

Lab Practice Problem 8

Filename: practice8_surname.c

In this activity, you will implement a program that loads and renders an ASCII art image from a text file. The program will read a set of coordinates and corresponding characters, place them on a 2D canvas, and display the final ASCII art enclosed by a border. When implemented correctly, your program should display the complete ASCII art on the terminal.

Try It Out

Write a program that reads input data from a file and prints the resulting ASCII art. The following function prototypes are provided that define the **interfaces** you need to implement:

```
void reset_art(char canvas[MAX_SIZE] [MAX_SIZE], int R, int C);  
    Initializes the entire canvas of size R × C, which includes the border area. The outermost rows  
    and columns are set to * to form the frame, while all interior cells are filled with blank spaces ' '.  
void print_art(char canvas[MAX_SIZE] [MAX_SIZE], int R, int C);  
    Prints the contents of the canvas on the screen using the actual dimensions R and C. Since the  
    border is already part of the stored canvas, this function simply outputs the array as-is, displaying  
    both the artwork and its frame.  
void draw_art(char canvas[MAX_SIZE] [MAX_SIZE], int R, int C, int x, int y, char s);  
    Places the symbol s on the canvas at the specified coordinate (x, y). The parameters R and C  
    correspond to the total number of rows and columns, which includes the border area.
```

In the `main()` function, you will open the input file and read the canvas dimensions. After obtaining these values, compute the actual dimensions by adding two to both the row and column values to account for the border. Then, call `reset_art()` to initialize the canvas, setting the outer edges as borders and the interior as blank spaces. Next, read each coordinate and its corresponding symbol from the file, and draw each symbol on the canvas using `draw_art()`. Once all symbols have been placed, use `print_art()` to display the completed ASCII art, including the border.

Input

The program reads data from `data.txt`. The first line contains two integers, `R` and `C`, specifying the number of rows and columns of the canvas area, excluding the border (where $0 < R, C \leq 50$). Each subsequent line contains three values: `x`, `y`, and `s`, representing the 1-based coordinates and the character symbol to be placed at that position within the canvas. The top-left corner of the canvas corresponds to the coordinate (1, 1). The border will be automatically added by the program when the canvas is initialized.

Output

Your program should display the completed ASCII art on the screen, surrounded by a border made of `*` characters. The total printed dimensions are $(R + 2) \times (C + 2)$, accounting for the top, bottom, and side borders in addition to the canvas area.

Sample Input

4	8	
2	3	X
2	4	X
3	5	X
3	6	X

Sample Output

```

*****  

*          *  

*    XX    *  

*          XX  *  

*          *    *  

*****
```

Expected Output of data.txt

Note: The following artwork was sourced from <https://www.asciiart.eu/> and remains the property of its original creator. It is used here solely to demonstrate the concept in this activity.

Practice: If you like to experiment further, you can use the additional input file provided [here](#).

Guide Questions

1. How can the program determine that there are no more coordinates left to read from the input file? What condition or function can be used to detect the end of the file has been reached?
2. Why is it important to reset the canvas before drawing any characters? What potential issues might occur if the canvas is not cleared first?
3. How are the coordinates (x , y) translated into positions within the 2D array representation of the canvas? Why is it necessary to understand this mapping?
4. What could happen if a coordinate in the input file falls outside the valid range of the canvas? If the coordinates exceed the specified R and C values but are still within the maximum array size (50), will those drawings be visible when the canvas is printed?
5. What ASCII art was produced when you used the other input file?

Sample Solution

The solution will be released after the submission window has closed. You may review it on the activity page at that time.