Base64 (2) P38550_es

En el problema Base64 (1), se explicaba como codificar un entero del 0 al 63 en un carácter `A-Z`, `a-z`, `0-9`, `+`, `/`. En este problema te pedimos que codifiques cadenas arbitrarias de caracteres. Para ello, deberás seguir el siguiente proceso:

- Agrupa los caracteres de la entrada de 3 en 3.
- Codifica en binario de cada uno de los 3 caracteres de cada grupo, dando lugar a $3 \cdot 8 = 24$ bits.
- Transforma cada grupo de 24 bits en 4 números de 6 bits cada uno (o sea, números del 0 al 63).
- Representa los 4 números en base 64, como en el problema anterior.

Por ejemplo, la codificación en binario de `ALA` es

01000001|01001100|010000001

que da lugar a los siguientes cuatro grupos de 6 bits,

010000|010100|1100010|000001

que se corresponde a los números

16|20|50|1

que se codifica como `QUyB`.

Si la entrada no tiene un número de caracteres que sea múltiplo de 3, se hace un tratamiento especial:

- Si sobra un grupo de 1 carácter, se añaden 4 bits 0, dando lugar a 12 bits, o sea, dos números de 6 bits. Se codifican ambos números, y al final de la codificación se añade == para indicar que *faltan* dos caracteres.
- Si sobra un grupo de 2 carácter, se añaden 2 bits 0, dando lugar a 18 bits, o sea, tres números de 6 bits. Se codifican los tres números, y al final de la codificación se añade = para indicar que *falta* un caracter.

Haz un programa que codifique y descodifique secuencias de caracteres en base 64.

Entrada

Un número arbitario de líneas, cada una de las cuales será de la forma `C XXX` o `D YYY`, donde `XXX` es una cadena arbitraria de caracteres que hay que codificar, y `YYY` es una codificación en base64 que hay que descodificar.

Salida

Para cada caso, escribe una línea con la codificación (`C`) o descodificación (`D`) según se pida. Se te garantiza que todos los caracteres que deberás codificar o decodificar no tienen ningún problema para leer o escribir (por ejemplo, no recibirás caracteres como el espacio, contrabarra, caracteres con valores menores a 32 o mayores a 126, etc.).

Puntuación

- **Test1:** **25 Puntos**

  Entradas donde únicamente se pide codificar textos con un número de caracteres múltiple de 3.

- **Test2:** **25 Puntos**

  Entradas donde únicamente se pide descodificar secuencias en base64 sin ningún caracter =, o sea, secuencias que dan lugar a textos con un número de caracteres múltiple de 3.

- **Test3:** **25 Puntos**

  Entradas donde se pide codificar textos de todo tipo.

- **Test4:** **25 Puntos**

  Entradas donde se pide descodificar secuencias en base64 de todo tipo.

Ejemplo de entrada 1

```
C ALA
C ALAALAALA
C 012345
C ABABAB
```

Ejemplo de salida 1

```
QUxB
QUxBQUxBQUxB
MDEyMzQ1
QUJBQkFC
```

Ejemplo de entrada 2

```
D QUxB
D QUxBQUxBQUxB
D MDEyMzQ1
D QUJBQkFC
D VGhpc0lzQVRleHRJbkJhc2U2NCEh
```

Ejemplo de salida 2

```
ALA
ALAALAALA
012345
ABABAB
ThisIsATextInBase64!!
```

Ejemplo de entrada 3

```
C HOLA_CARACOLA
C ThisIsATextInBase64
C ThisIsATextInBase64!
C Hi
```

Ejemplo de salida 3

```
SE9MQV9DQVJBQ09MQQ==
VGhpc0lzQVRleHRJbkJhc2U2NA==
VGhpc0lzQVRleHRJbkJhc2U2NCE=
SGk=
```

Ejemplo de entrada 4

```
D SE9MQV9DQVJBQ09MQQ==
D VGhpc0lzQVRleHRJbkJhc2U2NA==
D VGhpc0lzQVRleHRJbkJhc2U2NCE=
D SGk=
```

Ejemplo de salida 4

```
HOLA_CARACOLA
ThisIsATextInBase64
ThisIsATextInBase64!
Hi
```

Información del problema

Autoría: Omer Giménez

Generación: 2026-01-25T10:37:33.816Z