

### Indicaciones específicas:

- Esta evaluación contiene 12 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 PC4.
- 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) **Tema a evaluar:** Evalúa Recursividad

Realizar un programa que permita calcular la suma de los términos de una serie con una función recursiva. EL programa debe solicitar el número de términos al usuario y luego debe dar el resultado de la sumatoria de los términos:

$$S = 1 \times \frac{2}{3} + 2 \times \frac{3}{4} + 3 \times \frac{4}{5} + \cdots + n \times \frac{n+1}{n+2} \quad (1)$$

Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```
Cantidad de terminos: 3
Resultado:
4.566
```

Listing 2: Ejemplo 1

```
Cantidad de terminos: 6
Resultado:
17.435
```

Listing 3: Ejemplo 1

```
Cantidad de terminos: 18
Resultado:
157.195
```

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 recursividad</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 recursividad 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 recursividad 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 recursividad 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>

2. (5 points) **Tema a evaluar:** Evalúa archivos

Una farmacia están registrando los códigos de los medicamentos con las primeras letra de cada palabra del nombre, el número cero y un número correlativo. A continuación se tiene un ejemplo del archivo con los códigos utilizados .

Listing 4: Ejemplo de archivo medicamentos.txt

```
LS01, Levotiroxina sodica
MC01, Metformina clorhidrato
AI01, Aspirina infantil
CG01, Clonazepam gotas
```

Usando el archivo brindado (está disponible para descargar desde canvas), se te pide registrar nuevos codigos recibiendo el nombre del medicamento, y tu programa debe generar una clave que no se repita con los anteriores usando numeros consecutivos para generar el número al final del código.

**IMPORTANTE:** Debes actualizar el archivo medicamentos.txt con los nuevos códigos.

Algunos ejemplos de diálogo de este programa serían:

Listing 5: Ejemplo 1

```
Ingresar medicamento: Lorazepam solucion

#Archivo medicamentos.txt tiene el siguiente resultado:
LS01, Levotiroxina sodica
MC01, Metformina clorhidrato
AI01, Aspirina infantil
CG01, Clonazepam gotas
LS02, Lorazepam solucion
```

Listing 6: Ejemplo 1

```
Ingresar medicamento: Hidroclorotiazida tabletas

#Archivo medicamentos.txt tiene el siguiente resultado:
LS01, Levotiroxina sodica
MC01, Metformina clorhidrato
AI01, Aspirina infantil
CG01, Clonazepam gotas
LS02, Lorazepam solucion
HT01, Hidroclorotiazida tabletas
```

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 archivos</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 archivos 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 archivos 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 archivos al 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) **Tema a evaluar:** Evalúa el uso de algoritmos de búsqueda

La lista a continuación contiene información de las ciudades del Perú ubicadas a mayor altitud, incluyendo el departamento, la altitud y población de la ciudad.

Listing 7: Lista de distritos

```
lista = [
{'ciudad': 'La Rinconada', 'altitud': 5100, 'departamento': 'Puno', 'poblacion': 17000},
{'ciudad': 'Cerro de Pasco', 'altitud': 4380, 'departamento': 'Pasco', 'poblacion': 66860},
{'ciudad': 'Junin', 'altitud': 4105, 'departamento': 'Junin', 'poblacion': 10000},
{'ciudad': 'Yauri', 'altitud': 3976, 'departamento': 'Cusco', 'poblacion': 29772},
{'ciudad': 'Yanaoca', 'altitud': 3913, 'departamento': 'Cusco', 'poblacion': 11000},
{'ciudad': 'Ayaviri', 'altitud': 3907, 'departamento': 'Puno', 'poblacion': 20152},
{'ciudad': 'Lampa', 'altitud': 3878, 'departamento': 'Puno', 'poblacion': 14780},
{'ciudad': 'Ilave', 'altitud': 3862, 'departamento': 'Puno', 'poblacion': 22153},
{'ciudad': 'Azangaro', 'altitud': 3859, 'departamento': 'Puno', 'poblacion': 35230},
{'ciudad': 'Juli', 'altitud': 3850, 'departamento': 'Puno', 'poblacion': 23741},
{'ciudad': 'Yunguyo', 'altitud': 3847, 'departamento': 'Puno', 'poblacion': 11890},
{'ciudad': 'Juliaca', 'altitud': 3824, 'departamento': 'Puno', 'poblacion': 216716},
{'ciudad': 'Puno', 'altitud': 3810, 'departamento': 'Puno', 'poblacion': 120229},
{'ciudad': 'La Oroya', 'altitud': 3745, 'departamento': 'Junín', 'poblacion': 29417},
{'ciudad': 'Huancavelica', 'altitud': 3676, 'departamento': 'Huancavelica', 'poblacion': 41331},
{'ciudad': 'Sicuni', 'altitud': 3549, 'departamento': 'Cusco', 'poblacion': 42551},
]
```

Usando la lista mostrada se te solicita realizar un programa que calcule la diferencia de altitud y población entre dos ciudades, si alguna ciudad no se encuentra se debe indicar al usuario que no se puede calcular la diferencia.

Su programa debe solicitar el nombre de la ciudad y utilizar la búsqueda binaria para reportar todos los datos del diccionario respectivo y luego realizar el cálculo de la diferencia.

**IMPORTANTE:** En canvas se adjunta un archivo con la lista del ejemplo para que pueda realizar las pruebas de su solución.

Algunos ejemplos de diálogo de este programa serían:

Listing 8: Ejemplo 1

```
Ingrese el pais:Cerro de Pasco
{'ciudad':'Cerro de Pasco', 'altitud': 4380, 'departamento':
  'Pasco', 'poblacion': 66860}

Ingrese el pais:La Oroya
{'ciudad':'La Oroya', 'altitud': 3745, 'departamento':'Junín
  ', 'poblacion': 29417}

Diferencia de altitud: 635
Diferencia de poblacion: 37443
```

Listing 9: Ejemplo 2

```
Ingrese el pais:Arequipa
No fue encontrado

Ingrese el pais:Catar
{'ciudad':'Yauri', 'altitud': 3976, 'departamento':'Cusco',
  'poblacion': 29772}

No se puede hallar la diferencia
```

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 algoritmos de búsqueda</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 algoritmos de búsqueda 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 algoritmos de búsqueda 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 algoritmos de búsqueda al 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>

4. (5 points) **Tema a evaluar:** Evalúa el uso de algoritmos de ordenamiento

La lista a continuación contiene información de la calidad del aire de los países con el registro más bajo registrado en distintos países. Estos datos son del año 2020 y están tomados del Air Quality Life Index (ACLI) de la Universidad de Chicago.

La lista incluye datos de la contaminación de partículas finas del aire y el cálculo modelado de la pérdida potencial de expectativa de vida de la población debido a la contaminación .

Listing 10: Lista de distritos

```
calidad_aire = [  
{ 'pais': 'Bhutan', 'aire': 28.71, 'expectativa': -2.32 },  
{ 'pais': 'India', 'aire': 55.80, 'expectativa': -4.98 },  
{ 'pais': 'Nepal', 'aire': 47.13, 'expectativa': -4.13 },  
{ 'pais': 'Pakistan', 'aire': 44.17, 'expectativa': -3.84 },  
{ 'pais': 'Rwanda', 'aire': 32.95, 'expectativa': -2.74 },  
{ 'pais': 'Burundi', 'aire': 31.76, 'expectativa': -2.62 },  
{ 'pais': 'China', 'aire': 31.63, 'expectativa': -2.61 },  
{ 'pais': 'Equatorial Guinea', 'aire': 28.61, 'expectativa':  
  -2.31 },  
{ 'pais': 'Guatemala', 'aire': 28.45, 'expectativa': -2.30 },  
{ 'pais': 'Bangladesh', 'aire': 75.76, 'expectativa': -6.93 },  
{ 'pais': 'Cameroon', 'aire': 31.42, 'expectativa': -2.59 },  
{ 'pais': 'Qatar', 'aire': 29.19, 'expectativa': -2.37 },  
{ 'pais': 'Mongolia', 'aire': 31.47, 'expectativa': -2.59 },  
{ 'pais': 'Laos', 'aire': 28.00, 'expectativa': -2.25 },  
]
```

Con la lista dada, solicitar al usuario porque campo desea realizar el ordenamiento: país o expectativa y ordenar de forma ascendente por expectativa y de forma descendente por país. Finalmente, debe imprimir el país, calidad de aire y expectativa de vida reducida en el orden solicitado.

**IMPORTANTE:** En canvas se adjunta un archivo con la lista del ejemplo.

Algunos ejemplos de diálogo de este programa serían:

Listing 11: Ejemplo 1

```
Ingrese campo para ordenar (pais, expectativa): expectativa
La lista ordenada por expectativa ascendentemente es:
    Bangladesh      75.76  -6.93
      India          55.8   -4.98
      Nepal          47.13  -4.13
    Pakistan        44.17  -3.84
      Rwanda        32.95  -2.74
    Burundi         31.76  -2.62
      China          31.63  -2.61
    Cameroon        31.42  -2.59
    Mongolia        31.47  -2.59
      Qatar          29.19  -2.37
      Bhutan         28.71  -2.32
Equatorial Guinea   28.61  -2.31
    Guatemala       28.45   -2.3
      Laos           28.0   -2.25
```

Listing 12: Ejemplo 2

```
Ingrese campo para ordenar (pais, expectativa): pais
La lista ordenada por pais descendientemente es:
    Rwanda          32.95  -2.74
      Qatar          29.19  -2.37
    Pakistan        44.17  -3.84
      Nepal          47.13  -4.13
    Mongolia        31.47  -2.59
      Laos           28.0   -2.25
      India          55.8   -4.98
    Guatemala       28.45   -2.3
Equatorial Guinea   28.61  -2.31
      China          31.63  -2.61
    Cameroon        31.42  -2.59
    Burundi         31.76  -2.62
      Bhutan         28.71  -2.32
    Bangladesh      75.76  -6.93
```

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 algoritmos de ordenamiento</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 algoritmos de ordenamiento 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 algoritmos de ordenamiento 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 algoritmos de ordenamiento 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>