

# CS 261 – C and Assembly Language Programming

## Program 2: ASCII Art Arrays

### 1 Motivation

The goal of this project is to develop a C program making use of C array facilities and looping constructs. It will also serve as the basis for a later program which will use multiple sources files and Makefiles.

### 2 Overview

“ASCII Art” refers to any use of the printable characters in the ASCII character set to generate some sort of larger image. This can range from the nominally useful (ASCII banners displaying a word or phrase) to the mostly pointless but awesome (there’s a version of the ‘bullet time’ scene from *The Matrix* rendered in ASCII art), and everything in between.

For this assignment, you will be creating a simple program to generate scalable ASCII art images. Your program will first display a 5x5 character ASCII art “image” of your choice. It will then prompt the user to enter a positive integer,  $n$ , and will then print out the original image scaled up to a size of  $5n \times 5n$ . A full specification for how your output should be formatted follows.

### 3 Code Requirements

Your base image must be at least five characters square, and contain a minimum of four different characters including ‘\’ (backslash). Larger images are fine, but be sure the 5 x 5 in the upper-left corner satisfies the character requirements. Your program must print the base image, followed by a blank line, immediately after it starts.

Prompt the user for a positive scaling factor. You can assume that the user will enter an integer, but not necessarily a positive one. If the user enters an invalid number (i.e.  $\text{one} \leq 0$ ) immediately prompt for a new number. Do not include blank lines between prompts.

Once you have a valid scaling factor, print a blank line. Then, generate a scaled version of your image and print it to the terminal. You are not required to hold the scaled version in memory before printing it, and attempting to do so complicates the overall program. All that is required is that you *print* the scaled version. You don’t need to worry about large versions presenting oddly in the terminal. Assume the windows is always wide enough.

In addition to obeying the above specification, your program must also satisfy the following requirements:

- You create and call at least one (useful) function in addition to `main()`.
- You make use of at least one array (I would recommend a `char[][]` but it is up to you.)
- Do not use dynamically sized or allocated arrays.
- Your code is well-formatted, adequately commented, and free of global variables.
- Your code compiles without warning or error using `-std=c99`, `-pedantic`, and `-Wall`.

- All of your source code is contained in a single file named `art.c`.

Submit `art.c` using Autolab.

## 4 Example

The sample image here is called a glider. Please come up with your own image. Check out Wikipedia for more information: [https://en.wikipedia.org/wiki/Glider\\_\(Conway%27s\\_Life\)](https://en.wikipedia.org/wiki/Glider_(Conway%27s_Life))

---

```

Scale factor = 1
/---\
|.0.|
|..0|
|000|
\---/

```

Enter a scaling factor: 1

```

/---\
|.0.|
|..0|
|000|
\---/

```

---

Invalid input

```

/---\
|.0.|
|..0|
|000|
\---/

```

Enter a scaling factor: -1  
Enter a scaling factor: 0  
Enter a scaling factor: 1

```

/---\
|.0.|
|..0|
|000|
\---/

```

---

---

Scale factor = 3

```
/---\
|.0.|
|..0|
|000|
\---/
```

Enter a scaling factor: 3

```
///-----\\
///-----\\
///-----\\
|||...000...|||
|||...000...|||
|||...000...|||
|||.....000|||
|||.....000|||
|||.....000|||
|||000000000|||
|||000000000|||
|||000000000|||
\\-\\-----//
\\-\\-----//
\\-\\-----//
```

---