



503202/503203 Programación Laboratorio de Programación Intermedio

EQUIPO PROGRAMACIÓN

6 de junio de 2025

- 1.- Programa que debe leer una lista de N números enteros, luego, en una función debe construir y devolver otra lista con los números impares mayores que un valor p dado por el usuario. Este valor p debe ser mayor o igual al número menor almacenado en la lista y debe ser menor o igual al número mayor almacenado la lista, ambas condiciones deben ser verificadas. No use las funciones provistas por Python para obtener los valores mayor y menor de una lista, sino defínalas Ud. mismo.

Entradas: El algoritmo tiene varias entradas. La primera corresponde a un número entero N ($5 < N \leq 100$), lo cual debe verificarse y de no cumplirse la condición se debe realizar un nuevo intento de ingreso hasta que se ingrese correctamente. Luego vienen N números enteros cuyos valores no tienen restricción, estos deben almacenarse en una lista L . Por último, se debe ingresar un valor p cuyo valor debe cumplir la restricción $L_{min} \leq p \leq L_{max}$, donde L_{min} y L_{max} representan los valores menor y mayor de la lista L , respectivamente.

Salida: La única salida del programa es una lista S con los números impares de L mayores al valor p .

Ejemplo de entrada: $N = 10$, $L = [-1, 3, 5, 11, 4, 22, 35, -7, 44, 139]$ y $p = 5$

Ejemplo de salida: $S = [11, 35, 139]$

- 2.- ¿Cuántas palabras puedes formar con las letras de esa patente?. Una inocente pregunta que, en un viaje en auto, desafía a los más pequeños a leer la patente de otro automóvil y con sus letras, tratar de mencionar todas las palabras que se le ocurran. Amanda es una intrépida programadora que ya descubrió que puede escribir un programa en Python que –primero– ingrese un diccionario con todas las palabras que se sepa. Luego en una segunda etapa, ingresará las 4 letras de una patente y este le entregará todas las palabras del diccionario que se puedan escribir con ellas.

Las letras de la patente no incluyen las vocales.

Entrada: La entrada al programa se divide en dos partes. En la primera parte, Amanda ingresará al programa el diccionario con las palabras que tengan al menos 4 consonantes. Este es un requisito que, si no se cumple, la palabra es descartada. El ingreso del diccionario termina cuando se ingresa la palabra vacía (se presiona ENTER). Luego, en la segunda parte de la ejecución del programa, Amanda ingresará las 4 letras de una patente.

Salida: La salida del programa estará compuesta por el listado de palabras del diccionario que contengan las 4 letras de patente ingresada por Amanda.

Ejemplo de entrada:

CONSTRUCCION	ESPECTACULO	ESTRUCTURA	FRUSTRACION	PROBLEMA	TRAPOS	OBSERVACION
RAPTOS	INSTRUCCION	ESPECIAL	PORTAS	FESTIVAL	RESPLANDOR	
HOSPITAL	INVESTIGACION	LIMITACION	OBLIGACION	PARTOS	TRANSPORTE	

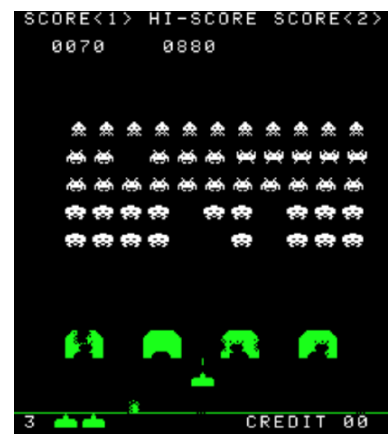
TRSP

Ejemplo de salida:

RAPTOS
PORTAS
PARTOS
TRAPOS
TRANSPORTE

3.- *Space Invaders* fue uno de los primeros shooters (videojuegos de disparos) de la historia. Publicado en 1978 como máquina de arcade, su creación requirió el diseño de hardware específico para soportar las necesidades computacionales y gráficas del juego.

Además de la propuesta jugable, una de las novedades que incorporó fue la tabla de mejores puntuaciones, en la que los jugadores podían registrar su nombre si conseguían los puntajes más altos del juego. Debido a las restricciones del hardware de la época, por “nombre” se entendía únicamente al registro de tres letras, de modo que los jugadores tenían que conformarse con poner sus apodos o sus iniciales o simplemente alguna combinación de tres letras que los identificara. Solo los jugadores más experimentados podían dejar mensajes más largos si conseguían todas las mejores puntuaciones de la máquina. Para eso, escribían tres letras de cada mejor puntaje obtenido de modo que si se leían todas en vertical aparecía un mensaje significativo.



Entradas: La entrada al programa se inicia con un número que indica cuántas mejores puntuaciones almacena una determinada máquina de arcade (al menos 1 y no más de 20). A continuación aparecen, por separado, los nombres de tres letras asociados a cada uno de los mejores puntajes.

Salidas: La única salida del programa es el mensaje que se puede deducir de la tabla de records.

Ejemplo de entrada: 5

S.D
PIE
ANR
CVS
EA!

Ejemplo de entrada: SPACE.INVADERS!

4.- *RPO (Retorno al Punto de Origen)* es una función automática que poseen los drones y que les permite volver al punto desde el que despegó cuando la batería está con poca carga.

En esta ocasión simularemos con un programa en Python el funcionamiento de la función RPO, ingresando las coordenadas del recorrido del drone y la capacidad de la batería cuando la nave despegue. Asumiremos que la batería se agota un 1 % por cada 50 metros recorridos y también supondremos que el drone regresa o por el mismo camino que hizo de ida (Caso 1) o yendo a la coordenada más cercana en el camino recorrido de ida y desde ahí de vuelta al punto de origen (Caso 2), lo que sea más corto. Por tanto, será responsabilidad del programa devolver el drone antes de que la batería se agote a la mitad de su capacidad de inicio.

Las coordenadas del recorrido estarán expresadas en tres dimensiones (x_i, y_i, z_i) y para efectos del cálculo de la distancia de este punto al punto (x_j, y_j, z_j) utilizaremos la fórmula:

$$D = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}.$$

Entrada: La entrada estará compuesta por varios números, el primero un valor entre 1 y 100 que corresponde al porcentaje de carga de batería del drone. A continuación viene una cantidad indeterminada de conjuntos de 3 valores correspondientes a las coordenadas del recorrido del drone (en metros). El ingreso termina cuando el programa detecta que el drone debe volver o cuando se ingresa las coordenadas $(0, 0, 0)$.

Salida: En cualquier caso, es decir, cuando el programa detecta que el drone debe volver o cuando se ingresa las coordenadas de retorno $(0, 0, 0)$, el programa debe indicar el porcentaje de batería final al aterrizar el drone en su punto de origen.

Ejemplo de entrada 1: 56

```
10 10 0
10 10 100
100 100 100
500 500 100
1250 1250 100
```

Ejemplo de salida 1: 18.9275

Ejemplo de entrada 2: 15

```
10 10 0
10 10 60
25 25 60
78 78 60
26 28 60
0 0 0
```

Ejemplo de salida 2: 10.434

