

<b>Pregunta 1: 20 ptos</b>						
<b>def disp(IDVIAJE, cap, compras):</b>	<b>0.5 pts</b>					
v=open(compras)	<b>1 pto</b>					
for linea in v:	<b>1 pto</b>					
id,cant,rut=linea.strip().split(':')	<b>1 pto</b>					
if IDVIAJE == id:	<b>1 pto</b>					
cap -= int(cant)	<b>1 pto</b>					
v.close()	<b>1 pto</b>					
return cap	<b>1 pto</b>					
<b>def informar(viajes, compras):</b>	<b>0.5 pts</b>					
infor=open('informe.txt','w')	<b>1 pto</b>					
texto='{0}#{1}#{2}\n'	<b>1 pto</b>					
viaj=open(viajes,'r')	<b>1 pto</b>					
for linea in viaj:	<b>1 pto</b>					
linea=linea.strip().split('#')	<b>1 pto</b>					
id,ruta,hora,cap=linea	<b>1 pto</b>					
cap = int(cap)	<b>1 pto</b>					
cupos = disp(id,cap,compras)	<b>1 pto</b>					
nline=texto.format(ruta,hora,cupos)	<b>1 pto</b>					
infor.write(nline)	<b>1 pto</b>					
viaj.close()	<b>1 pto</b>					
infor.close()	<b>1 pto</b>					
<b>Las líneas que están en colores pueden estar en otra posición, no es la única posición</b>						
<b>Podrían estar intercambiadas entre ellas.</b>						
<b>Revisar que la función esté correcta, la solución expuesta NO ES LA ÚNICA</b>						

Pregunta 2:	40 pts				
Item	0 Ptos	4 Ptos	6 Ptos	9 Ptos	12 Ptos
manejo de funciones y estructuras de datos	No demuestra saber cómo trabajar con las estructuras de datos, funciones ni archivos	Sabe trabajar con funciones y estructura de datos (definirlas y usarlas) y trabajar con archivos.			
Función status()	No demuestra saber cómo implementar la función status, o bien comete más de 6 errores en la implementación de su solución.	Demuestra saber cómo implementar la función, pero en su implementación comete <b>a lo más 6 errores</b> , tales como: 1.- No abre el archivo en el modo correcto 2.- No crea el diccionario vacío 3.- Comete error en la separación de las líneas del archivo 4.- No busca al tomadotchi pasado como parámetro 5.- No considera el caso en que el tomadotchi esté muerto 6.- No cierra el archivo (o cierra el archivo después del return) 7.- Otros	Demuestra saber cómo implementar la función, pero en su implementación comete <b>a lo más 4 errores</b> de los detallados en el punto anterior.	Demuestra saber cómo implementar la función, pero en su implementación comete <b>a lo más 2 errores</b> como los señalados en el ítem de los 4 pts.	Implementa correctamente la solución de la función status, <b>sin cometer errores</b> en su implementación
Función alimento()	No demuestra saber cómo implementar la función status, o bien comete más de 6 errores en la implementación de su solución.	Demuestra saber cómo implementar la función, pero en su implementación comete <b>a lo más 6 errores</b> , tales como: 1.- No abre el archivo en el modo correcto 2.- No crea el diccionario vacío 3.- Comete error en la separación de las líneas del archivo 4.- No busca al alimento pasado como parámetro 5.- No considera el caso en que el alimento no se encuentre en el archivo 6.- No cierra el archivo (o cierra el archivo después del return) 7.- Otros	Demuestra saber cómo implementar la función, pero en su implementación comete <b>a lo más 4 errores</b> de los detallados en el punto anterior.	Demuestra saber cómo implementar la función, pero en su implementación comete <b>a lo más 2 errores</b> como los señalados en el ítem de los 4 pts.	Implementa correctamente la solución de la función alimento, <b>sin cometer errores</b> en su implementación
Función alimentar()	No demuestra saber cómo implementar la función status, o bien comete más de 6 errores en la implementación de su solución.	Demuestra saber cómo implementar la función, pero en su implementación comete <b>a lo más 6 errores</b> , tales como: 1.- Si no usa las funciones anteriores, considerar los posibles errores detallados en este mismo ítem (4pts) pero para las otras funciones. 2.- Llama a las funciones anteriores pero no pasa los parámetros correctos o no los pasa. 3.- No considera el caso de que el Tomadotchi esté muerto 4.- No considera el caso de que el Tomadotchi esté vivo pero no se encuentre el alimento 5.- No crea un diccionario vacío para actualizar el estado del Tomadotchi 6.- No sabe actualizar (sumar) los diccionarios del tomadotchi con el del alimento 7.- Usa print en vez de return 8.- otros	Demuestra saber cómo implementar la función, pero en su implementación comete <b>a lo más 4 errores</b> como los señalados en el ítem de los 4 pts.	Demuestra saber cómo implementar la función, pero en su implementación comete <b>a lo más 2 errores</b> como los señalados en el ítem de los 4 pts.	Implementa correctamente la solución de la función alimentar, <b>sin cometer errores</b> en su implementación
<b>Descuentos</b>	<b>0 Pto</b>		<b>-4 Puntos</b>		
Forma y expresiones	No comete errores graves con respecto a la forma; a lo más errores atribuibles más a descuidos que ignorancia (por ejemplo, olvida poner un 'si' o un 'no' en los distintos caminos después de una condición, olvida una o dos flechas en los conectores entre bloques)		Demuestra no entender construcciones importantes de los diagramas (por ejemplo: No identifica el camino del 'si' o 'no' después de los condicionales, no uno los bloques por flechas, sólo líneas.		

Pregunta 3:	40 pts			
Item	0 Pts	5 Pts	10 Pts	15 Pts
Manejo de archivos y procesamiento de texto.	No demuestra tener noción de cómo usar archivos ni procesamiento de texto.	Demuestra tener conocimiento de cómo usar correctamente las funciones de procesamiento de textos y archivos		
Función actualizar(ad, ah,ah2)	No crea la función solicitada o no tiene nociones de cómo hacerlo.	Crea la función pero comete <b>3 o 4 errores</b> como: 1) No transforma los '0' y '1' a tipo entero o flotante 2) No calcula correctamente probabilidad de padecer enfermedad 1 o 2 3) No considera el carácter espacio en blanco en archivo 'historico.txt' 4) No captura de manera adecuada las probabilidades de enfermedad 1 o 2 del archivo 'historico.txt'.	Crea correctamente la función pero comete <b>1 o 2 errores</b> .	Crea correctamente la función sin cometer errores.
Escritura del archivo 'historico2.txt'	No demuestra tener noción de cómo generar el archivo 'historico2.txt' con la información y formato solicitado.	Crea correctamente el archivo 'historico2.txt' en modo escritura pero comete <b>3 o 4 errores</b> leves como: 1) No transforma resultados numéricos a string, 2) No agrega el salto de línea '\n' al final de cada línea o el carácter espacio antes de cada porcentaje de enfermedad 3) Hace un split() después de un split() 4) No quita los espacios en blanco o el caracter '%' para obtener los datos históricos	Crea correctamente el archivo en modo escritura pero comete <b>1 o 2 errores</b> de los señalado en el punto anterior	Crea correctamente el archivo en modo escritura y no comete errores de escritura de líneas, respeta los tipos de datos y el formato y considera los siguientes casos: 1.- Si el paciente estaba en el histórico y no en el de datos 2.- Si estaba en el histórico y en el de datos 3.- Si no estaba en el histórico y sí en el de datos <b>Por cada caso no considerado quitar 3 pts a la pregunta.</b>
Cálculo de probabilidad de enfermedad 1 y 2	No demuestra tener noción de como hacer el cálculo de la probabilidad para las enfermedades 1 y 2	Implementa correctamente el cálculo de la probabilidad de las enfermedades 1 y 2. Considera el género del paciente y convierte a entero la existencia o no de la sustancia.		
<b>Descuentos</b>	<b>0 Pto</b>		<b>-4 Puntos</b>	
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.	