

Problema 1		SI	NO
1	Implementa las cuatro funciones solicitadas		
2	Utiliza correctamente la sintaxis para definir funciones en Python: encabezado parte con la palabra reservada def , seguida del nombre de la función, seguido de los parámetros en paréntesis, seguido por dos puntos; el cuerpo de la función está indentado correctamente		
3	Las funciones están bien encapsuladas: la única entrada son los parámetros formales; la única salida es el valor de retorno; no usa variables que toman un valor fuera de la función; no coloca valores en variables que se usan fuera de la función		
4	Respetar los nombres de funciones indicadas en el enunciado; respetar los parámetros definidos en el enunciado para cada función		
5	Usa nombres representativos para las variables que utiliza; no usa variables demás; no escribe código sin propósito		
6	La función que calcula el promedio de una lista de notas funciona en los siguientes casos: [1.0, 3.0, 5.0] → 4.5 [3.0, 4.0, 3.0, 4.0] → 4.0 Devolviendo valores redondeados a un decimal		
7	La función que determina si la lista sólo contiene notas suficientes funciona en los siguientes casos: [4.0, 4.5, 5.0, 5.5] → True [3.0, 4.5, 5.0, 5.5] → False [4.0, 3.5, 5.0, 5.5] → False [4.0, 4.5, 5.0, 3.5] → False [4.0, 4.5, 3.0, 3.5] → False		
8	La función que elimina la peor nota funciona en los siguientes casos: [5.0] → [] [3.0, 4.0, 5.0] → [4.0, 5.0] [3.0, 3.0, 5.0] → [3.0, 5.0] [4.0, 3.0, 2.5, 5.0] → [4.0, 3.0, 5.0] [4.0, 5.0, 5.5, 2.0] → [4.0, 5.0, 5.5]		
9	La función que calcula el promedio semestral funciona en los siguientes casos: [1.0, 5.0, 4.4, 3.3] → Error [1.0, 5.0, 4.4, 3.3, 3.3] → 4.0 [5.7, 5.2, 5.7, 1.0, 1.0] → 5.5 [3.8, 5.6, 5.8, 1.0, 1.0] → 5.1 Devolviendo valores redondeados a un decimal y validando que la lista de notas recibida contenga cinco notas		

Problema 2		SI	NO
1	Implementa la función solicitada		
2	Utiliza correctamente la sintaxis para definir funciones en Python: encabezado parte con la palabra reservada def , seguida del nombre de la función, seguido de los parámetros en paréntesis, seguido por dos puntos; el cuerpo de la función está indentado correctamente		
3	La función está bien encapsulada: la única entrada son los parámetros formales; la única salida es el valor de retorno; no usa variables que toman un valor fuera de la función; no coloca valores en variables que se usan fuera de la función		
4	Reconoce que la función requiere un parámetro formal: el nombre del archivo de entrada; utiliza un nombre representativo para el parámetro		
5	Reconoce que la función debe construir y devolver un diccionario Python, que use como llave un string con el RUN del estudiante y como valor un flotante con la nota parcial registrada		
6	Usa nombres representativos para las variables que utiliza; no usa variables demás; no escribe código sin propósito		
7	La función lee correctamente con el archivo "controles.txt" devolviendo el diccionario: {'18218710-1': 5.0, '18366994-1': 5.1, '17621352-3': 4.4, '18554914-2': 4.5, '18657533-5': 5.7, '18586016-4': 5.8, '17818587-8': 4.2, '18583757-K': 5.2, '18235295-1': 5.8, '18639141-4': 4.7, '17646122-7': 5.5, '18622812-9': 4.3, '18976597-K': 5.0, '18638797-9': 6.1, '17746253-2': 4.6, '18677212-6': 6.3, '18726939-6': 4.3, '18346127-3': 4.9}		

Problema 3a		SI	NO
1	Implementa las tres subrutinas solicitadas, identificando que la primera (que crea la estructura “notas curso”) ha de ser una función, la segunda (para agregar una nota parcial) pueden ser una función o un procedimiento y que la tercera (para escribir el resultado consolidado) debe ser un procedimiento (= no devuelve valor)		
2	Utiliza correctamente la sintaxis para subrutinas funciones en Python: encabezado parte con la palabra reservada def , seguida del nombre de la subrutina, seguido de los parámetros en paréntesis, seguido por dos puntos; el cuerpo de la subrutina está indentado correctamente		
3	Las subrutinas están bien encapsuladas: la única entrada son los parámetros formales; no usa variables que toman un valor fuera de la función; no coloca valores en variables que se usan fuera de la función		
4	Reconoce que la estructura “nota curso” debe implementarse como un diccionario Python, que use como llave un string con el RUN del estudiante y como valor una “lista de notas”		
5	Respeto los nombres de subrutinas indicadas en el enunciado; respeta los parámetros definidos en el enunciado para cada subrutina		
6	Usa nombres representativos para las variables que utiliza; no usa variables demás; no escribe código sin propósito		
7	La función que crea una estructura “notas curso” transfiere el diccionario de notas parciales recibido correctamente. Probar el siguiente caso: {'11111111-1': 5.0, '22222222-2': 5.1, '33333333-3': 3.4} → {'11111111-1': [5.0], '33333333-3': [3.4], '22222222-2': [5.1]}		
8	La subrutina que agrega una nota parcial a la estructura “notas curso” transfiere el diccionario de notas parciales recibido correctamente. Estudiantes que no estaban registrados aparecen con un 1.0 en la nota anterior. Estudiantes que están registrados en el curso pero que no tienen la nueva nota parcial, aparecen con un 1.0 como nueva nota parcial. Probar el siguiente caso: {'11111111-1': [5.0], '33333333-3': [3.4], '22222222-2': [5.1]} + {'11111111-1': 4.0, '33333333-3': 5.6, '44444444-4': 4.3} → {'44444444-4': [1.0, 4.3], '11111111-1': [5.0, 4.0], '33333333-3': [3.4, 5.6], '22222222-2': [5.1, 1.0]}		
9	La subrutina que escribe el resultado semestral según una estructura “notas curso” genera el archivo de salida correctamente. La subrutina usa las operaciones definidas para la estructura “lista de notas”. Los estudiantes aparecen ordenados por RUN, de menor a mayor. Probar el siguiente caso: {'11111111-1': 5.0, '22222222-2': 5.0, '33333333-3': 3.5} + {'11111111-1': 4.0, '33333333-3': 5.5, '44444444-4': 4.0} + {'11111111-1': 4.5, '44444444-4': 4.0, '22222222-2': 4.0, '33333333-3': 6.0} + {'44444444-4': 5.0, '11111111-1': 1.0, '22222222-2': 5.0, '33333333-3': 1.0} + {'11111111-1': 1.0, '22222222-2': 3.0, '33333333-3': 1.0, '44444444-4': 1.0} genera el archivo de salida: 11111111-1 5.0 4.0 4.5 1.0 1.0 4.5 22222222-2 5.0 1.0 4.0 3.0 5.0 4.3 33333333-3 3.5 5.5 6.0 1.0 1.0 5.0 44444444-4 1.0 4.0 4.0 1.0 5.0 3.5		

Problema 3b		SI	NO
1	Presenta un modelo de solución, que tiene claramente tres secciones: entrada de datos, procesamiento de datos y salida de datos; alternatively puede mezclar la entrada de datos y el procesamiento (es decir, leer un archivo de notas parciales, agregarlo a las notas del curso, leer el siguiente archivo de notas parciales, y así sucesivamente)		
2	Se observa una estrategia general que sigue un orden lógico que lleva a la solución del problema		
3	Hace referencia a los tipos de datos abstractos propuestos en el enunciado y utiliza las operaciones definidas para ellos (aunque no las haya implementado)		
4	El modelo, una vez implementadas las estructuras de datos correspondientes, puede escribirse fácilmente en Python		
5	La respuesta es ordenada y puede seguirse fácilmente; se observa preocupación por la caligrafía y ortografía		