

## Ejercicios sobre Control de flujo de programa

1. Programe los siguientes procedimientos usando las instrucciones de control de flujo del procesador 8088 y los procedimientos en la biblioteca de funciones **PCLIB06**. Escriba el código fuente en un editor de texto y almacénelo en un archivo llamado **P1.asm**, ensamble el código con **TASM** y encadénelo con **TLINK**. Verifique la correcta implementación de los procedimientos por medio de ejecutar el programa con distintos valores de prueba.

- a) **gets**: Almacena en memoria una cadena de caracteres ingresada por el usuario por medio del teclado. La captura termina cuando el usuario presiona la tecla **ENTER**. La cadena se almacena a partir de la dirección **DS:BX** y con un 0 (*crudo*, null) al final que indica el fin de cadena. Cuando el usuario teclea la cadena, se deben visualizar en pantalla los caracteres ingresados.
  - a. Modifique el procedimiento **gets** de forma que acepte la tecla **BACKSPACE**. Cada que el usuario presiona esta tecla, el último carácter es borrado de pantalla y de la cadena en memoria. El procedimiento no debe usar **gotoxy**, haga uso de **getch/getchar** y **putchar**.

*Información:* El código ASCII de la tecla **ENTER** es 13 (denominado Carriage Return, CR). El **BACKSPACE** es 8. Se hace uso de **getch/getchar** para recibir cada carácter individual del teclado y se verifica si se recibió un **ENTER** ó **BACKSPACE**, para proceder a terminar la captura o hacer el borrado del último carácter ingresado.

- b) **getsAlpha**: El procedimiento es similar a **gets**, a excepción de que sólo acepta caracteres del abecedario a – z y A – Z.
- c) **getsNum**: El procedimiento es similar a **gets**, a excepción de que sólo acepta caracteres numéricos 0 – 9.
- d) **toUpperCase**: Convierte a mayúscula todos los caracteres alfabéticos de una cadena terminada en null. La cadena está apuntada por **DS:BX**.
- e) **removeChar**: Elimina todas las ocurrencias del carácter indicado en **AL** de una cadena de caracteres terminada en null. La cadena está apuntada por **DS:BX**. Ejemplo:

*Cadena original:* Baja California.

*Carácter en AL:* a

*Cadena resultante:* Bj Cliforni.

2. Realice un cómic donde describa las visiones de la programación estructurada: la **visión clásica** sobre el control de flujo y la **visión moderna** sobre la división en módulos independientes, así como las ventajas de programar siguiendo las tres facetas de la programación estructurada: diseño descendente, programación modular y código estructurado (uso de secuencia, selección e iteración).

Se sugieren las siguientes herramientas para hacer el cómic:

<https://www.makebeliefscomix.com/>

<http://www.wittycomics.com/>

<https://www.pixton.com/>

<http://stripgenerator.com/>