

1.a)		20 puntos				
f1						
li	i	v	p			
[3,2,1,7,4]				1 pto	Pantalla	
	1					
		2			[1, 2, 3, 4, 7]	2 ptos
			1	1 ptos		
[3,3,1,7,4]					<i>Considerar el print solo si demuestra saber hacer el ruteo.</i> <i>Los puntos se asignan por bloque, no ingresar puntos intermedios.</i> <i>Bloque bueno, bloque con puntos.</i>	
			0			
[2,3,1,7,4]				2 ptos		
	2					
		1				
			2	1 ptos		
[2,3,3,7,4]						
			1			
[2,1,3,7,4]				2 ptos		
[2,2,3,7,4]						
			0			
[1,2,3,7,4]				2 ptos		
	3					
		7				
			3	1 ptos		
	4					
		4				
			4	1 ptos		
[1,2,3,7,7]						
			3			
[1,2,3,4,7]				2 ptos		
1.b)	Ordena una lista de menor a mayor			5 ptos	<i>*Ordena una lista (3 ptos)</i>	

Pregunta 2:		40 pts		
Item	0 Pto	4 Ptos	7 Ptos	11 Ptos
Manejo de estructuras de datos	No demuestra saber como manejar las estructuras de datos del problema	Demuestra saber como manejar las estructuras de datos utilizadas al menos en una función.	Demuestra uso impecable de las estructuras de datos en todas las funciones.	
Función cantidades	No demuestra nociones de cómo resolver el problema	La función tiene la lógica descrita pero comete 1 o 2 errores. Ej: recorre la lista pero de forma incorrecta, en una misma variable cuenta las cantidades de cada fruta, de usar diccionario, no sabe como agregar una llave nueva a éste.	La lógica de la solución está correcta pero comete algún error menor. Ej: de utilizar diccionario, retorna este en vez del .items(), cuenta una fruta de más o una de menos.	La función está correcta completamente.
Función frutas_mas_repetidas	No demuestra nociones de cómo resolver el problema	La función tiene la lógica descrita pero comete 1 o 2 errores. Ej: ocupa mal el método sort, no retorna una lista sino la fruta mayor, la lista tiene cantidades en vez del nombre de la fruta.	La lógica de la solución está correcta pero comete algún error menor. Ej: agrega solo una fruta mayor a la lista y no todas, de ordena lo hace por el criterio equivocado.	La función está correcta completamente.
Función fruta_exclusiva	No demuestra nociones de cómo resolver el problema	La función tiene la lógica descrita pero comete 1 o 2 errores. Ej: asume 12 meses, no recorre correctamente los meses.	La lógica de la solución está correcta pero comete algún error menor. Ej: de ocupar el complemento de conjuntos incluye el mes evaluado, equivoca el signo del complemento.	La función está correcta completamente.
Descuentos		-4 Puntos		
Sintaxis y expresiones	No comete errores graves con respecto al lenguaje; a lo más errores atribuibles más a descuidos que ignorancia (por ejemplo, en una ocasión olvidar los dos puntos al final de la línea, confunde = y == o confunde () y [] o , ocupa ≤ en lugar de <=, pero el resto de las veces lo hace bien)	Demuestra no entender construcciones importantes del lenguaje (por ejemplo: no indenta, le pone una condición al else, hace while con rango, confunde el string x con la variable x, asigna valor a variables de forma incorrecta 4=x). Un descuento de -4 por tipo de error.		

Pregunta 3:		40 pts		
Item	0 Pto	3 Ptos	6 Ptos	10 Ptos
Manejo de conjuntos	No demuestra saber como manejar los conjuntos del problema	Demuestra saber como recorrer un conjunto o realizar operaciones de conjuntos, pero comete ciertos errores en algunos casos proceso, ej: realiza intersección en vez de unión, agrega valores a un conjunto mientras lo recorre.	Demuestra uso impecable de los conjuntos en todo el problema.	
Manejo de diccionarios	No demuestra saber como manejar el diccionario del problema	Demuestra saber como recorrer un diccionario, pero comete ciertos errores en algunos casos proceso, ej: olvida paréntesis del .items(), confunde llave con valor o vice-versa.	Demuestra uso impecable de los diccionarios en todo el problema.	
Filtra a posibles candidatos (Considera solo a los vivos)	No demuestra nociones de cómo resolver el problema	Recorre todos los posibles candidatos al trono, pero incluye nobles muertos entre las posibilidades. Filtra a algunos muertos no a Todos	Determina a todos los posibles candidatos al trono correctamente, filtrando a los nobles ya muertos.	
Determinar padres de un noble	No demuestra nociones de cómo resolver el problema	Logra identificar los padres de un noble en particular, pero comete algún error menor en el proceso, ej: asume que siempre hay un padre y una madre.	Logra identificar los padres de un noble en particular correctamente.	
Determinar cantidad de generaciones nobles.	No demuestra saber como solicitar los datos y mostrar el resultado final	Cuenta su generación y la de sus padres (si corresponde), pero no sabe como seguir identificando si hay generaciones más antiguas.	Reune todos los ancestros de un noble, pero no logra contar la cantidad de generaciones reunidas. O cuenta la cantidad de generaciones para un noble pero con errores, ej: no sabe cuando terminar de buscar si quedan generaciones por contar, o solo considera un padre por noble (no padre y madre).	Resuelve este problema correctamente.
Generar lista de candidatos	No demuestra nociones de cómo resolver el problema	Compara correctamente la cantidad de generaciones de cada noble, pero sólo guarda la cantidad mayor y no a quienes corresponde, o indica un sólo posible rey y no una lista de candidatos.	Entrega correctamente una lista con el nombre de los candidatos a rey.	
Descuentos		0 Pto	-4 Puntos	
Sintaxis y expresiones	No comete errores graves con respecto al lenguaje; a lo más errores atribuibles más a descuidos que ignorancia (por ejemplo, en una ocasión olvidar los dos puntos al final de la línea, confunde = y == o confunde () y [] o , ocupa ≤ en lugar de <=, pero el resto de las veces lo hace bien)		Demuestra no entender construcciones importantes del lenguaje (por ejemplo: no indenta, le pone una condición al else, hace while con rango, confunde el string x con la variable x, asigna valor a variables de forma incorrecta 4=x). Un descuento de -4 por tipo de error.	