

---

**Cifrado con contraseña numérica****Z63415\_es**

---

Escribe una función llamada `cifrar_avanzado_numerico`. Dicha función recibirá dos parámetros de tipo cadena de caracteres. La función debe devolver una cadena de caracteres.

La función deberá cifrar la cadena de caracteres de entrada (primer parámetro) usando un mecanismo muy parecido al que se vio en prácticas para el problema “Cifrado con contraseña”. Cada carácter del mensaje de entrada se sustituirá por otro carácter que obtendremos al avanzar en el alfabeto un número determinado de posiciones. Si decidimos que vamos a adelantar dos posiciones, el carácter “a” se convertirá en “c”, el carácter “b” se convertirá en “d”, etc.

Para avanzar en el alfabeto, haremos servir las funciones `ord` y `chr` vistas en clase. La función `ord` recibe un parámetro de tipo carácter y devuelve su código Unicode, es decir, un número entero positivo que lo identifica. Dichos códigos siguen generalmente el orden alfabético. La función `chr` recibe un parámetro de tipo entero y devuelve el carácter asociado al código Unicode pasado como parámetro.

Para cifrar una cadena, por cada carácter de la cadena original, debemos obtener su código Unicode con `ord`, sumarle el incremento correspondiente, obtener el carácter cifrado con `chr` y añadirlo a la cadena cifrada.

El incremento a utilizar será se extraerá del segundo parámetro de la función. El segundo parámetro siempre será una cadena de caracteres que únicamente podrá contener caracteres numéricos, es decir, “0”, “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8” o “9”. Llamaremos a este parámetro “cadena de incrementos”. Sea `c0` el primer carácter de la cadena de incrementos, `c1` el segundo, etc. El incremento aplicado sobre el primer carácter de la cadena a cifrar debe ser el valor entero de `c0` obtenido con la función `int`. El incremento aplicado sobre el segundo carácter de la cadena a cifrar debe ser el valor entero de `c1`, y así sucesivamente. Como la cadena de avances puede ser más corta que la cadena a cifrar, cuando lleguemos al final de la misma, volveremos a empezar.

Por ejemplo, si ciframos la cadena “casa” pasamos y empleamos “786” como cadena de avances, el resultado será “jiyh” porque habremos incrementado la “c” en 7 posiciones, la “a” en 8 posiciones, la “s” en 6 posiciones y última “a” en 7 posiciones.

Para que tu función pueda ser evaluada correctamente por el juez en línea, tu código deberá tener la siguiente forma:

```
import sys

def cifrar_avanzado_numerico (cad, incrementos):
    ...

c=sys.stdin.readline().strip()
a=sys.stdin.readline().strip()
resultado=cifrar_avanzado_numerico(c,a)
print(resultado)
```

## **Entrada**

(Si utilizas el fragmento de código definido más arriba, no debes preocuparte por esto) La primera línea contendrá la cadena a cifrar y, la segunda, la cadena de avances a emplear.

## **Salida**

(Si utilizas el fragmento de código definido más arriba, no debes preocuparte por esto) La cadena cifrada.

## **Información del problema**

Autor : Juan Morales García

Generación : 2025-09-03 19:33:55

© *Jutge.org*, 2006–2025.

<https://jutge.org>