

1. Programe el procedimiento **erase** el cual borra una porción de una cadena. Recibe en **BX** un apuntador a una cadena terminada en 0, en **AL** la posición del primer carácter a borrar y en **CL** la cantidad de caracteres.

Si la cadena es más corta que los caracteres solicitados en **CL**, el procedimiento borra todos los posibles. Si la posición en **AL** es mayor que la longitud de la cadena, el procedimiento retorna un **-1** en **AH**, caso contrario retorna **0**.

Ejemplo:

```
mov bx, offset cadena      ; si la cadena es "Hola mundo"
mov al, 2                  ; borrar a partir de la posición 2
mov cl, 4                  ; borrar 4 caracteres
call erase                 ; cadena es "Houndo", retorna AH=0
```

2. Programe el procedimiento **substr** el cual almacena en una cadena una copia de una porción de otra cadena. Recibe en **SI** un apuntador a una cadena fuente, en **DI** un apuntador a la cadena destino, en **AL** la posición inicial a copiar y en **CL** la cantidad de caracteres.

Si la cadena es más corta que los caracteres solicitados en **CL**, el procedimiento copia todos los posibles. Si la posición en **AL** es mayor que la longitud de la cadena, el procedimiento retorna un **-1** en **AH**, caso contrario retorna **0**.

Ejemplo:

```
mov si, offset cadena      ; si la cadena es "Hola mundo"
mov di, offset destino
mov al, 1                  ; copiar a partir de la posición 1
mov cl, 5                  ; copiar 5 caracteres
call substr               ; destino es "ola m", retorna AH=0
```

3. Programe el procedimiento **strcmp** el cual compara dos cadenas lexicográficamente. Recibe en **SI** y **DI** dos apuntadores a cadenas terminadas en 0, y retorna en **AH** un **0** si son iguales, un valor **< 0** si la cadena en **SI** es menor que **DI**, o un valor **> 0** de lo contrario.

Ejemplo:

```
mov si, offset cadena1     ; si cadena1 es "Hola mundo"
mov di, offset cadena2     ; si cadena2 es "hola mundo"
call strcmp                ; retorna AH=-32 (0xE0)
```

4. Programe el procedimiento **strchr** el cual retorna la posición de un carácter en una cadena. Recibe en **BX** un apuntador a una cadena terminada en 0 y en **AL** el carácter a buscar. Retorna en **AH** la posición del carácter en la cadena (la primera que encontró) o un **-1** si no se encuentra en la cadena.

Ejemplo:

```
mov bx, offset cadena      ; si la cadena es "Hola mundo"
mov al, 'o'                ; buscar la letra 'o'
call strchr                ; retorna AH=1
```

5. Programe el procedimiento **strstr** el cual retorna la posición de una cadena en otra cadena. Recibe en **SI** y **DI** dos apuntadores a cadenas terminadas en 0. Retorna en **AH** la posición en que está la cadena en **DI** en la apuntada por **SI** (la primera que encontró) o un **-1** si no se encuentra en la cadena.

Ejemplo:

```
mov si, offset cadena1      ; si cadena1 es "Hola mundo"
mov di, offset cadena2      ; si cadena2 es "mundo"
call strstr                  ; retorna AH=5
```