

---

## Decodificar Base64 (1)

Z32581\_es

---

### Preliminares

En base 2 tenemos 2 dígitos (0 y 1), en base 10 tenemos 10 (0, 1, 2, ..., 9), y en base 16 tenemos 16 (el hexadecimal, que utiliza 0, 1, 2, ..., 9, A, B, C, D, E, y F).

La **base 64** es pues, simplemente, una base con muchos más dígitos. Pero dado que necesitamos 64 símbolos diferentes para representar todos los dígitos, es necesario usar todas las letras del código ASCII (26 mayúsculas y 26 minúsculas, en total 52), y todos los dígitos decimales (0 a 9, en total 10). De hecho esto no es suficiente, porque nos da 62 símbolos en total. Para llegar a los 64, se eligieron el '+' para el dígito 62 y '/' para el 63.

Las correspondencias exactas entre dígitos en base 64 y sus caracteres respectivos son estas:

- Del 0 al 25, la correspondiente letra mayúscula,
- del 26 al 51, la correspondiente letra minúscula,
- del 52 al 61, el correspondiente dígito decimal,
- el 62 es el carácter '+'; y
- el 63 es el carácter '/'.

Así pues, si mostramos los caracteres ASCII utilizados en base 64 (que, por cierto, es un estándar), todos juntos y por orden, tenemos:

ABCDEFGHIJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/-

### Objetivo

Haz una **función** que, dado un carácter que representa un dígito en base 64, retorne el dígito como un entero (entre 0 y 63). La función tiene la cabecera:

```
/**  
 * @brief Pasa un carácter que representa un dígito en base 64 a su  
 * valor correspondiente  
 *  
 * @pre Precondición: c es una letra, un dígito decimal, o bien '+' o '/'  
 */  
int char_to_base64(char c);
```

### Observación

Este problema tiene como centros de interés la **corrección** y la **legibilidad**. Solo hay que enviar la función pedida, el programa principal será ignorado. No se pueden utilizar las funciones `isupper`, `islower` de la librería estándar.

## Ejemplo de entrada

A  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z  
a  
b  
c  
d  
e  
f  
g  
h  
i  
j  
k  
l  
m  
n  
o  
p  
q  
r  
s  
t  
u  
v  
w  
x  
y  
z  
0  
1  
2  
3  
4  
5  
6

7  
8  
9  
+  
/

<b>Ejemplo de salida</b>	
0	32
1	33
2	34
3	35
4	36
5	37
6	38
7	39
8	40
9	41
10	42
11	43
12	44
13	45
14	46
15	47
16	48
17	49
18	50
19	51
20	52
21	53
22	54
23	55
24	56
25	57
26	58
27	59
28	60
29	61
30	62
31	63

## Información del problema

Autor : Pau Fernández  
Generación : 2025-10-30 12:09:34

© *Jutge.org*, 2006–2025.  
<https://jutge.org>