#### char \*var = "Hola"

//Reserva zona de memoria para un puntero
// a un char llamado "var", reserva zona de memoria para la secuencia
// de caraceteres "Hola" e iguala var a esa zona de memoria

? ? ? ? ? ? ? ? H o I a \ 0 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?	7 2 2 2 2 2 2 2 2 2
8928 8930 8930 8931 8931 8931 8931 8931 8931 8931 8931	8968 8969 8970 8971 8973 8974 8975 8976 8976

# printf("%d",var ); // Se imprime el valor de la variable var : 8937 printf("%c", \*var); // Imprime como carácter el valor de aquello a lo que apunta var : printf("%c", \*var+1); // Imprime como carácter el valor de aquello a lo que apunta var // aumentado en 1. 'H' + 1 = 'I' // Modifica 'el carácter al que apunta var (y que está en la // posición 8937) por una 'A'. Sin embargo, esta operación // falla. Según el estándar C (Sección 6.4.5) el comportamiento // de una operación que intenta modificar un literal de cadena // es indefinido. printf("%c", var[0]); // Imprime el primer elemento de la matriz var, que es a su // vez la primera letra de la cadena : 'H' printf("%c", var[1]); // Imprime el segundo elemento de la matriz var, que es a su // vez la segunda letra de la cadena : 'o' printf("%c", 1[var]); // Equivalente a var[1]. No usar, salvo para impresionar

# printf("%s", var); // Imprime la

// Imprime la cadena a la que apunta var : "Hola"

## printf("%d", var+1);

// Imprime la dirección de memoria en la que iría el siguiente dato al que // apunta var. como var apunta a un char, y un char ocupa un byte, // la siguiente dirección de memroia sería la 8938.

# printf("%c", \*(var+1));

// Imprime el valor de aquello a lo que apunta la dirección var + 1. Dado // que var+1 es 8938, el carácter que hay en 8938 es 'o'  $\,$ 

#### printf("%s", var+1);

// Imprime la cadena a la que apunta var+1 (8938) : "ola"

# printf("%d",strlen(var));

// Imprime la longitud de la cadena a la que apunta var. Al ser esta cadena // "Hola", imprime 4  $\,$ 

# printf("%d",strlen(var+1));

// Imprime la longitud de la cadena a la que apunta var+1. Al ser esta cadena // "ola", imprime 3

# printf("%d",&var[1]);

// Imprime la dirección de memoria donde se encuentra la segunda letra // de la cadena : 8938

# printf("%c", &1[var]);

// Equivalente a &var[1]. No usar, salvo para impresionar. Imprime 8938