



Primeros programas 2

Problema 1

(2,5 puntos) Posibles

Escribe una función iterativa **iterPower** (**base**, **exp**) que calcula el exponencial base^{exp} simplemente usando multiplicación sucesiva.

Por ejemplo, `iterPower (base, exp)` debe calcular base^{exp} multiplicando `base` por sí mismo `exp` veces.

Esta función debe tomar dos valores - `base` puede ser un float o un número entero; `exp` será un número entero ≥ 0 . Debe devolver un valor numérico. El código debe ser iterativo - el uso del operador `**` no está permitido.

```
def iterPower(base, exp):
    ...

    base: int or float.
    exp: int >= 0

    returns: int or float, base^exp
    ...

    # Your code here
```

Problema 2

(2,5 puntos posibles)

El máximo común divisor de dos números enteros positivos es el número entero más grande que divide cada uno de ellos sin resto. Por ejemplo,

`gcd (2, 12) = 2`

`gcd (6, 12) = 6`

`gcd (9, 12) = 3`

`gcd (17, 12) = 1`

Escribe una función iterativa, **gcdIter** (**a**, **b**), que implementa esta idea. Una forma fácil de hacer esto es comenzar con un valor de prueba igual al menor de los dos argumentos de entrada, y de forma iterativa reducir este valor de la prueba en 1 hasta llegar al caso en el que la prueba divide a y b sin resto, o llegar a 1.

Problema 3

(2,5 puntos posibles)

Tiempo estimado para terminar: 15 minutos

Supongamos `s` es una cadena de caracteres en minúscula.

Escribir un programa que imprime el número de veces que la cadena 'bob' se produce en `s`.

Por ejemplo, si `s='azcbobobegghakl'`, entonces su programa debe imprimir

Número de veces que ocurre es: 2

Problema 4

(2,5 Puntos Posibles)

Una empresa de catering que ha contratado para ayudar con la organización y preparación de los pedidos del cliente. Se le da una lista de los artículos deseados de cada cliente, y debe escribir un programa que va a contar el número de cada uno de los artículos necesarios para los cocineros para preparar. Los artículos que un cliente puede pedir son: ensalada, hamburguesa, y agua.

Escribir una función llamada `item_order` que toma como entrada una cadena denominada `orden`. La cadena contiene sólo palabras de los artículos que el cliente puede pedir separadas por un espacio. La función devuelve una cadena que cuenta el número de cada elemento y las consolida en el siguiente orden:

`ensalada: [# ensalada] hamburguesa: [# hambruger] agua: [# agua]`

Si un pedido no contiene un elemento, entonces el recuento de ese elemento es 0. Tenga en cuenta que cada elemento tiene el formato de `[nombre del elemento] [un símbolo de :] [recuento de Artículo]` y todos los grupos de artículos están separadas por una espacio.

Por ejemplo:

Si `orden = "ensalada hamburguesa agua ensalada hamburguesa"`, entonces la función devuelve `"ensalada: 2 hamburguesa: 2 agua: 1"`

Si `orden = "hamburguesa hamburguesa agua"`, entonces la función devuelve `"ensalada: 0 hamburguesa: 2 agua: 1"`