

Assignment 5: Ascii and Boxes

Complete assignment due: Thursday, October 2nd, 9 p.m.

General notes about using C on assignments:

All C programs for this course will be compiled using **gcc** on *lectura*. The **-Wall** flag will be used. To receive full credit, the program must compile cleanly: no errors, no warnings. Points lost due to compilation warnings or errors can not be gotten back via a re-grade.

Problem 1 (50 points): Ascii.

Write a C program, named **ascii.c** that takes two command-line arguments. **ascii** use the **atoi()** function to convert the first argument to an **int**. It will take the first character of the second command-line argument. **ascii** will print two lines to stdout.

The int obtained from the first argument will be a range. The char obtained from the second argument will be the mid-point of a sequence of ascii characters. The sequence will extend for range chars on either side of the mid-point. On the first line, print the characters in the sequence. On the second line, print the same sequence, but this time convert all lower-case letters to upper-case and vice versa.

When the range will cause the sequence to start below ascii value of **1**, start the sequence at **1**. When the range will cause the sequence to start above the last ascii character, end the sequence at the last ascii character.

Here are some examples:

```
lectura-> ascii 5 'C'
>?@ABCDEFGH
>?@abcdefgh
lectura-> ascii 12 'Z'
NOPQRSTUVWXYZ[\]^_`abcdef
nopqrstuvwxyz[\]^_`ABCDEF
lectura-> ascii 1 '5'
456
456
lectura-> ascii 0 '5'
5
5
lectura->
```

Turnin: Use the **turnin** program to turn in your completed **ascii.c** program. The command is:

```
turnin 352assign5 ascii.c
```

See the man page for the **turnin** program for details on what turn in can do and how you can confirm that your file was turned in.

Problem 2: boxes

Write a C program named **boxes.c** that will print “boxes”. There will be two command-line arguments. The first argument will specify the size of the box. Use `atoi()` to convert the argument to an int.

The second command-line argument will be **0** or **1**. If the argument is **0**, then a box consists of `*`'s along the top, bottom, and sides of the box. The interior spaces in the box will be blank spaces. If the argument is **1**, then a box will consist of `*`'s everywhere; that is, the box will be filled with `*`'s.

We will guarantee that both command-line arguments will be present, and each will be convertible to an integer. Your program will check that the requested size is in the range **1** to **80** (inclusive). If it is outside this range, print a Usage message to `stderr` and return **1**.

If the command-line argument is valid, print the requested box to `stdout` and return **0**.

Some examples:

```
lectura-> boxes 5 0
*****
*   *
*   *
*   *
*****
lectura-> boxes 1 1
*
lectura-> boxes -3 0
Usage: boxes size
lectura-> boxes 1040 1
Usage: boxes size
lectura-> boxes 6 1
*****
*****
*****
*****
*****
*****
```

Turnin: Use the **turnin** program to turn in your completed **boxes.c** file. The command is:
turnin 352assign5 boxes.c

Final Summary:

There are two files that you will need to turnin for full credit on this assignment:

```
ascii.c
boxes.c
```