
- 

## Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	5	
2	5	
3	5	
4	5	
Total:	20	

1. (5 points) **Bucles anidados - Matriz de coordenadas**

Escriba una programa que solicite dos números enteros positivos  $M, N$  los cuales representan el número de filas  $M$  y el número de columnas  $N$  de un arreglo rectangular (no es necesario validar los datos ingresados). Luego, el programa debe imprimir las coordenadas de cada elemento del arreglo. Algunos ejemplos de diálogo de este programa son:

Listing 1: Ejemplo 1

```
Ingrese el numero de filas:3
Ingrese el valor de columnas:3

El arreglo de coordenadas es:

fil:1-col:1  fil:1-col:2  fil:1-col:3
fil:2-col:1  fil:2-col:2  fil:2-col:3
fil:3-col:1  fil:3-col:2  fil:3-col:3
```

Listing 2: Ejemplo 2

```
Ingrese el numero de filas:5
Ingrese el valor de columnas:4

El arreglo de coordenadas es:

fil:1-col:1  fil:1-col:2  fil:1-col:3  fil:1-col:4
fil:2-col:1  fil:2-col:2  fil:2-col:3  fil:2-col:4
fil:3-col:1  fil:3-col:2  fil:3-col:3  fil:3-col:4
fil:4-col:1  fil:4-col:2  fil:4-col:3  fil:4-col:4
fil:5-col:1  fil:5-col:2  fil:5-col:3  fil:5-col:4
```

Listing 3: Ejemplo 3

```
Ingrese el numero de filas:3
Ingrese el valor de columnas:4

El arreglo de coordenadas es:

fil:1-col:1  fil:1-col:2  fil:1-col:3  fil:1-col:4
fil:2-col:1  fil:2-col:2  fil:2-col:3  fil:2-col:4
fil:3-col:1  fil:3-col:2  fil:3-col:3  fil:3-col:4
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y codificación (4 pts)	Elabora un algoritmo preciso, definido y finito que da solución exacta a lo que el enunciado requiere. <b>Anida estructuras de control</b> para codificar el algoritmo y lo hace con el 100% de precisión. (4pts)	Elabora un algoritmo preciso, definido y finito que da solución al menos al 80 % de lo que el enunciado requiere. Anida estructuras de control para codificar el algoritmo y lo hace con al menos el 80% de precisión. (3pts)	Elabora un algoritmo preciso, definido y finito que da solución al menos al 65 % de lo que el enunciado requiere. Anida estructuras de control para codificar el algoritmo y lo hace con al menos el 65% de precisión. (2pts)	Elabora un algoritmo que hace menos del 65% de lo que el enunciado requiere. Anida estructuras de control para codificar el algoritmo y lo hace con menos del 65% de precisión. (0pts)
Sintaxis y legibilidad (1 pt)	El algoritmo es correcto, y es codificado sin errores de sintaxis. El nombre de las variables y funciones son descriptivas. (1pts)	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, pero que no afectan el resultado de manera significativa. El nombre de las variables y funciones son descriptivas. (0.75pts)	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, que afectan el resultado de manera mínima, o el nombre de las variables y funciones no son descriptivas. (0.5pts)	El algoritmo es incorrecto o es codificado con errores de sintaxis, que afectan el resultado de manera significativa. El nombre de las variables y funciones no son descriptivas. (0pts)

2. (5 points) **For - ADN**

Las cadenas de ADN estan conformadas por 4 bases nitrogenadas: adenina (A), timina (T), guanina (G) y citosina (C). El proceso mediante el cual el ADN envía un mensaje al citoplasma se denomina transcripción. En el proceso de transcripción, las cadenas de ADN generan una copia de estas mediante los siguientes pasos:

- Las bases adeninas generan timinas,
- las timina generan adeninas,
- las guanina generan citosina,
- las citosina generan guanina.

Por ejemplo:

- Una cadena de ADN

*ACTG*

genera la copia

*TGAC*

Se solicita que escriba un programa que realice la copia del ADN. Trabaje bajo el supuesto que el usuario ingresa los datos de manera correcta. Algunos ejemplos de diálogo de este programa serían:

Listing 4: Ejemplo 1

```
Ingrese su cadena de ADN:ACTG
La copia de ADN es: TGAC
```

Listing 5: Ejemplo 2

```
Ingrese su cadena de ADN:AATTGGCC
La copia de ADN es: TTAACCGG
```

Listing 6: Ejemplo 3

```
Ingrese su cadena de ADN:GGTTCCAA
La copia de ADN es: CCAAGGTT
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y codificación (4 pts)	Elabora un algoritmo preciso, definido y finito que da solución exacta a lo que el enunciado requiere. <b>Utiliza for</b> para codificar el algoritmo y lo hace con el 100% de precisión. <b>(4pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 80 % de lo que el enunciado requiere. Utiliza for para codificar el algoritmo y lo hace con al menos el 80% de precisión. <b>(3pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 65 % de lo que el enunciado requiere. Utiliza for para codificar el algoritmo y lo hace con al menos el 65% de precisión. <b>(2pts)</b>	Elabora un algoritmo que hace menos del 65% de lo que el enunciado requiere. Utiliza for para codificar el algoritmo y lo hace con menos del 65% de precisión. <b>(0pts)</b>
Sintaxis y legibilidad (1 pt)	El algoritmo es correcto, y es codificado sin errores de sintaxis. El nombre de las variables y funciones son descriptivas. <b>(1pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, pero que no afectan el resultado de manera significativa. El nombre de las variables y funciones son descriptivas. <b>(0.75pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, que afectan el resultado de manera mínima, o el nombre de las variables y funciones no son descriptivas. <b>(0.5pts)</b>	El algoritmo es incorrecto o es codificado con errores de sintaxis, que afectan el resultado de manera significativa. El nombre de las variables y funciones no son descriptivas. <b>(0pts)</b>

### 3. (5 points) **Funciones - Coprimos**

Dos números enteros mayores o iguales a 2 son coprimos si el único divisor que tienen en común es el 1. Así:

- 6 y 49 son coprimos, debido a que el único divisor en común es el 1.
- 24 y 32 no son coprimos, debido a que el 4 es un divisor de ambos números.

Elabore una función que determine si dos números son coprimos, además elabore un programa que solicite un entero positivo (debe validar este dato) e imprima todos los números coprimos menores a este (en el programa debe utilizar la función diseñada).

Algunos ejemplos de diálogo de este programa serían:

Listing 7: Ejemplo 1

```
Ingrese su numero:10
Los numeros coprimos y menores a 10 son:
3 7 9
```

Listing 8: Ejemplo 2

```
Ingrese su numero:-5
Ingrese su numero:1
Ingrese su numero:9
Los numeros coprimos y menores a 9 son:
2 4 5 7 8
```

Listing 9: Ejemplo 3

```
Ingrese su numero:15
Los numeros coprimos y menores a 15 son:
2 4 7 8 11 13 14
```

La rúbrica para esta pregunta es:

<b>Criterio</b>	<b>Excelente</b>	<b>Adecuado</b>	<b>Mínimo</b>	<b>Insuficiente</b>
Algoritmo y codificación <b>(4 pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución exacta a lo que el enunciado requiere. <b>Utiliza funciones</b> al codificar el algoritmo y lo hace con el 100% de precisión. <b>(4pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 80 % de lo que el enunciado requiere. Utiliza funciones al codificar el algoritmo y lo hace con al menos el 80% de precisión. <b>(3pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 65 % de lo que el enunciado requiere. Utiliza funciones al codificar el algoritmo y lo hace con al menos el 65% de precisión. <b>(2pts)</b>	Elabora un algoritmo que hace menos del 65% de lo que el enunciado requiere. Utiliza funciones al codificar el algoritmo y lo hace con menos del 65% de precisión. <b>(0pts)</b>
Sintaxis y legibilidad <b>(1 pt)</b>	El algoritmo es correcto, y es codificado sin errores de sintaxis. El nombre de las variables y funciones son descriptivas. <b>(1pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, pero que no afectan el resultado de manera significativa. El nombre de las variables y funciones son descriptivas. <b>(0.75pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, que afectan el resultado de manera mínima, o el nombre de las variables y funciones no son descriptivas. <b>(0.5pts)</b>	El algoritmo es incorrecto o es codificado con errores de sintaxis, que afectan el resultado de manera significativa. El nombre de las variables y funciones no son descriptivas. <b>(0pts)</b>



#### 4. (5 points) Cadenas - Contador de palabras

Elabore un programa que solicite una frase y una palabra como entradas; e imprima la cantidad de veces que aparece la palabra en la frase. Considere que:

- En la frase todas las palabras estan separadas por un espacio o una coma;
- la frase no contiene signos de puntuación (a excepción de la coma);
- no se utilizan tildes;
- el programa distingue entre mayúsculas y minúsculas, es decir, para el programa no son iguales las palabra *Hola* y *hola*.

Trabaje bajo el supuesto que el usuario ingresa los datos de manera correcta. Algunos ejemplos de diálogo de este programa serían:

Listing 10: Ejemplo 1

```
Ingrese la frase:hola hola juan
Ingrese la palabra a contar:hola
El numero de veces que aparece la palabra es: 2
```

Listing 11: Ejemplo 2

```
Ingrese la frase:hola,hola,hola juan
Ingrese la palabra a contar:hola
El numero de veces que aparece la palabra es: 3
```

Listing 12: Ejemplo 3

```
Ingrese la frase:hola hola hola juan
Ingrese la palabra a contar:mario
El numero de veces que aparece la palabra es: 0
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y codificación (4 pts)	Elabora un algoritmo preciso, definido y finito que da solución exacta a lo que el enunciado requiere. <b>Utiliza strings</b> y lo hace con el 100% de precisión. <b>(4pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 80 % de lo que el enunciado requiere. Utiliza strings y lo hace con al menos el 80% de precisión. <b>(3pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 65 % de lo que el enunciado requiere. Utiliza strings y lo hace con al menos el 65% de precisión. <b>(2pts)</b>	Elabora un algoritmo que hace menos del 65% de lo que el enunciado requiere. Utiliza strings y lo hace con menos del 65% de precisión. <b>(0pts)</b>
Sintaxis y legibilidad (1 pto)	El algoritmo es correcto, y es codificado sin errores de sintaxis. El nombre de las variables y funciones son descriptivas. <b>(1pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, pero que no afectan el resultado de manera significativa. El nombre de las variables y funciones son descriptivas. <b>(0.75pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, que afectan el resultado de manera mínima, o el nombre de las variables y funciones no son descriptivas. <b>(0.5pts)</b>	El algoritmo es incorrecto o es codificado con errores de sintaxis, que afectan el resultado de manera significativa. El nombre de las variables y funciones no son descriptivas. <b>(0pts)</b>