



El futuro digital
es de todos

MinTIC

«Misión TIC 2022»

RETO DE LA SEMANA 2

Los primeros “trabajadores” del conocimiento son los desarrolladores de software.



Universidad de Caldas



Reto : Semana 4



Mundo ASCII : Parte 2

Información para el formador

Este es un reto de ejemplo planteado para la semana 4, mi estrategia para los retos es desarrollar un gran proyecto en 5 retos. Ustedes pueden plantear la estrategia que consideren la más adecuada para sus grupos.

Temas que abarca este reto:

- IDEAL
- Expresiones lógicas
- Ciclos
- Listas
- Definición de Funciones
- Parámetros y Argumentos
- Módulos de usuario
- Comentarios
- Invocación/llamado de funciones (propias y de terceros)
- Cadenas de caracteres
- Composición de funciones
- Entrada y salida por consola
- Validación de la entrada de datos



El futuro digital
es de todos

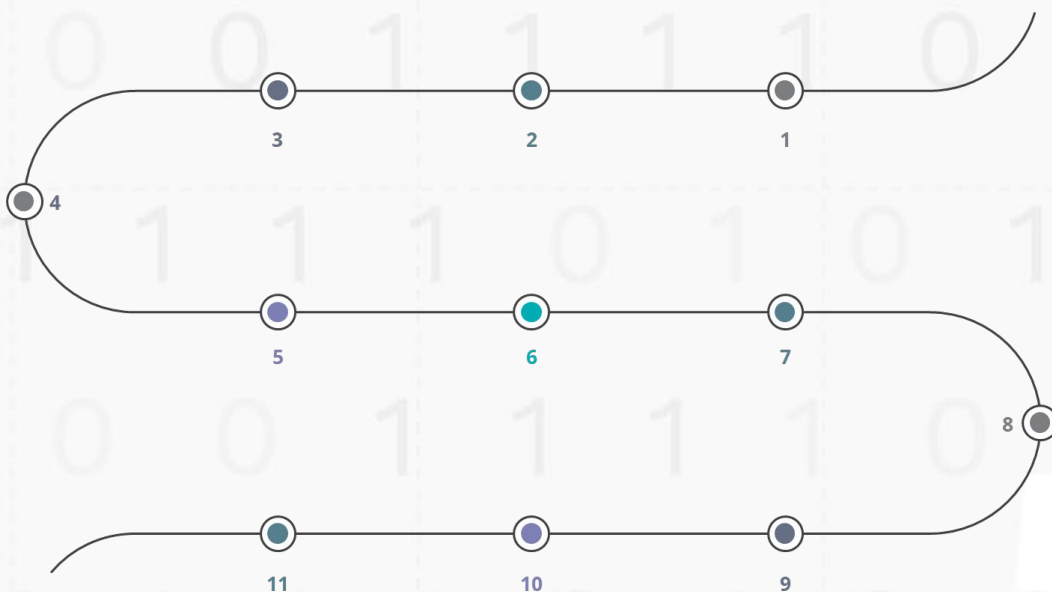
MinTIC

«Misión
TIC 2022»

ASCII WORLD

En el ASCII world están un poco mas tranquilos ya que aunque saben que no pueden bajar la guardia, la asignación de la etapa de vacunación fue todo un éxito gracias a la aplicación que implementamos.

La ciudad de los *humasciianos* tiene una distribución muy simple, viven distribuidos en una sola calle que está numerada desde -1000 hasta 1000 y solo se pueden transportar por esa calle que va en ambas direcciones.



Universidad de Caldas



A continuación se presenta el segundo reto:

Que debes hacer

1. Aplicar el proceso IDEAL completamente, es decir.
 - a. Identificar el problema
 - b. Definir el problema
 - c. Estrategias que dividan el problema
 - d. Algoritmos condicionales
 - e. Logros
2. Implementar la aplicación en Python
 - a. Utilizando instrucciones condicionales
 - b. Utilizando ciclos while-for
 - c. Utilizando funciones para cadenas de caracteres(str)
 - d. Utilizando funciones para listas
 - e. Definiendo funciones con parámetros
 - f. Invocando funciones correctamente
 - g. Documentando el código
 - h. Probando la aplicación
 - i. Invocando funciones de terceros

Primera parte IDEA

En esta primera parte te presentamos el contexto del problema que debes solucionar, debes hacer un documento en donde desarrolles, de manera similar a los laboratorios y talleres de la semana 1, los primeros 4 pasos del método IDEAL.

Reto2 : Asignación del Centro de Vacunación SCII CVA

Este reto se trata de implementar un programa que asigne el CVA respectivo a cada *humasciano* de acuerdo con su distancia y el día y la hora de la vacuna.



Asignación de CVA

Esta función tiene como objetivo asignar un CVA a los habitantes del planeta ASCII, para ello se ingresa una lista con la dirección de cada *humasciiano* en cada posición. Las direcciones de los *humasciianos* son un número entre -1000 y 1000 con la letra I si es a la izquierda de la calle y D si esta a la derecha de la calle.

Ejemplo

```
Lista_direcciones=["10-D", "-65-I", "-30-I", "900-D", "65-I"]
```

El gobierno del planeta ASCII ha construido 10 CVA a lo largo de la única calle de la ciudad, las direcciones de cada CVA, que solo es un número entero entre -1000 y 1000 sin letras ya que cada CVA ocupa ambos lados de la calle.

Se debe implementar una función que permita:

- Recibir como parámetros de entrada la lista de direcciones de los *humasciianos* y la lista de direcciones de los 10 CVA
- Retornar una lista para cada CVA con todas las direcciones de las personas que debe vacunar, el criterio de asignación es el CVA mas cercano a su dirección.

Citas de vacunación

En cada CVA por cada vacuna se demoran 10 minutos, y necesitan una aplicación que asigne la cita a cada uno de los pacientes asignados al CVA teniendo en cuenta las siguientes restricciones:

- Los días del planeta ASCII solo duran 5 horas es decir 300 minutos.
- La vacunación inicia el día 1 a las 0:0 y se asignan cada 20 minutos
- Cuando se llega a la 15ª cita del día se pasa al siguiente día.

Se debe implementar una función que permita:

- Recibir como parámetro de entrada la lista de direcciones de los *humasciianos* asignados a CVA
- Retornar una lista con el día y hora de la cita correspondiente a cada dirección de la lista ingresada



Ejemplo

Si entra

```
Lista_direcciones=["10-D", "-65-I", "-30-I", "900-D", "65-I", "15-D", "89-I", "35-I", "9-D", "650-I", "-900-D", "5-I", "10-D", "-65-I", "-30-I", "900-D", "65-I"]
```

Sale

```
Lista_citas=["1-0:0", "1-0:20", "1-0:40", "1-1:0", "1-1:20", "1-1:40", "1-2:0", "1-2:20", "1-2:40", "1-3:0", "1-3:20", "1-3:40", "1-4:0", "1-4:20", "1-4:40", "2-0:0", "2-0:20"]
```

A solucionar el problema

Plantea una solución a cada función del reto aplicando las 4 primeras actividades del método ideal, utiliza lo que necesites: dibujos, investiga fórmulas e Google, busca opciones de solución, plantea estrategias, escribe algoritmos y especifica requisitos!

Te sugiero que más que seguir el método a ciegas, sácale provecho a lo que te aporta cada etapa. Si te es más fácil hacerlo con papel y lápiz o en un tablero en tu casa (o pared, vidrio, etc.) muchísimo mejor; luego le tomas fotos a la solución y las pegas a un documento en Word.

A programar en Python

Recuerda que este reto debe hacerse en Repl.it, con tu cuenta de Gmail. Tu profesor formador te dará un lugar en donde podrás escribir tu código y lo más importante. ¡Probar si quedo bien!, (no desde el punto de vista sintáctico, esos errores te los informara Repl.it y podrás solucionarlos o pedir ayuda). Probaremos que los resultados obtenidos sean los esperados.



Que entregar

1. Un documento con el resultado de aplicar IDEA: este debes subirlo al enlace que se te habilitará en el aula virtual de Moodle.
2. Un programa en Python (L), que solucione el reto acorde con lo entregado en el punto 1.