

Nombre y Apellido:

Legajo:

Examen Parcial de Programación 2

Tiempo mínimo para el examen: 1 hora reloj.

Tiempo máximo para el examen: 2 horas reloj.

El examen se compone los siguientes ejercicios. Se solicita documentar bien su código (por ej., cuando deba definir funciones/métodos indique claramente los tipos de los parámetros, el tipo del valor de retorno y explique brevemente qué hace la función/método).

1. Implemente una función recursiva llamada `invertir_cadena_rec` que recibe como parámetro una cadena de caracteres y devuelve la cadena invertida. También implemente una función iterativa `invertir_cadena_it` que realice la misma tarea.

Ejemplo de Uso:

- `invertir_cadena_rec('hola')` debería devolver ‘‘aloh’’.
- `invertir_cadena_it('mundo')` debería devolver ‘‘odnum’’

Aclaración: trabaje en todo momento con cadenas, no transforme a otro tipo de dato.

2. Considere la siguiente función:

```
def f(lista: list) -> int:  
    resp = 0  
    n = len(lista)  
    for i in range(n):  
        for j in range(n):  
            if (i+j) % 2 == 0:  
                resp += (lista[i] - lista[j])  
    return resp
```

y responda las siguientes preguntas:

1. ¿Cuántas operaciones se realizan en total en la función `f`?
 2. ¿Cuál es el orden de complejidad temporal de la función?
3. **Sistema de Gestión de Cursos**

Implementación de un sistema de gestión de gestión de cursos utilizando POO. Para ello, implemente las siguientes clases:

1. Clase Estudiante:

- Atributos que agrega:
 - `dni (str)`: DNI del estudiante.
 - `nombre (str)`: nombre del estudiante.
 - `cursos (list[Curso])`: lista de cursos en los que está inscrito el estudiante.
- Métodos:
 - `__init__`: inicializa los atributos.
 - `inscribir_curso`: inscribe al estudiante en un curso.

- `mostrar_cursos`: muestra todos los cursos a los que está inscrito.
- `__str__`: devuelve una representación en cadena del estudiante.

2. Clase Docente:

- Atributos:
 - `dni (str)`: DNI del docente.
 - `nombre (str)`: nombre del docente.
 - `especialidad (str)`: especialidad del docente.
- Métodos:
 - `__init__`: inicializa los atributos y llama al constructor de la clase base.
 - `__str__`: devuelve una representación en cadena del docente.

3. Clase Curso:

- Atributos:
 - `codigo_curso (str)`: código único del curso.
 - `nombre (str)`: nombre del curso.
 - `duracion (int)`: duración del curso en horas.
 - `docente (Docente)`: docente a cargo del curso.
 - `estudiantes (list[Estudiante])`: lista de estudiantes inscritos en el curso.
- Métodos:
 - `__init__`: inicializa los atributos.
 - `inscribir_estudiante`: inscribe un estudiante en el curso.
 - `mostrar_estudiantes`: muestra todos los estudiantes inscritos en el curso.
 - `__str__`: devuelve una representación en cadena del curso.

4. Clase GestionCursos:

- Atributos:
 - `cursos (dict[str, Curso])`: diccionario de cursos (clave: código del curso, valor: objeto Curso).
 - `estudiantes (dict[str, Estudiante])`: diccionario de estudiantes (clave: DNI, valor: objeto Estudiante).
- Métodos:
 - `__init__`: inicializa los atributos.
 - `agregar_curso`: agrega un curso al sistema.
 - `eliminar_curso`: elimina un curso del sistema.

- `agregar_estudiante`: agrega un estudiante al sistema.
- `inscribir_estudiante_en_curso`: inscribe a un estudiante a un curso dado su DNI.

```
# Ejemplo de uso
# Creando docentes
docente1 = Docente("12345678", "María González", "Matemáticas")
docente2 = Docente("87654321", "Carlos Rodríguez", "Historia")

# Creando cursos
curso1 = Curso("MAT101", "Matemáticas I", 30, docente1)
curso2 = Curso("HIS101", "Historia General", 40, docente2)

# Creando estudiantes
estudiante1 = Estudiante("87654321", "Juan Pérez")
estudiante2 = Estudiante("87654322", "Ana López")

# Creando la gestión de cursos
gestion = GestionCursos()

# Agregando cursos a la gestión
gestion.agregar_curso(curso1)
gestion.agregar_curso(curso2)

# Agregando estudiantes a la gestión
gestion.agregar_estudiante(estudiante1)
gestion.agregar_estudiante(estudiante2)

# Inscribiendo estudiantes en los cursos
gestion.inscribir_estudiante_en_curso("87654321", "MAT101") # Juan Pérez en Mat I
gestion.inscribir_estudiante_en_curso("87654322", "MAT101") # Ana López en Mat I
gestion.inscribir_estudiante_en_curso("87654321", "HIS101") # Juan Pérez en H. Gral

estudiante1.mostrar_cursos() # Muestra cursos de Juan Pérez
estudiante2.mostrar_cursos() # Muestra cursos de Ana López
```