Use of named pipes
I used named pipes because I was using my version 2 code to start version 3, and I wouldn't have been able to pass a pipe or a pointer to a pipe with execlp very easily. I also had already set up the pipe code basically since I could just name the pipes the same way I named the output files and still use fopen, fscanf, fgetc, and etc on the named pipe in my child process, so no changes were necessary for writing or reading the output of the child processes.

Implementation of the pattern search
First I read in the image and put all the information into a 2D array by first reading the first line of the image file and saying the first number is the number of rows present, while the second is the number of columns and I allocated space for the number of 1D arrays inside the 2D array based on the number of rows, and the number of elements in each 1D array is based on the number of columns. I do the same thing for every pattern file, and store the 2D arrays in a 3D array. Then the comparisons can begin. I did my comparisons in groupings of 3 characters and looping through the rows of the image. I started out by making a string out of the first 3 characters in a given row of the image and I check that against the first line of the pattern file. If they match, I check to see if there's a match, with the second line of the pattern file, at the same columns as where the first match was found and one row down. If those match I can do a similar check for if the 3rd line of the pattern file matches the offset 3 row, same columns, in the image file.

possible difficulties encountered

- It was difficult to follow back segmentation faults that occurred in any child processes, even with using gdb.
- difficulties with putting the code in a shared folder (when editing the code in the vm), as in I got different output when it was in a shared folder vs when it was out.
- print statements not showing up if there was any fault in the execution, unless I had "\n" at the end of them, that took a while to figure out.
- the directory operations function given was not the full path for opening a file, only the file name, needed to concatenate it with the full path from the argument
- had to repeat the code where I determined what the output file name would be, one for the parent process, and once for the child process.

possible limitations

- doesn't work if the file path is bad
- doesn't work if the first line of a pattern or image file is not a in the %d %d\n format
- if the pattern file isn't a 3x3 array, the image isn't compared against the whole pattern.