Computer Science 304
Computer Organization
Spring 2025
Assignment 3 – Base Converter

Due: Thursday, 2/27/2025 11:59 p.m.

For this project, you will write a program in C that displays a value in various base representations. The program's execution is based on the following menu:

```
**************************************
* Base Values:          Input Mode: Dec *
*   Hex     :  0000                    *
*   Octal   :  000000                  *
*   Decimal :  0                       *
**************************************

Please select one of the following options:

O  Octal Mode
H  Hexadecimal Mode
D  Decimal Mode

C  Clear Value
S  Set Value

Q  Quit

Option:
```

Note that the value in various base representations is normally displayed right before the menu is displayed. The menu option can be entered as either an upper- or lower-case character (note that the option should be received as a 10-character string to handle errors; the option should be located at position 0). If an invalid option is entered (more than one character or a character that is not a menu option), the program should state that an invalid option has been entered and redisplay only the menu. This behavior should be repeated until the Quit option (Q) is selected.

The O, H, and D modes allow the user to change the base for which values are entered from the user (default is decimal). When a valid mode is selected, the program should print the new mode (e.g., "**`Mode: Octal`**") as output and update the **`Input Mode`** on the menu (**`Dec`**, **`Oct`**, or **`Hex`**). From this point forward, all numerical values should be entered in this mode, until the mode is changed.

The C option resets the value to 0. Note that the initial value should be set to 0 when the program begins.

The S option allows the user to enter a value based on the mode. For octal, hexadecimal, and decimal, standard conversion characters can be used in **`scanf`**, combined with the character 'h' to indicate that the scanned value be converted to **`short int`**, which is the correct type of integer to use for this project.

Your code must be modular and use functions, **switch** statements, character arrays, **printf**, and **scanf** (not **getchar**). Note that all input should be echo printed. Create the following functions, no more and no less, with the given interfaces to implement the program:

```
short get_operand (char mode)           // read in numeric value in mode; return value
void print_bases (short val, char mode) // print out base values and str for mode
char print_menu (void)                  // print menu; get option until valid; return option
int main (void)                         // main menu loop; execute option or call
                                        //  appropriate function
```

If you need to assign a string, use **strcpy()** (must include **string.h**). To uppercase a string, use **toupper()** (must include **ctype.h**).

All code should be submitted in a single source file named **bconv.c**.

Your program can be compiled with the command:

```
gcc bconv.c –o bconv
```

..and run with the command to test with the sample input file on the class webpage:

```
./bconv < test.txt
```

or with the command (to save the output):

```
./bconv < test.txt > mytestout.txt
```

Compare results with sample results on the class webpage using the command:

```
diff mytestout.txt testout.txt
```