503202/503203 Programación
Programación usando Listas

EQUIPO PROGRAMACIÓN

25 de mayo de 2025

1.- **Lista de números** Construya un programa en Python que lea una lista de m números enteros ($0 < m \leq 20$). Luego el programa debe determinar y desplegar el valor mayor, el valor menor y la respectiva posición de ambos datos en la lista. Finalmente, el programa debe desplegar la lista de números pero ordenada.

Entradas: La entrada al programa estará compuesta de un primer valor entero positivo correspondiente al valor de m. A continuación vendrán m valores enteros.

Salidas: Este programa tendrá como salida 4 números enteros `men`, `may`, `pmen` y `pmay` correspondientes al valor menor, valor mayor, posición del valor menor y posición del valor mayor, respectivamente. A continuación vendrán los m números de la lista pero ordenados de menor a mayor.

Ejemplo de entrada: `7, 4, 9, 17, -4, -2, 8` y `12`

Ejemplo de salida:

```
menor=-4 en posición=4
mayor=17 en posición=3
-4  -2  4  7  8  9  12  17
```

(notar que la primera posición cuenta como 1).

2.- **Romanos** Construya un programa Python al que se ingresa n strings, los cuales representan números en notación romana. Su programa debe transformar y desplegar estos números romanos en números en notación decimal. Tenga en cuenta que los valores de los símbolos en notación romana y sus equivalencias en decimal son las siguientes:

símbolo	valor
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Entradas: La entrada está compuesta por un número entero n ($n > 0$), seguido por n líneas, cada una con un string representando un número en notación romana. Se debe validar que cada string está compuesto sólo de los símbolos en la tabla.

Salidas: La salida estará compuesta de n números, cada uno correspondiente al equivalente decimal de su respectivo número romano en la entrada.

Ejemplo de entradas:

```
3
MMMDCVII
CMXLIII
CDXLIV
```

Ejemplo de Salidas:

```
3607
943
444
```

3.- **Moda** Construya un programa en Python que lea una lista de $n(3 < n \leq 10000)$ números enteros no negativos. Luego el programa debe determinar la moda del conjunto de datos (e.d., el valor que más se repite en el conjunto). Considere que los datos no están necesariamente ordenados.

Entradas: La entrada al programa estará compuesta de un primer valor entero positivo correspondiente al valor de n. A continuación vendrán n valores enteros.

Salidas: Este programa tendrá como salida 2 números enteros `moda` y `rmoda` correspondientes a la moda y la cantidad de veces que se repite la moda, respectivamente.

Ejemplo de entrada: `15 4, 0, 0, 9, 17, 4, 2, 6, 4, 4, 0, 4, 6, 8` y `17` (n=15 y luego los 15 valores)

Ejemplo de salida: `moda=4` y se repite `5` veces

4.- **Transformación de palabras** Shakespeare tenía un extraño poder consistente en escribir una palabra con su pluma, la cual se iba transformando en otra palabra sólo con cambiar una letra en un orden misterioso.

Se sabe que las reglas para transformar una palabra en otra son:

a) En cada paso, sólo se puede cambiar una letra de la palabra anterior

b) Cada letra se puede cambiar sólo una vez

c) Todas las letras en una palabra deben ser cambiadas

Entradas: En la entrada habrán varias series de palabras transformadas desordenadas. Para cada serie de palabras habrá una línea con dos números, la cantidad de palabras q ($q > 0$) de la serie y la cantidad de letras l ($l > 0$) de cada palabra en la serie. A continuación la primera palabra de la serie (que se mantendrá como primera en la respuesta) y luego el resto de las palabras transformadas pero desordenadas. Las palabras puede venir en minúsculas y/o mayúsculas, pero se puede suponer que están compuestas sólo de letras.

Salidas: Para cada serie, se debe desplegar las palabras ordenadas correctamente.

Ejemplo de entrada:

```
6 5
remar
pitos
remas
remos
retos
ritos
```

```
5 4
pato
lisa
pata
pita
pisa
```

Ejemplo de salida:

```
remar
remas                                   pato
remos                                   pata
retos                                   pita
ritos                                   pisa
pitos                                   lisa
```

5.- **Corrector ortográfico** Algunos procesadores de texto tienen una función para la corrección de palabras mal escritas, esto es, un *Corrector Ortográfico Automático (COA)*. El tipo de correcciones realizado por un COA se puede clasificar en tres categorías:

a) Letra ausente (p.e., cuando se escribe **ación** en vez de **acción**) o también letra duplicada (p.e., cuando se escribe **accción**).

b) Letra errónea (p.e., cuando se escribe **axión** en vez de **acción**).

c) Intercambio erróneo de letras adyacentes (p.e., **accóin** en vez de **acción**)

Los COA están basados en un diccionario de palabras conocidas. Cuando un texto contiene una palabra que no está en el diccionario el COA intenta reemplazar esta palabra por una palabra similar en el diccionario. Dos palabras son similares si podemos transformar una palabra en otra aplicando exactamente una de las correciones listadas anteriormente. Una palabra desconocida es dejada sin cambios si no hay palabras simmilares en el diccionario.

Entradas: La primera línea de la entrada contendrá la cantidad de palabras en el diccionario ($n, n \leq 10000$). Las siguientes n líneas contendrán, cada una, una palabra del diccionario. La siguiente línea contendrá la cantidad de palabras por las que se hará una consulta ($q, q \leq 1000$) y a continuación las q palabras. Cada palabra ingresada debe tener entre 1 y 25 letras ("a" a la "z").

Salidas: Por cada una de las q palabras consultadas se debe desplegar la propia palabra y una de las siguientes opciones:

a) `es una palabra correcta`, si la palabra está en el diccionario

b) `es una palabra similar a <x>`, donde `<x>` es la palabra del diccionario más similar.

c) `es una palabra desconocida`, si no aplica ninguno de los casos anteriores.

Ejemplo de entrada:

```
10
este
es
un
diccionario
que
usaremos
para
corregir
palabras
erróneas
6
nu
dicionario
```

```
palabras
nueva
korregir
esste
```

Ejemplo de salida:

```
nu es una palabra similar a un
dicionario es una palabra similar a diccionario
palabras es una palabra correcta
nueva es una palabra desconocida
korregir es una palabra similar a corregir
esste es una palabra similar a este
```

6.- **Código de Da Vinci** El Código de da Vinci es un libro escrito por Dan Brown, popularizado por la exitosa película del mismo nombre protagonizada por Tom Hanks. En este libro (película) se exhibe una interesante técnica de encriptación (códificación) para ocultar mensajes secretos, la cual usa los números de la serie de Fibonacci para cifrar el texto. Su tarea será escribir un programa en Python que permita leer un mensaje cifrado, que lo decodifique y despliegue el mensaje oculto.

La técnica se describirá usando un ejemplo. Cualquier texto cifrado consistirá de un conjunto de números (todos pertenecientes a la Serie de Fibonacci) y un mensaje oculto como el siguiente:

```
233 144 89 55 1 21 5 8 2 13
ASI, LLAMO "AN"!
```

cuya salida sería:

```
LA MONA LISA
```

Para este problema se debe suponer que los dos primeros números de la Serie de Fibonacci son el 1 y el 2. Los siguientes términos de la serie se obtienen sumando los dos términos previos, logrando la secuencia 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Entonces, ¿cómo se obtiene "`LA MONA LISA`" desde el string "`ASI, LLAMO AN!`"?. Acá es donde entran en juego los números de Fibonnaci de la primera línea. El primero es el 233, el cual es el 12vo término de la Serie de Fibonacci. Luego, la primera letra A debe ir ubicada en la 12va posición del texto descifrado. El segundo número es 144 que corresponde al 11vo término de la Serie de Fibonacci, por tanto, la letra S va en la 11va posición del texto descifrado. Luego viene el 89 que es el décimo término de la Serie de Fibonacci, por tanto la I va en la 10ma posición del texto descifrado. Este proceso continúa hasta que se completa el descifrado del mensaje. Notar que sólo las letras se consideran, los otros símbolos son simplemente distractores.

Si falta un número de Fibonacci en la secuencia de entrada entonces, se debe dejar un espacio en la respectiva posición del término faltante. En el ejemplo, faltan los términos 3 y 34, por tanto, se insertan espacios en la tercera y octava posición del mensaje cifrado.

Entradas: Está compuesta por un valor T correspondiente a los casos de prueba. Cada caso de prueba consiste en tres líneas. La primera línea contiene un entero positivo N seguido de N números pertenecientes a la Serie de Fibonacci. Los números estarán separados por un espacio. Finalmente, la tercera línea contiene el texto cifrado.

Salidas: Para cada caso de prueba, la salida consiste en el respectivo texto descifrado.

Ejemplo de entrada:

```
2
10
233 144 89 55 1 21 5 8 2 13
ASI, LLAMO "AN"!
15
34 21 13 144 1597 3 987 610 8 5 89 2 377 2584 1
O, DRACONIAN DEVIL!
```

Ejemplo de salida:

```
THE MONA LISA
LEONARDO DA VINCI
```