

Alejandro Porter

Dr. Lehr

CSC 17A

06 November 2022

Project 1: CandyLand

Introduction

This project is of the classic game known as CandyLand. It involves a number of players taking turns spinning a spinner, moving their pawns through a colorful board themed around candy. I chose this board game because it was one of my favorites as a small child. I also think CandyLand is an enjoyable game that many people can recall having a good time in their own childhoods where they have played it. CandyLand was created in 1948 during the polio epidemic to be played in hospitals by adolescent patients, and I wanted to code this game as a way of keeping it alive.

Summary

My program is about 325 lines of code, and it meets the criteria for the first project primarily because it utilizes the concepts we have learned so far, such as pointers, structured data, functions with both pointers and structured data, and using character arrays. I used a fair amount of variables to handle the beginning of the game where the user selects a color for their pawn, to dictate when to begin the game, and to keep track of numbers like the random number to be generated for the player's new position on the board. I used a structure to hold the important variables like a player's total moves, previous position and new position on the board.

This project was pretty challenging since I had to use relatively new concepts that I am still in progress of fully learning. There were many hours spent updating a version of my project so that it used the concepts of passing structures into functions and pointers.

Description

My general path for programming the solution for this project was to create a very basic version of the game and use that as a starting point for adding the newer concepts in the rubric. For example, I did not start with using a structure or any pointers in my initial version, but I slowly replaced the existing code with these newer concepts. I also wanted my project to be as straightforward as possible so I made drawing random numbers for the players into a singular action from the user. The program will prompt the user to enter “d” and when done, it will draw a random number for both players, simulating how in real life, the players would use a spinner to get a random number. Below is a screenshot of this process.

```

Output ×
CIS17AProject1V3 (Build, Run) × CIS17AProject1V3 (Run) ×

Welcome to the game of Candyland
Objective: Be the first player to reach the Candy Castle by making it to end of the path.

Contents: 1 Colorful Gameboard with 85 linear spaces
          2 Gingerbread Character Pawns
          64 Cards Numbered 1-5

Type 'Begin' to begin your journey or 'Exit' to leave the game.
Begin
Player 1, Choose a Pawn Color (1 for Red, 2 for Blue)
1
Player 2, Choose a Pawn Color (1 for Red, 2 for Blue)
2

Player 1 - Red Pawn
Player 2 - Blue Pawn

Enter 'd' to draw a card for each player.
d

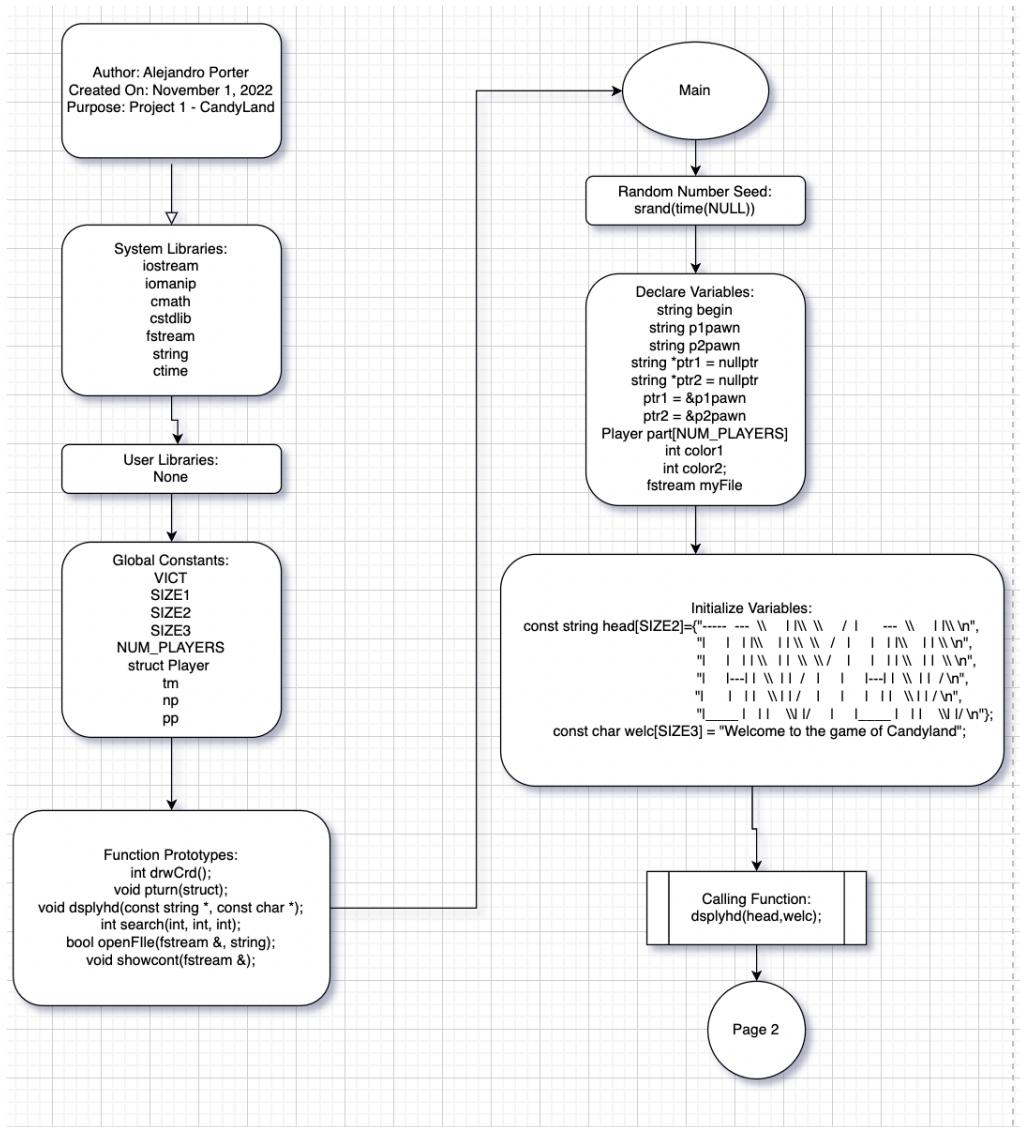
Player 1
Move # - Card Drawn(1-5) - Current Position - New Position - shortcut Tile Reached(if any)
1      2      0      2

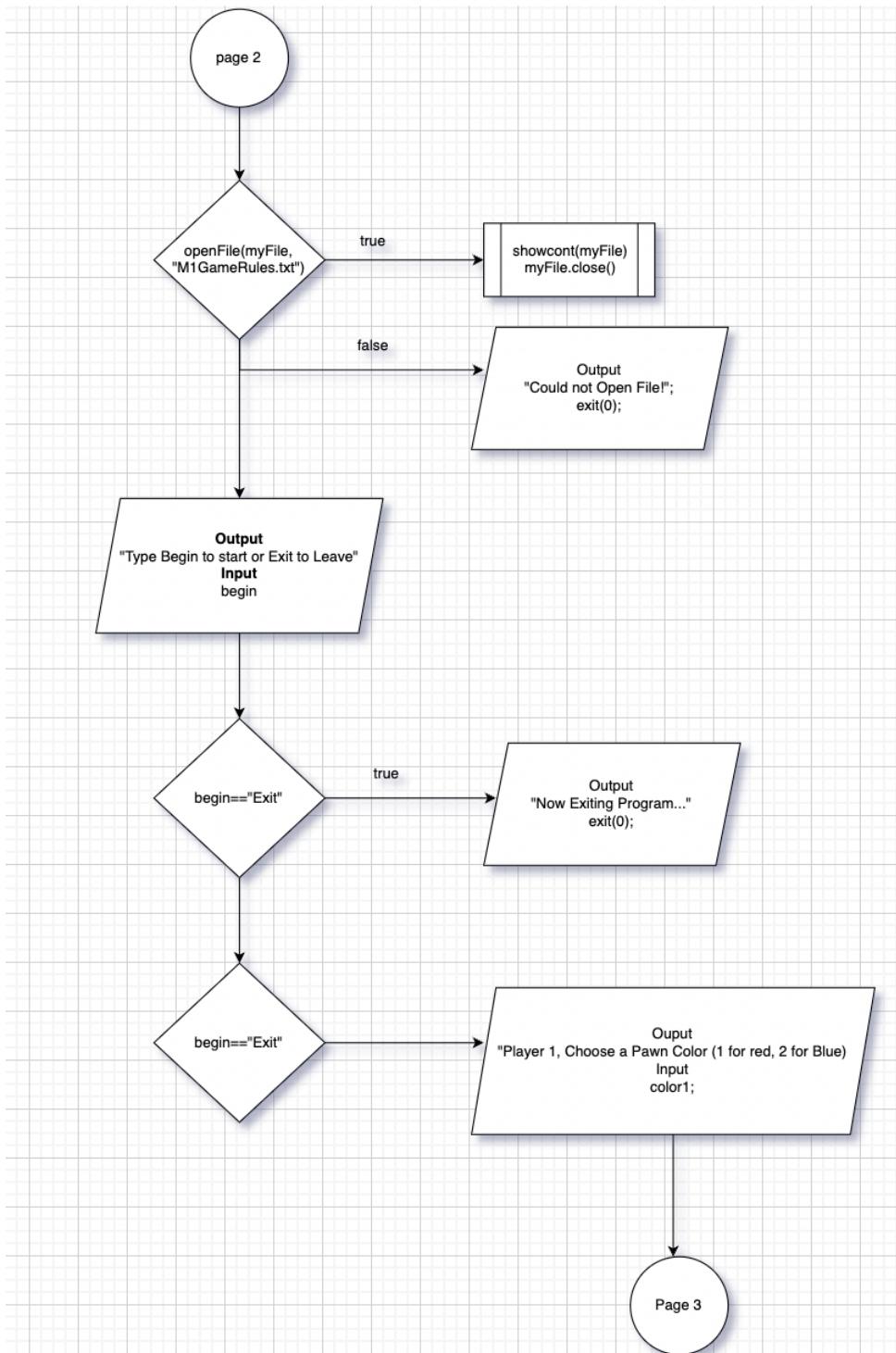
Player 2
Move # - Card Drawn(1-5) - Current Position - New Position - shortcut Tile Reached(if any)
1      5      0      5

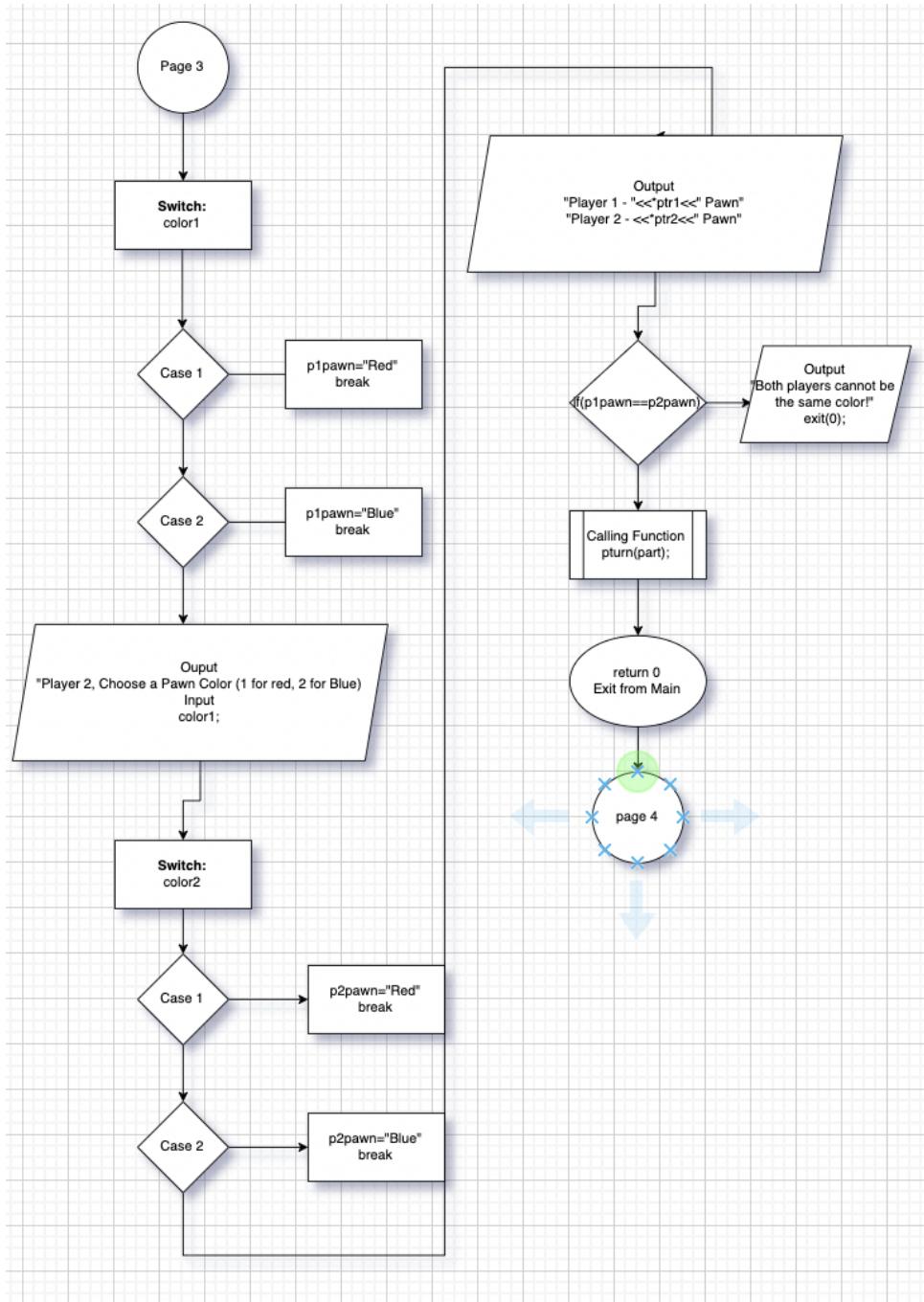
Enter 'd' to draw a card for each player.

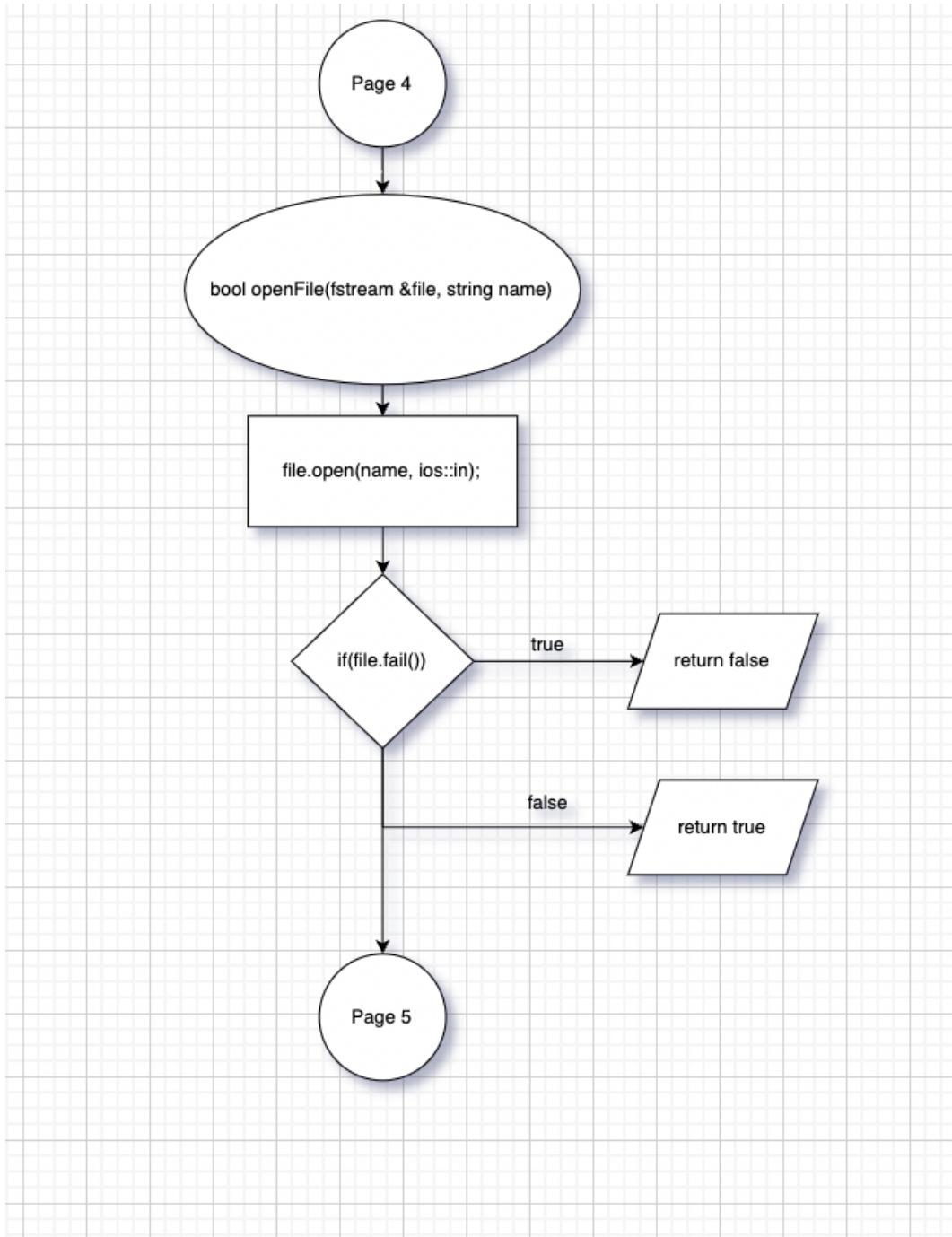
```

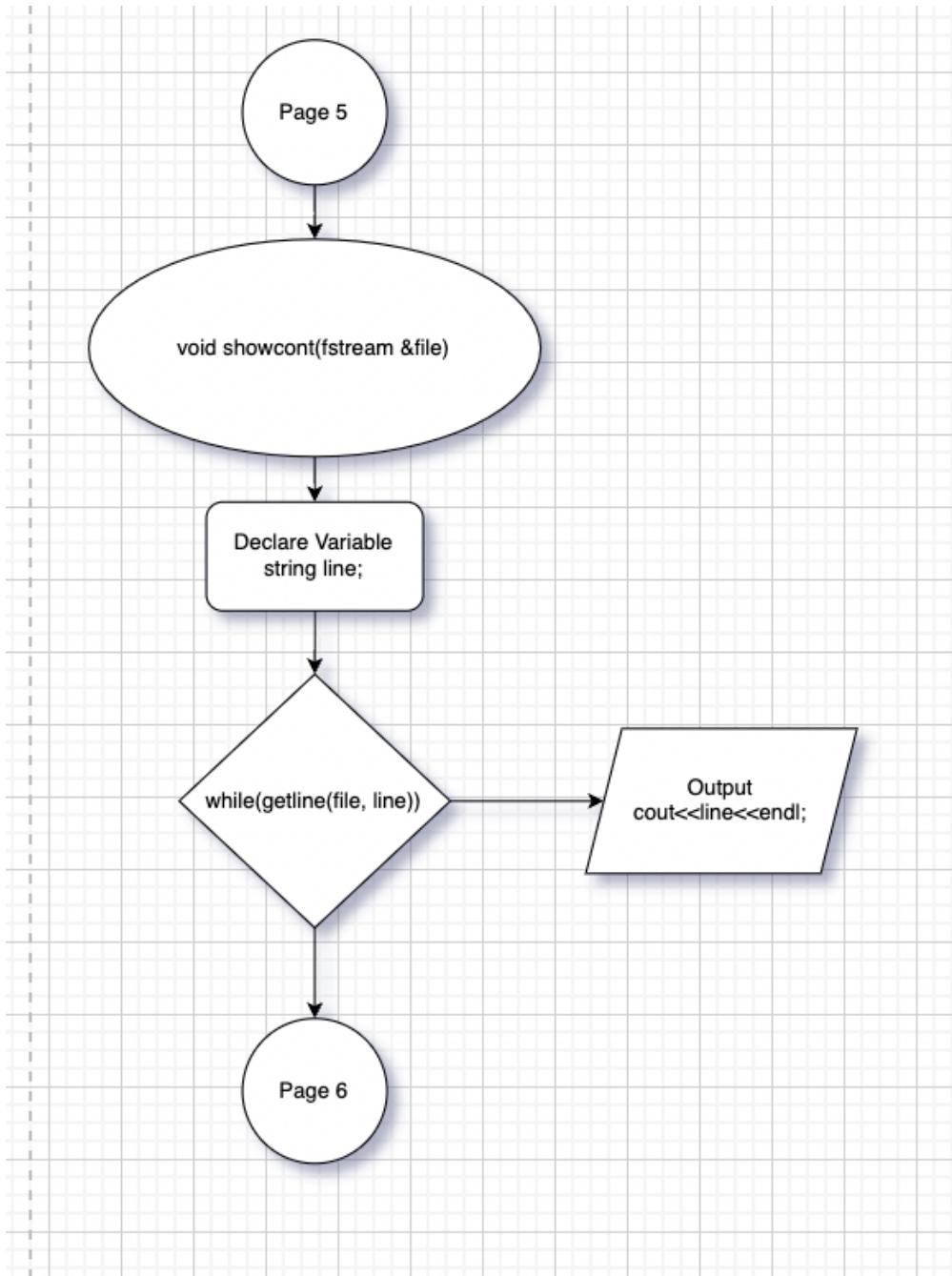
Flowcharts/Pseudocode/UML

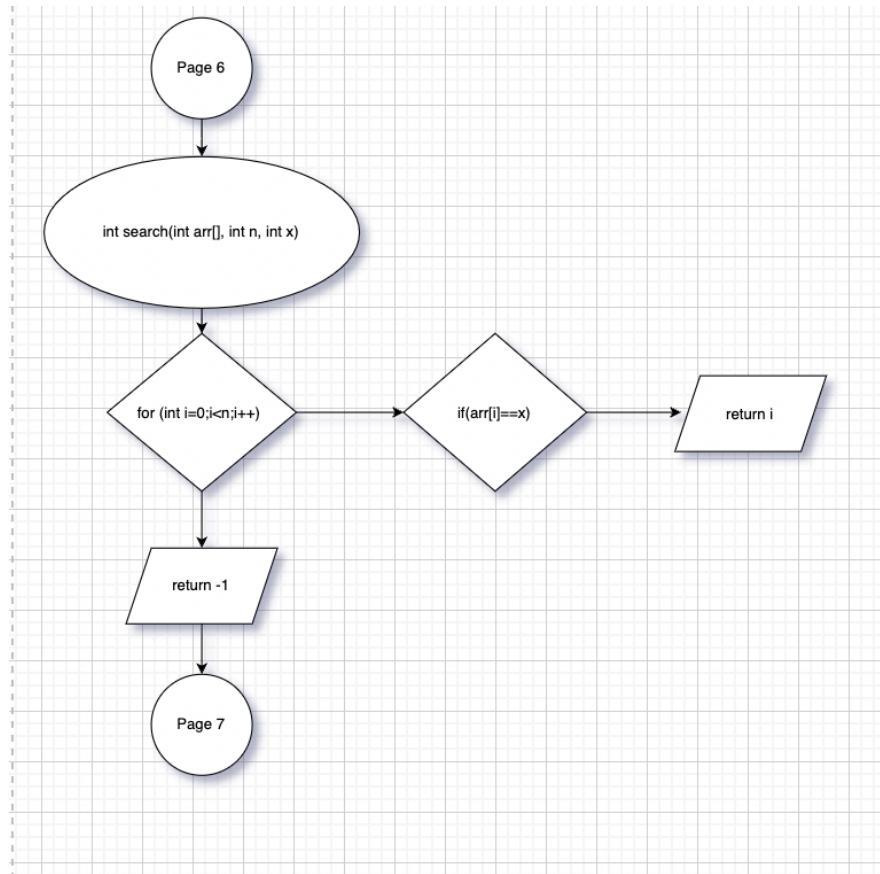


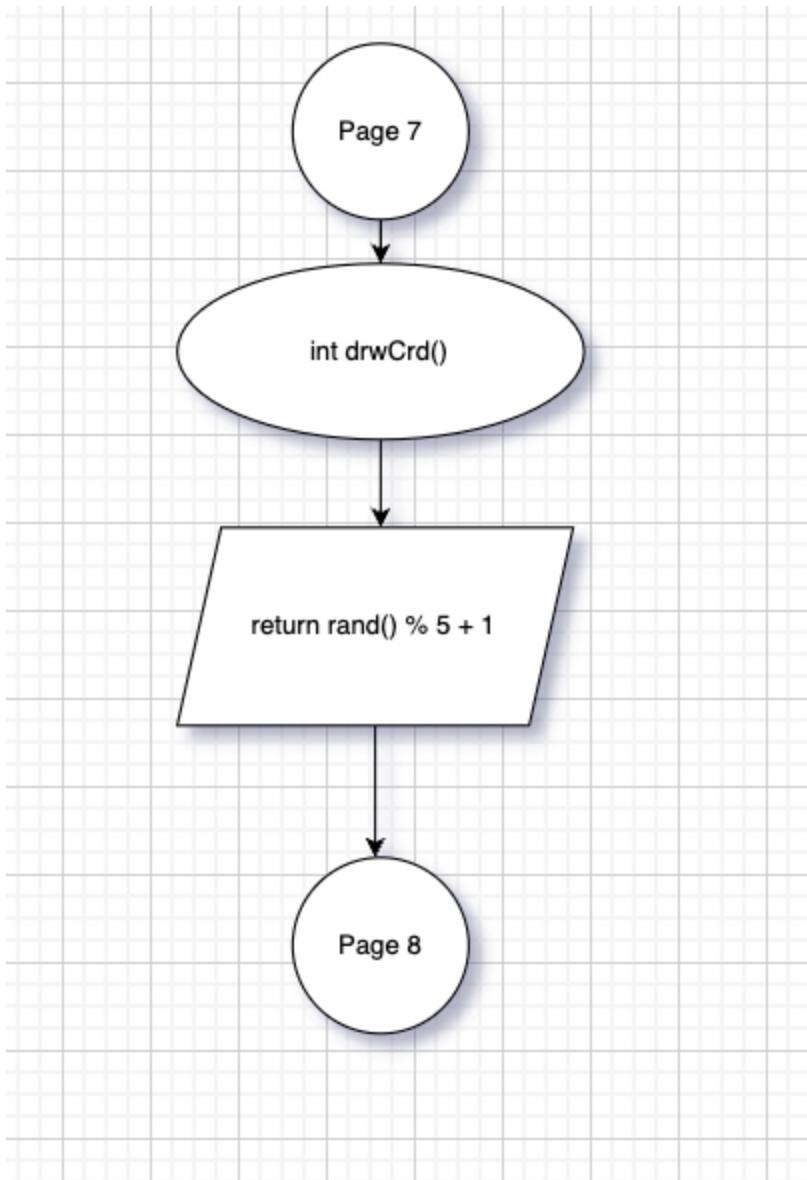


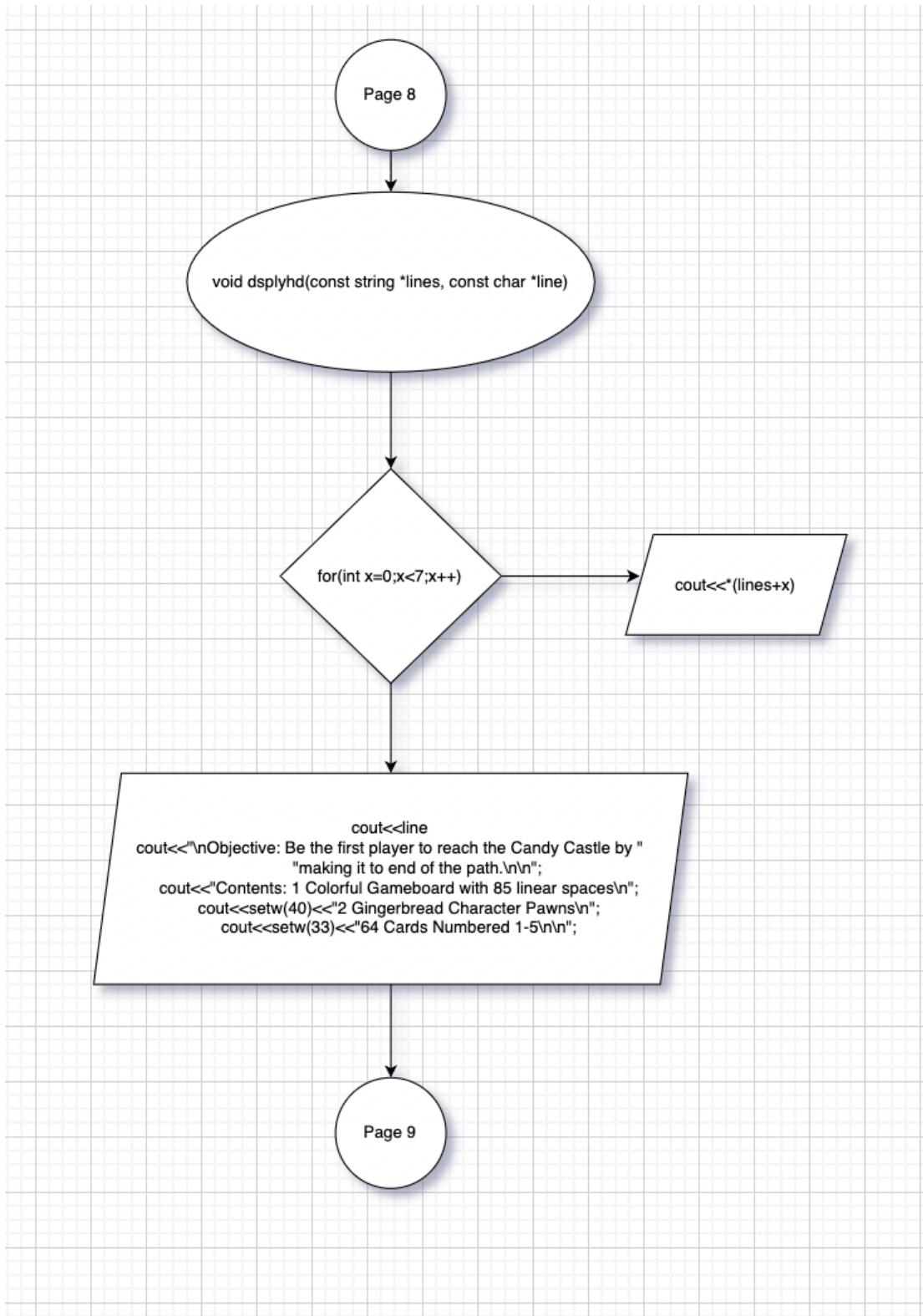


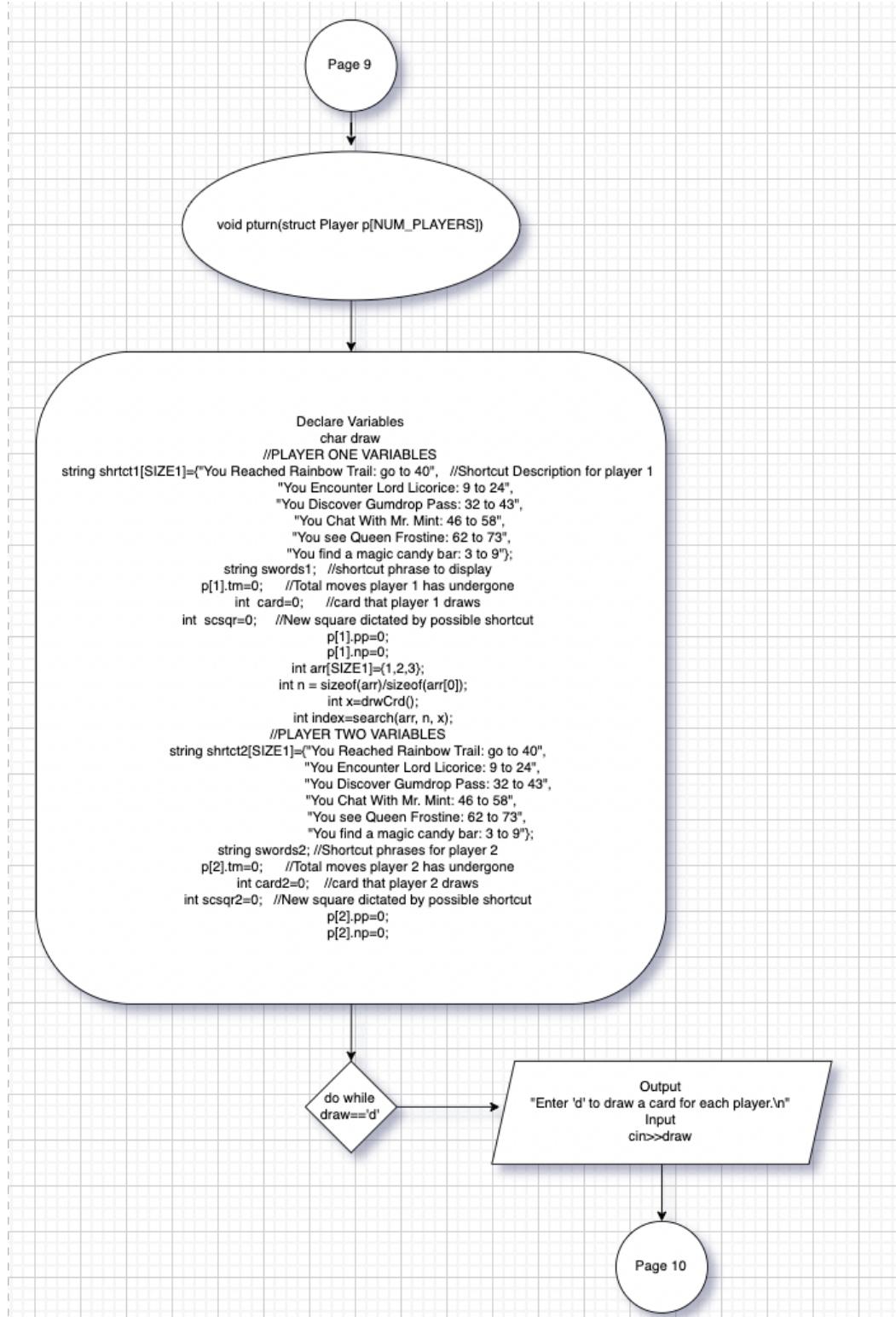


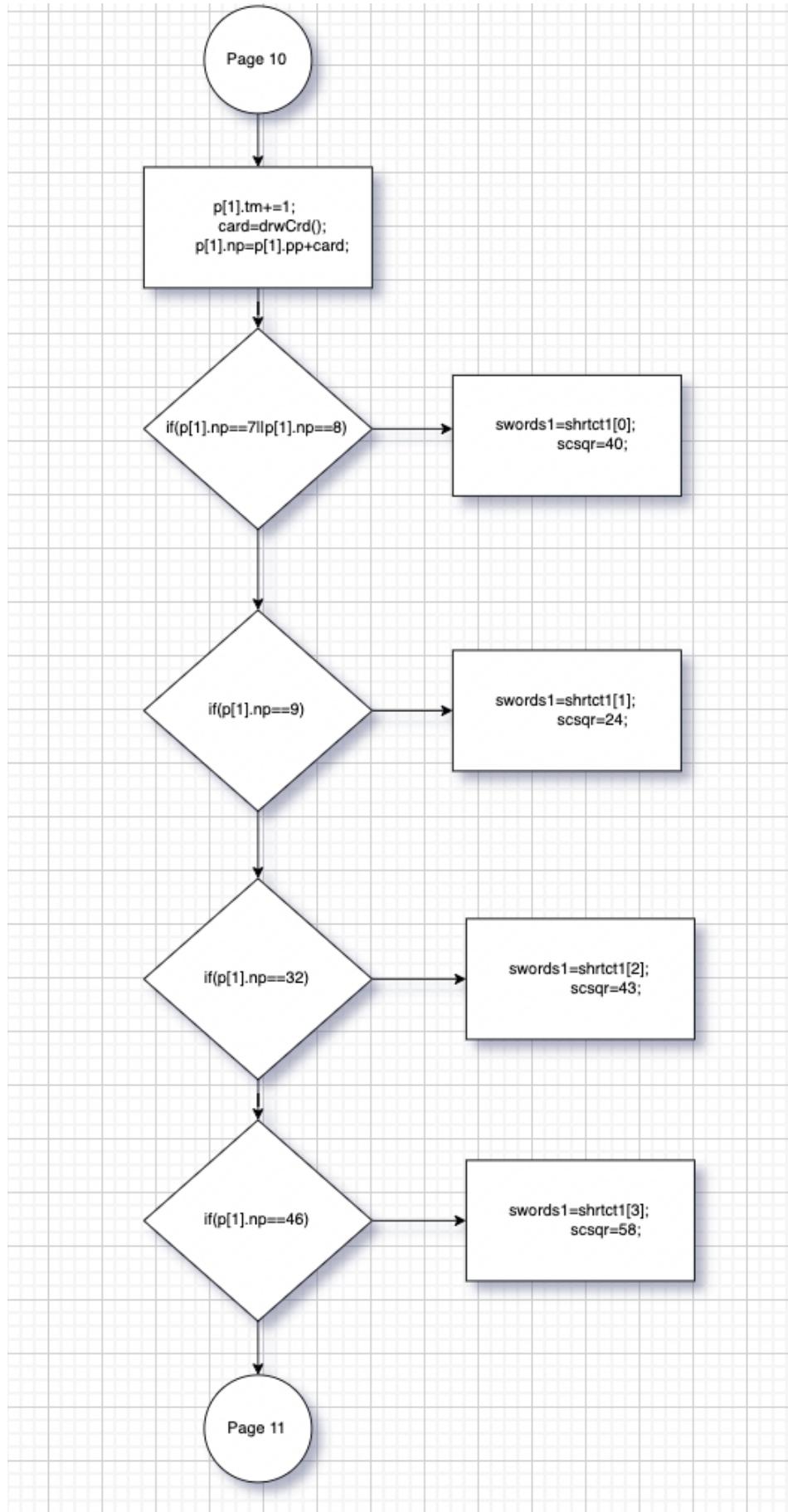


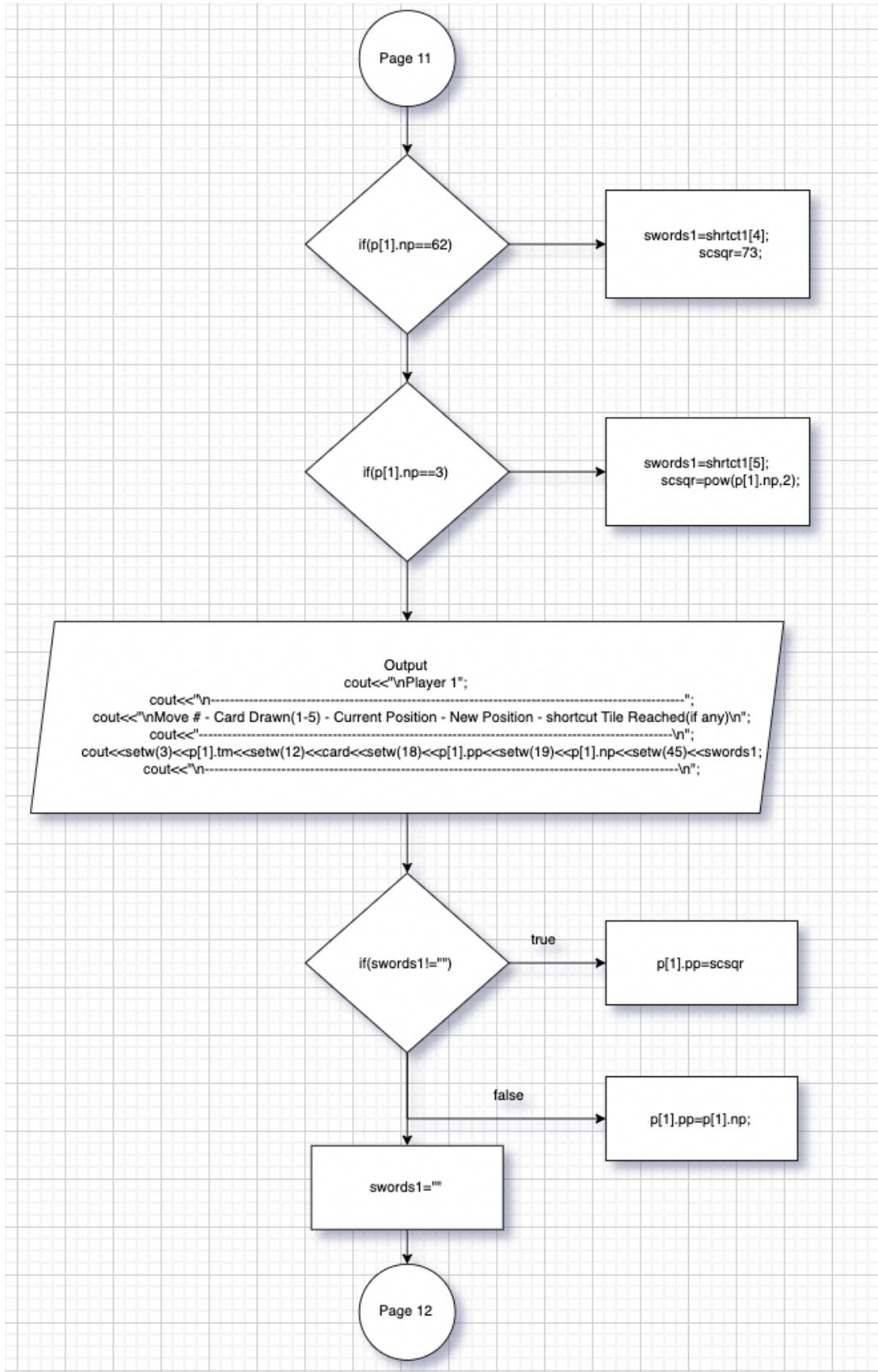


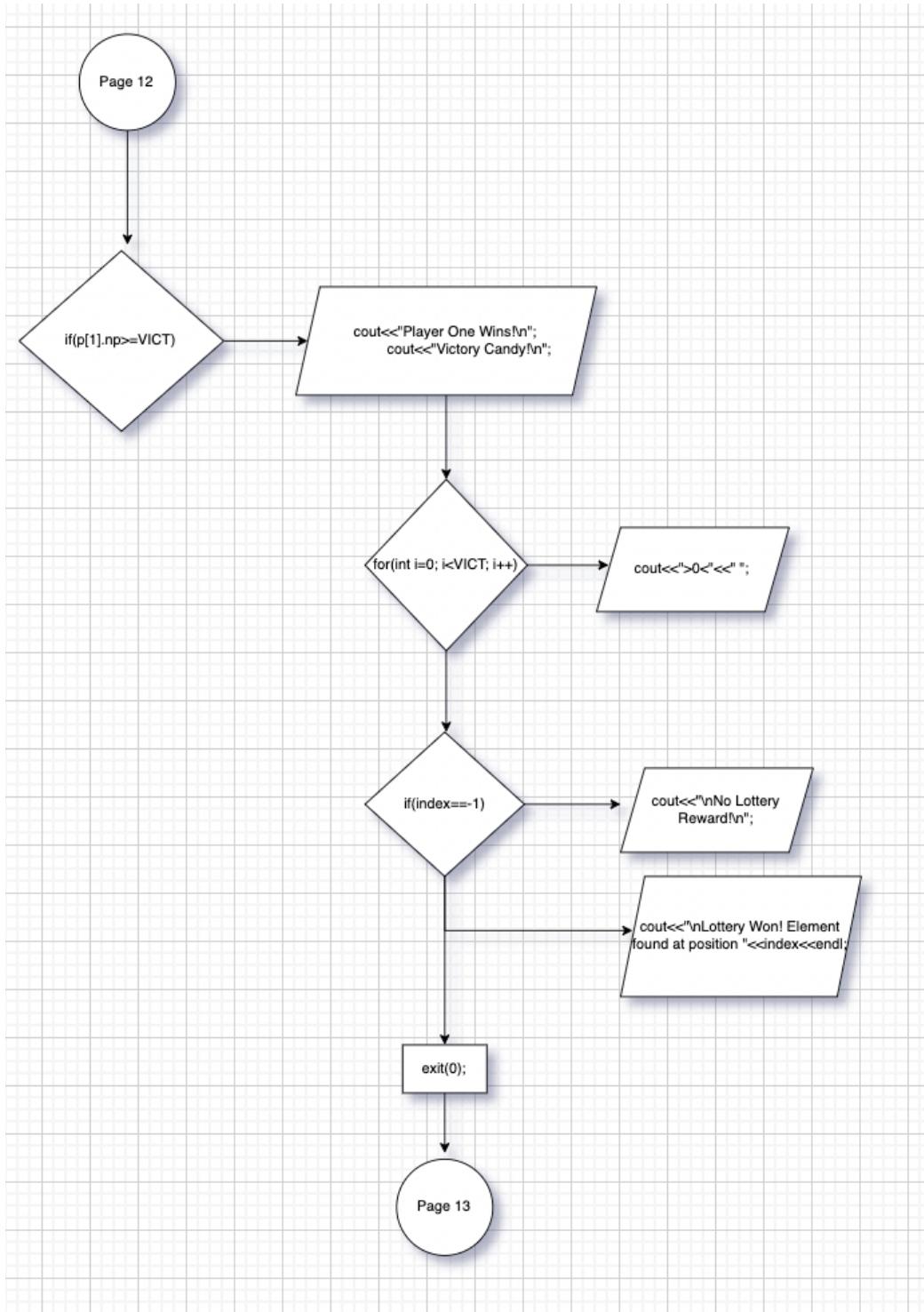


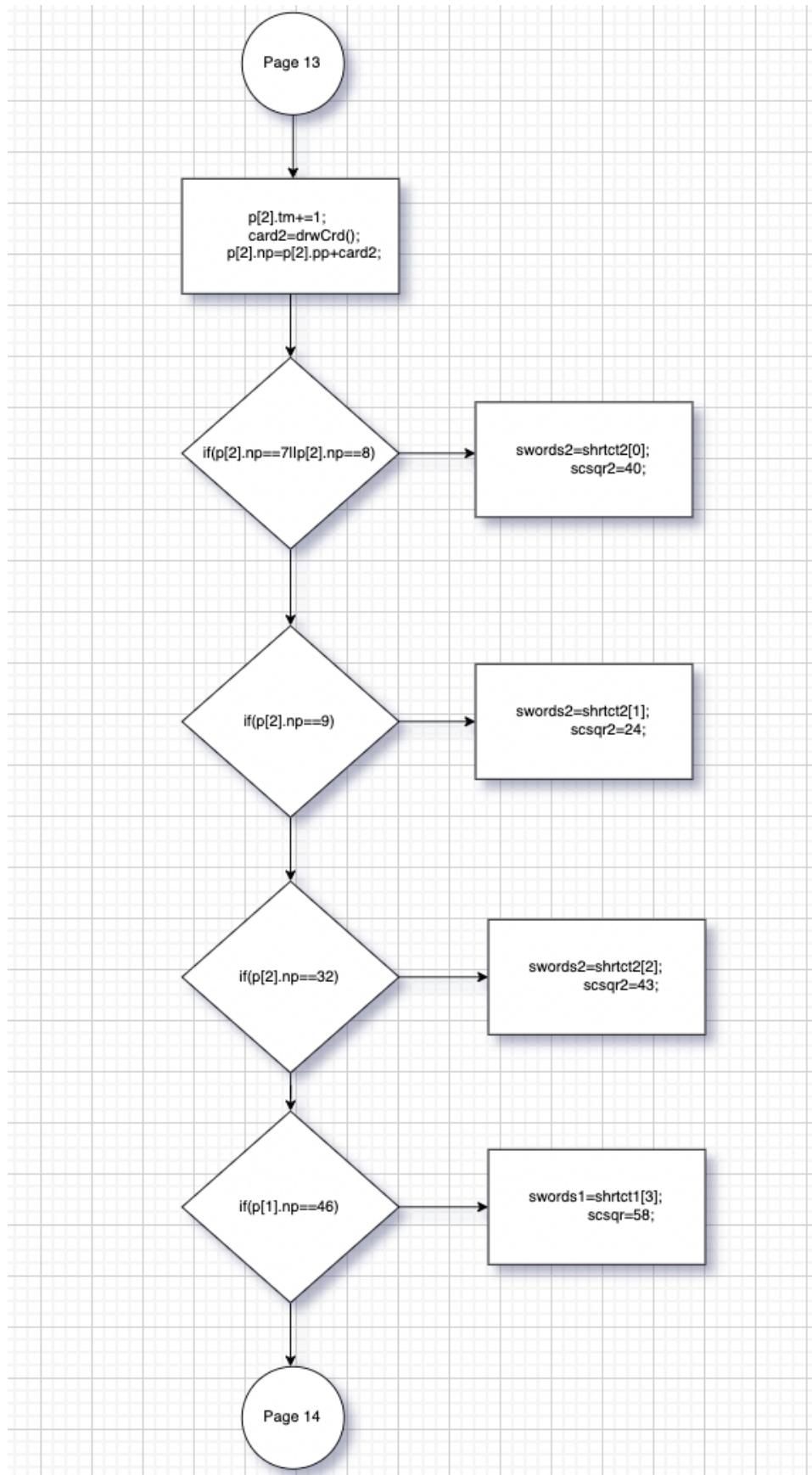


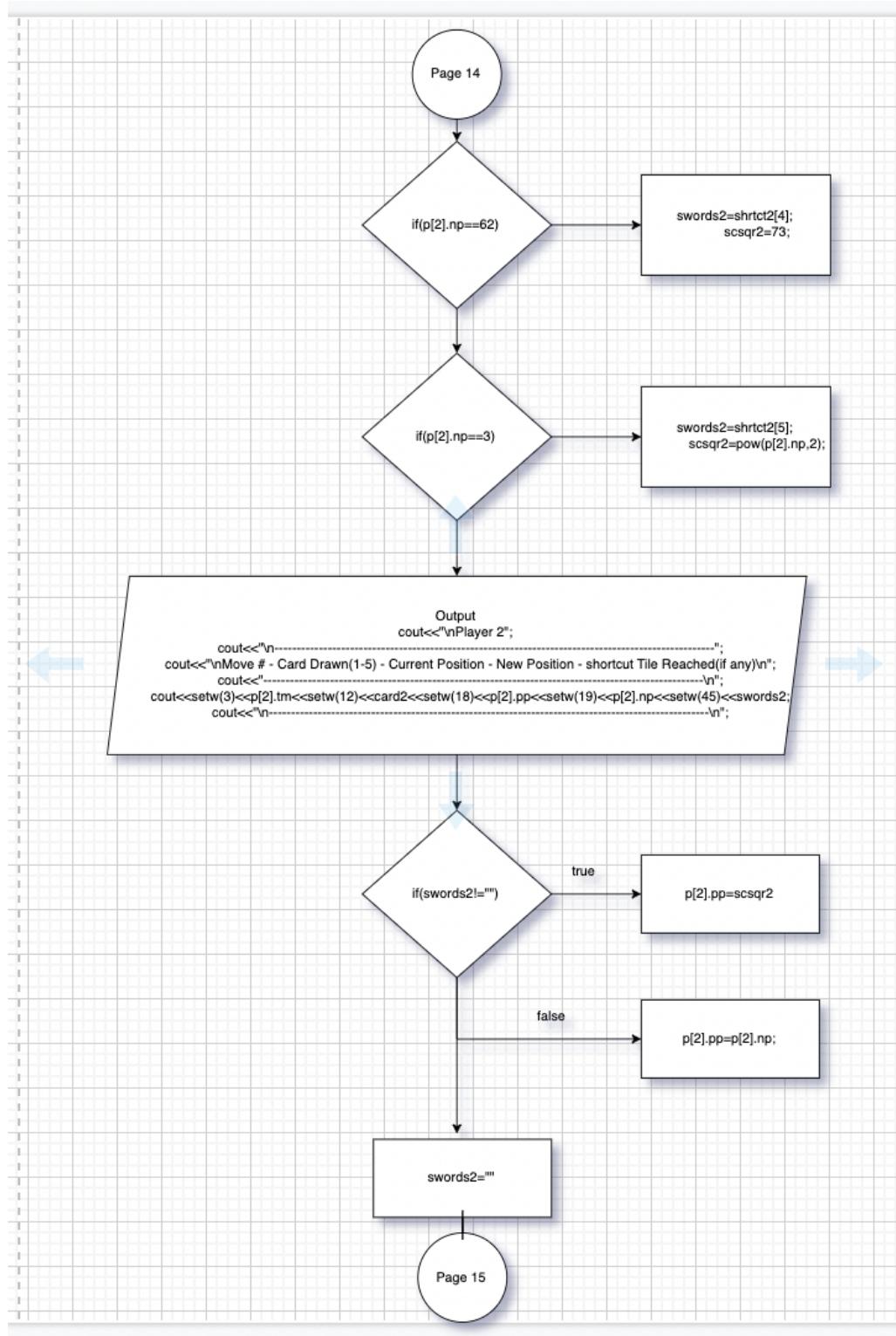


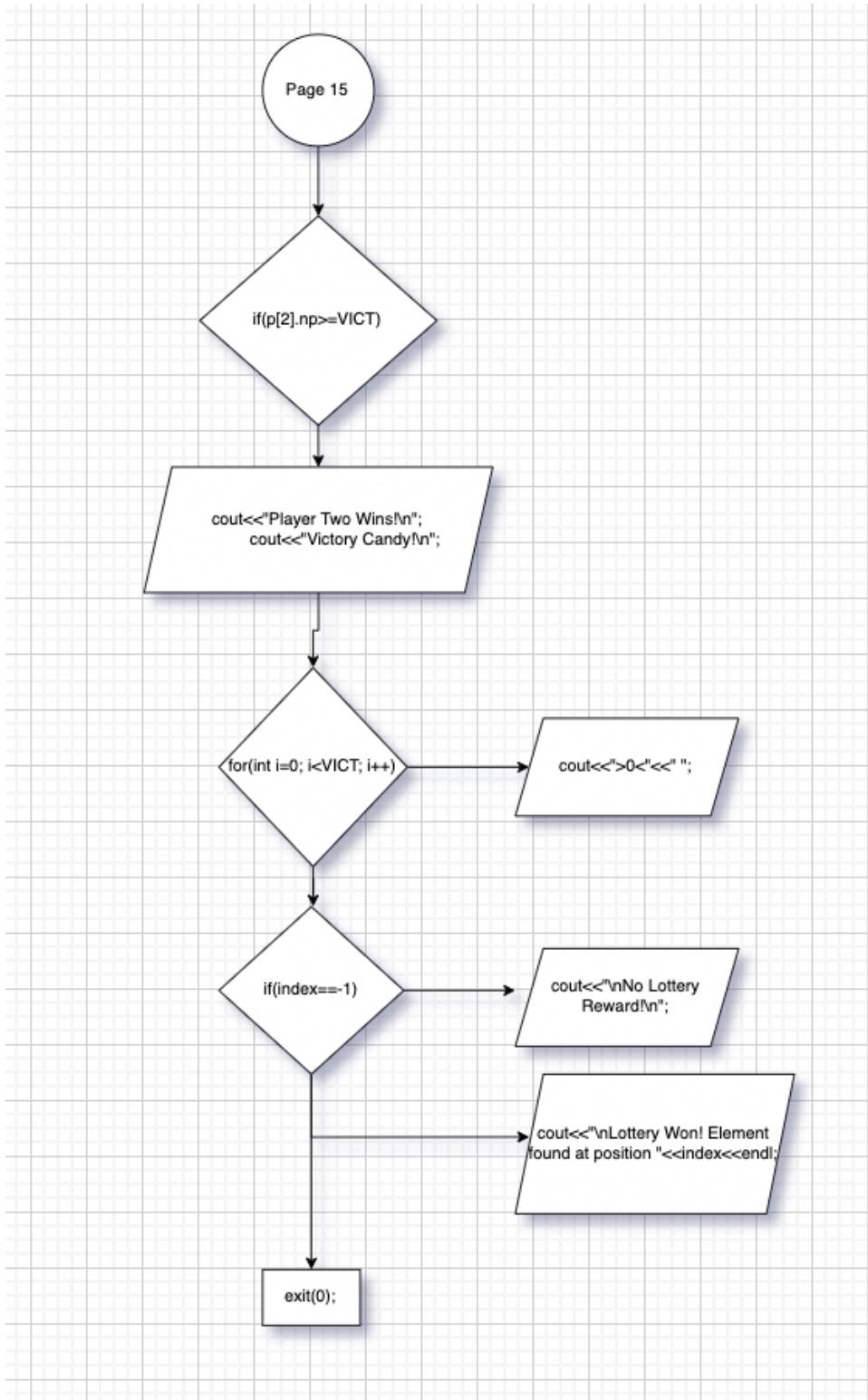












Pseudo-code

Initialize player structure

Initialize constants for victory condition, number of players, and number of elements in various arrays in the program.

If “Exit” is entered by user

 Exit Program

Else if “Begin” is entered by user

 Prompt user to choose pawn color

 Call the pturn() function

 Do

 Initialize total moves, previous position, and new position of player 1 and player 2.

 Prompt user to enter “d” to draw a random number for players 1 and 2.

 If player 1 or player 2’s position is equal to one of the shortcut tiles, add that shortcut to the player’s new position.

 If player 1 or 2’s new position is greater than or equal to victory square, output a candy reward and candy lottery.

 While user enters “d”.

Concepts/Check Off Sheet

Check-Off

CSC/CIS 17A Project 1 Check-Off Sheet

Chapter	Section	Concept	Points for Inclusion	Location in Code	Comments
9	Pointers/Memory Allocation				
	1 Memory Addresses				
	2 Pointer Variables	5	49		
	3 Arrays/Pointers	5	101		
	4 Pointer Arithmetic				
	5 Pointer Initialization		49		
	6 Comparing				
	7 Function Parameters	5	35		
	8 Memory Allocation	5			
	9 Return Parameters	5			
	10 Smart Pointers				
10	Char Arrays and Strings				
	1 Testing				
	2 Case Conversion				
	3 C-Strings	10	66		
	4 Library Functions				
	5 Conversion				
	6 Your own functions		66 and 35		
	7 Strings	10	60		
11	Structured Data				
	1 Abstract Data Types		54		
	2 Data		20		
	3 Access		168		
	4 Initialize		168		
	5 Arrays	5	54		
	6 Nested	5			
	7 Function Arguments	5	34		
	8 Function Return	5			
	9 Pointers	5	229		
	10 Unions ****				
	11 Enumeration	5			
12	Binary Files				
	1 File Operations		58		
	2 Formatting	2			
	3 Function Parameters	2			
	4 Error Testing				
	5 Member Functions	2			
	6 Multiple Files	2			
	7 Binary Files	5			
	8 Records with Structures	5			
	9 Random Access Files	5			
	10 Input/Output Simultaneous	2			
	Total		100		

References

I borrowed the CandyLand title that was made out of backslash and forward slash characters from a similar project online. The rest of the code was borrowed from the textbook, especially the code involving using an array of a structure as a function parameter.

Program (GitHub Link)

https://github.com/aporter30/2022_Fall_CSC_CIS_17a/blob/main/Project/1