

Examen Laboratorio Junio 2017

Fundamentos de Ordenadores y Sistemas Operativos

Mario Martínez Zarzuela
marmar@tel.uva.es

Enunciado

- Este segmento de datos está disponible en el archivo "data_1.asm"

```
1      .data
2  abc:  .ascii "abcdefghijklmnopqrstuvwxyz"
3  input: .byte  'f'
4  output: .word  -1
5  msj:   .ascii "\nPosicion: "
6  msj2:  .ascii "\nCaracter: "
7  input2: .byte  ' '
```

Enunciado

- Zonas de memoria:
 - **abc**: contiene una cadena de texto con los caracteres del abecedario inglés.
 - **input**: contiene un caracter almacenado como un BYTE.
 - **output**: servirá para almacenar el índice del caracter input en el abecedario como un WORD.
 - **msj/msj2**: cadenas de texto para imprimir en pantalla.
 - **input2**: espacio reservado para un nuevo caracter.

Enunciado

- Resolver:
 - **A) (3.0p)** Escribe un bucle que calcule el índice (posición) del caracter input en la cadena **abc**.
 - Considera que el caracter 'a' de la cadena ocupa la posición 0.
 - La posición debe almacenarse en memoria en la etiqueta **output**.
 - **B) (1.0p)** Introduce las modificaciones necesarias para que se imprima en pantalla el texto en **msj** y el valor almacenado en **output**.
 - **C) (2.0p)** Introduce las modificaciones necesarias para que el caracter almacenado en **input** se solicite por teclado.

Enunciado

- Salida esperada:

- Apartados A y B:

- Posicion: 5

- Apartado C:

- Caracter: h

- Posicion: 7

Enunciado

- Este segmento de datos está disponible en el archivo "data_2.asm"

```
1      .data
2  abc:  .ascii "abcdefghijklmnopqrstuvwxyz "
3  clave: .ascii "qdyvtkfuompciasrxznlwebgjh_"
4  input: .byte  'f'
5  output: .word  -1
6  msj:   .ascii "\nPosicion: "
7  input3: .ascii "texto a encriptar"
```

Enunciado

- Zonas de memoria:
 - **abc**: contiene una cadena de texto con los caracteres del abecedario inglés y un espacio.
 - **clave**: contiene una cadena de texto con los caracteres del abecedario inglés desordenados y un guión bajo.
 - **input**: contiene un caracter almacenado como un BYTE.
 - **output**: servirá para almacenar el índice del caracter input en el abecedario como un WORD.
 - **input3**: contiene un texto que será encriptado.

Enunciado

- Resolver:
 - **D) (2.0p)** Elimina el código que necesitaste para el apartado C y adapta el programa cambiando el código que elaboraste para el apartado A para escribir una función que reciba un caracter como argumento y devuelva una posición.
 - La función debe almacenar al menos el valor de un registro en la pila, ocupando el mínimo espacio necesario en memoria.
 - La salida del programa debe ser la misma que para el apartado B del enunciado anterior.

Enunciado

- Resolver:

- **E) (1.0p)** Escribe un bucle que recorra la cadena **input3** y codifique cada uno de sus caracteres utilizando el caracter indicado en la cadena **clave**.
 - El carácter 'a' (posición 0 en abc) se codifica con el carácter 'q' (posición 0 en clave).
 - Para conocer la posición de los caracteres en input3 llama a la función que escribiste en el apartado D.
- **F) (1.0p)** Imprime cada uno de los caracteres codificados por pantalla.
 - Usa la **llamada al sistema 11**: necesario el valor ASCII a imprimir en \$a0

Enunciado

- Salida esperada:

- **Apartado D:**

- `Posicion: 5`

- **Apartado F:**

- `ltgls_q_tayzorlqz`