Modelos de ejercicios de Funciones 2do parcial

Ejercicio 1:

Se desea implementar un termostato dentro de un invernadero que cuenta con 10 ambientes distintos, y se cuenta con las siguientes funciones ya implementadas:

- ✓ obtenerGradosF(n) que retorna la temperatura del ambiente *n* en grados Farenheit.
- ✓ transformar(gradosF) que recibe la temperatura en grados Farenheit y devuelve su correspondiente en grados Celsius.
- ✓ encenderCalefon(n), apagarCalefon(n), encenderAire(n), apagarAire(n) que controlan el calefon (o aire) en el ambiente n
- ✓ estaPrendidoCalefon(n), estaPrendidoAire(n) que indica si esta prendido o apagado el calefon (o aire) en el ambiente *n*.
- ✓ esperarSegundos(n), que pausa el programa por n segundos

Se desea chequear la temperatura en cada ambiente cada 60 segundos, para que se mantenga entre aproximadamente 20 y 23 grados Celsius. Si la diferencia es mayor que 3 grados, se debe prender el aparato correspondiente (aire o calefactor) y verificar cada 1 segundo el nuevo valor de temperatura hasta que esté dentro del rango deseado, momento en el cual debe apagarse el calefactor (o aire). Realizar un programa que controle la temperatura de todos los ambientes del invernadero.

Ejercicio 2:

Hacer un programa para automatizar la inscripción de estudiantes a materias de la Universidad. Usted dispone de una lista con los DNI de los estudiantes y otra con las materias a las que desean inscribirse. El programa debe modificar estas listas de forma tal que solo queden DNI y materias aceptadas.

Para ello cuenta con las siguientes funciones:

- ✓ correlativas(materia): Devuelve una lista de las correlativas a esa materia.
- ✓ aprobada(dni, materia): Devuelve verdadero si el estudiante tiene aprobada la materia.
- ✓ horario(materia): Devuelve el horario de la materia.
- ✓ controlaSuperposicion(dni, horario) Devuelve verdadero si el estudiante tiene libre ese horario.

DNI:

30435473	35552144	11421159	28667796	26959663	30435473	29927766

Materias:

Ingles I	Programación	Ingles I	Matemática	Matemática	Programación	Ingles I

Ejercicio 3:

Se desea implementar un sistema que controle que los servidores de un sitio web se encuentren funcionando.

Se cuenta con las siguientes funciones ya implementadas.

- ✓ pedirServidores(): Devuelve una lista de enteros, donde cada entero representa un servidor.
- ✓ funciona(servidor): Devuelve true si el servidor está funcionando (si nos podemos conectar) y false sino.
- ✓ tiempoRespuesta(servidor): Se conecta al servidor e indica el tiempo de respuesta actual del servidor en milisegundos. No podemos preguntarle el tiempo de respuesta a un servidor que no funciona.
- ✓ tiempoRespuestaMaximo(servidor):Indica el tiempo máximo de respuesta aceptable para un servidor determinado, en milisegundos
- ✓ buscarAdministrador(servidor):Dado un servidor, devuelve un entero que representa al administrador a cargo de ese servidor.
- ✓ enviarSMS(mensaje, administrador): Envía un SMS a un administrador determinado.

- ✓ registrarTiempoRespuesta(servidor, tiempoRespuesta): Almacena en un archivo el tiempo de respuesta para un servidor dado, y la fecha y hora en que esto se produjo.

Escribir una función que realice el control de los servidores. Si un servidor no funciona, o funciona pero su tiempo de respuesta supera el máximo aceptable, se deberá notificar esta situación al administrador de ese servidor. Además, la función debe registrar todos los tiempos de respuesta medidos.

Ejercicio 4:

El equipo de Arnaldito se inscribió al torneo de supervivencia de robots. En un partido varios robots participan en una plataforma cerrada y gana el equipo de robots que logra tener el último robot sobreviviente. Aquel robot que sufre una colisión con algún obstáculo, borde u otro robot es liquidado. Cada robot cuenta con cuatro sensores y cuatro turbinas que le permiten conocer la distancia a los objetos más cercanos y movilizarse. Se cuenta con las siguientes funciones:

- ✓ distanciaNorte(n), distanciaSur(n), distaciaEste(n), distanciaOeste(n): Devuelven la distancia en cm del robot n al objeto más cercano en esa dirección.
- ✓ prenderNorte(n): Prende la turbina ubicada en extremo norte del robot n provocando un movimiento hacia el sur.
- ✓ prenderSur(n): Prende la turbina ubicada en extremo norte del robot n provocando un movimiento hacia el norte.
- ✓ prenderEste(n): Prende la turbina ubicada en extremo norte del robot n provocando un movimiento hacia el oeste.
- ✓ prenderOeste(n): Prende la turbina ubicada en extremo norte del robot n provocando un movimiento hacia el este.
- ✓ apagarNorte(n), apagarSur(n), apagarEste(n), apagarOeste(n): Apagan la turbina correspondiente del robot n.
- ✓ dameVivos(), devuelve una lista con los números de los robots de su equipo que siguen en competencia.

Arnaldito diseñó la siguiente estrategia para su equipo de robots y necesita que alguien la programe. Si la distancia a algún objeto es menor a 10 cm debe intentar alejarse de este, de lo contrario debe mantener apagada la turbina correspondiente a esa dirección para ahorrar recursos, juega hasta que pierde todos sus robots.

Será descalificado del juego aquel robot que tenga encendidas dos turbinas de direcciones opuestas al mismo tiempo.

Hacer una función que controle al equipo de robots, esta función será llamada desde un programa principal cada 10 milisegundos.

(Está permitido apagar -o prender- las turbinas aunque ya se encuentren apagadas -o prendidas-)

Ejercicio 5:

Starbuck nos pidió ayuda para administrar sus desayunos:

Los desayunos se representan con una lista:

desayunos = [$desayuno_1$, $desayuno_2$, … $desayuno_n$]

Donde cada desayuno es una lista de productos, por ejemplo:

$desayuno_1$ = [“café”, “manteca”]

$desayuno_2$ = [“manteca”, “medialuna”, “medialuna”]

Se necesita hacer un programa que decida si los pedidos de desayuno se pueden cumplir. Es decir:

- Que se tenga stock de todos los productos
- Que ningún producto este vencido

Para ello se cuentan con las funciones:

- ✓ Stock(producto, cantidad): que devuelve verdadero si el producto tiene esa cantidad de stock.
- ✓ Vencido(producto): que devuelve verdadero si el producto esta vencido
- ✓ Costo(producto): que devuelve el costo del producto
- ✓ Venta(desayuno): que devuelve el precio de venta de un desayuno

Se deberá informar

- - Los desayunos que no se pueden cumplir.
- - La suma de las ganancias de los desayunos que si se pueden cumplir.

La ganancia se interpreta como: venta – costo.

Ejercicio 6:

Una agencia de alquiler de automóviles alquila autos de distintas categorías a sus clientes. Existen dos tipos de alquiler, uno que es ilimitado en cuando a kilometraje realizado y otro que se tiene un kilometraje base y si el cliente se pasa de ese kilometraje se le cobra además de por los días de alquiler también por el kilometraje excedido. El precio del kilometraje excedido depende de la categoría del auto. En el caso de ser el plan ilimitado, el alquiler se calcula en base a la cantidad de días por el costo diario más un 10%. El vehículo se entrega con el tanque de combustible lleno y al devolverse la empresa cobra por el combustible faltante. Si el cliente se toma el trabajo de devolver el auto con el tanque "casi" lleno, este cargo será de 0.

El programa ya cuenta con una serie de funciones programadas que debemos utilizar. Se tienen las siguientes funciones:

- ✓ auto(alquiler) que recibe como parámetro un alquiler y devuelve el auto correspondiente a ese alquiler.
- ✓ Una función que categoriza las unidades en alquiler según la clase de automóvil y el año de fabricación categoriaUnidad(auto) devuelve un entero con 1, si es de categoría 1, 2 si es de categoría 2 y 3 si es de categoría 3.
- ✓ precioDiario(alquiler) devuelve un float que representa el precio diario que se paga por el alquiler.
- ✓ cantidadDias(alquiler) que recibe como parámetro un alquiler y determina el total de días alquilados.
- ✓ kmExcedidos(alquiler) que recibe como parámetro un alquiler y devuelve un entero que representa los kilómetros excedidos.
- ✓ valorKmExcedido(alquiler) y devuelve un float con el valor que se cobra por cada km excedido dependiendo de la categoría.
- ✓ ilimitado(alquiler) que recibe como parámetro un alquiler y devuelve True si el plan, en cuyo caso el km excedido se facturará como $23 por kilómetro excedido.
- ✓ tanque(auto) que devuelve un float que indica la cantidad de dinero que se cobrará por la falta de combustible en el tanque de ese auto.
- ✓ resgistrar(alquiler, importe) que recibe como parámetro un alquiler y el importe que se debe pagar por ese alquiler y lo registra en un archivo con fines estadísticos los datos del alquiler.

Se pide entonces que programe la función costo(alquiler) que como parámetro recibe un alquiler y que como resultado devuelve al programa llamador el importe a pagar por dicho alquiler (un float). La función también registra el alquiler y este importe en el archivo de estadísticas pero sólo si corresponde a un auto de categoría 2.

El importe se calcula teniendo en cuenta el valor diario de alquiler según la categoría a la que corresponde el auto alquilado, el kilometraje excedido por el valor que se cobra por el mismo para cada categoría de auto si es que corresponde, (kilómetros excedidos sólo se cobra a los alquileres que no son ilimitados). Los alquileres ilimitados, agregan en lugar del kilometraje excedido un 10% del costo por días de alquiler. Y todos los alquileres agregan al importe el dinero que indique devuelva la función tanque.

La signatura de la función será:

```
def costo(alquiler):
… cuerpo de la función
    return (importe)
```

Ejercicio 7:

Se desea automatizar el cálculo de la tarifa telefónica para una lista de clientes, la empresa informa que cada *llamado* tiene un costo por conexión más un costo por el tiempo consumido en segundos. Se cuenta con las siguientes funciones ya implementadas.

- ✓ obtenerCantidadLlamados(nroCliente) retorna la cantidad de llamados para un cliente.
- ✓ obtenerTiempoPorLlamada(nroCliente, nroLlamada) retorna la cantidad de segundos de un llamado de un cliente.
- ✓ obtenerCostoPorLlamada() retorna el costo fijo por cada llamada.

- ✓ obtenerCostoPorTiempo(segundos) dada la cantidad de segundos retorna el costo.

Realizar un programa que indique el monto de la factura para cada cliente.

Además, el sistema registra los clientes en una base para promociones pero con el siguiente comportamiento. Los clientes que ya estaban en la base, se los da de baja para evitar volver a ofrecer 2 veces seguidas la promoción. Si el tiempo total de comunicaciones para un cliente supera las dos horas, se lo agrega en la base para ofrecerle la promoción, salvo los dados de baja previamente. Para ello se cuenta con las siguientes funciones ya implementadas.

- ✓ esta(nroCliente) que indica si el cliente actualmente en la base de promociones.
- ✓ alta(nroCliente) que agrega el cliente en la base.
- ✓ baja(nroCliente) que da de baja el cliente en el registro de promociones.

Ejercicio 8:

Wall-E, el famoso robot de la película, necesita que le hagamos un favor. Wall-E es capaz de determinar su ubicación mediante la ayuda de antenas desplegadas por diferentes zonas. Lo que hace es enviar una señal, y en base a la distancia a las antenas, logra determinar su ubicación. Si por cualquier motivo no logra recibir respuesta a su señal desde por lo menos 3 antenas, entonces Wall-E suspende todas sus actividades y se pone a dormir, puesto que no es capaz de determinar su ubicación y ello es peligroso para él. Dado que una tormenta de arena afectó sus circuitos, necesita que la siguiente función sea definida nuevamente por los alumnos de IP.

Wall-E cuenta con las siguientes funciones ya implementadas:

- ✓ dameNrosAntenasCercanas(): devuelve una lista de los números de antenas en las cercanías.
- ✓ distancia(nroAntena): devuelve la distancia en metros a la antena número nroAntena.
- ✓ determinarUbicacion(nrosAntenas, distancias): toma dos listas, la de los números de antena y la de sus respectivas distancias a Wall-E y determina la ubicación del robot. No devuelve nada, pero es fundamental para el correcto funcionamiento del sistema.
- ✓ dormir(): Detiene las actividades del robot, poniéndolo en un estado de reposo. No devuelve nada.
- ✓ despertar(): Retoma las actividades del robot. No devuelve nada.
- ✓ estoyDormido(): Devuelve verdadero cuando el robot está dormido, y falso en caso contrario.
- ✓ Definir una función llamada controlDeUbicacion() que utilice las funciones listadas arriba. De haber 3 o más antenas en las cercanías, la función debe averiguar las distancias a todas las antenas cercanas, y determinar la posición del robot, de lo contrario, dormirlo. Esta tarea se debe repetir indefinidamente. Es posible que antenas en desuso se pongan a funcionar y sea entonces posible determinar la ubicación de Wall-E luego de estar dormido un tiempo, por lo que la función deberá despertarlo en esa situación.

La perrera Municipal de San Miguel rescata perros de la calle y los guarda hasta encontrarles un mejor destino. Cada perro se guarda en una jaula individual, hay 50 jaulas ocupadas al menos.

La municipalidad invirtió en tecnología de punta para la perrera, por lo que dispone de un sistema de alimentación automático.

El encargado quiere que cada vez que se active el sistema de alimentación, se haga lo siguiente:

- se chequee que el animal haya comido y bebido algo desde la última vez que se activó el sistema, de no ser así, que se alerte al encargado de dicha situación preocupante.
- se rellene el plato de alimento hasta llegar a 500 gramos.
- se rellene el plato de agua hasta llegar a 5 litros.

El sistema provee la siguientes funciones:

- ✓ obtenerPesoAlimento(nroJaula) devuelve el peso de alimento en el plato medido en gramos.
- ✓ obtenerLitrosAgua(nroJaula) devuelve la cantidad de litros de agua en el plato.
- ✓ volcarAlimento(nroJaula, cantGramos) pone cantGramos de alimento balanceado en la jaula nroJaula
- ✓ agregarAgua(nroJaula, cantLitros) pone cantLitros de agua en la jaula nroJaula.
- ✓ alertarEncargado(nroJaula) manda un alerta al celular del encargado avisándole que el perro de la jaula *nroJaula* tiene poco apetito.

Se necesita escribir la función principal para hacer que el sistema funcione como lo desea el encargado para todas las jaulas.

Ejercicio 9:

Una fábrica de aviones de papel y otros materiales, recibe pedidos que serán realizados por sus empleados. En cada pedido se encuentran las especificaciones del avión encargado. Entre otras cosas, cada avión tiene los siguientes atributos, relevantes para nuestro caso, tipo de material con el que debe ser realizado, modelo de avión, gama de color, y el precio que se le pasó por este pedido al cliente.

Deberá programar una función con el siguiente encabezado:

def fabricarAviones(pedidos, unEmpleado)

donde **pedidos** es una lista de pedidos de aviones para fabricar y **unEmpleado** es un código que identifica al empleado encargado de realizar esa lista de pedidos.

La función debe tomar **cada pedido de la lista**, **fabricar** el avión. Si la fabricación del avión pudo realizarse, **acumular el precio** de ese avión a un **total** por los trabajos realizados, qeu al finalizar todos los pedidos, debe registrarse. Los pedidos que no hayan podido realizarse deberán agregarse a una **lista de imposibles**.

La función deberá registrar el total de dinero que se recibirá por la acumulación del dinero que cada cliente pagará al recibir su pedido que fue realizado por este empleado.

La función **fabricarAviones** devolverá como salida la **lista de pedidos imposibles**.

- ✓ Se cuenta con las siguiente funciones:
- ✓ tipoDeMaterial(de un pedido): devuelve el material que solicita el cliente.
- ✓ tipoDeAvion(de un pedido): devuelve el modelo de avión que solicita el cliente.
- ✓ gamaDeColor(de un pedido): devuelve el color que solicita el cliente.
- ✓ hacerAvion(material, modelo, color): es una funión que lanza la fabricación del avión. Devuelve True si el avión pudo construirse con éxito y False, si el pedido no pudo concretarse.
- ✓ precio(de un pedido): devuelve un float que representa el precio que por ese avión que se le cobrará al cliente.
- ✓ registar**(float, un empleado):** registra el dinero total (float) que representa el haber finalizado con éxito los pedidos.