Project 1

Title

Go Fish

course

CIS-5

Author

Paola Alcala

Game introduction

12 cards
The goal is to collect as many complete sets of 2 numerical matching cards.
Collect cards needed from players or go fishing from the remaining cards.

How to play Go Fish

1. Deal 3 cards to each player
2. Player 1 chooses a player and asks for a card.
3. Either other players gives up the card or says "Go fish".
4. If player told to "Go fish!" they grab a card from remaining deck and add it to their hand.
6. If player gets set of 4 matching cards remove from hand and add to matching cards pile.
7. The player's turn will end when they went "fishing" for card.
8. If player secures a card from another player they get another turn.
9. Continue until one player runs out of cards.
10. The game is won by the player who has collected the most matching set of 4 cards.

Development Summary

Lines of code: 205

Comment Lines: 71

Total lines of code: 259

The game is not fully functional. I ran into many blockers trying to figure out how accurately manipulate strings. I was able to get the game logic up to where a player picks a card from the remaining deck after being told to "Go Fish". After that I could not figure out how to accurately access the chars in the string via appropriate indexes.

Version 1:

In version 1 I created variables for players and cards. I created an input prompt for the players. I also attempted to figure out how to hold and display all 52 cards and their values in the variable: deckOfCards without using an array.

Version 2:

In version 2 I simplified the project to 2 players and decided to only make the deck of cards hold 20 cards. I decided to consider the deck as a sequence of numbers from 1 to 10 that repeats twice.

Version 3

In version 3 I implement the cstdlin and the ctime libraries to shuffle the deck of cards.

Version 4

In version 4 I updated the way to assign the player number based on user input. I updated the deck to represent a full deck of 52 cards. I implemented a way to split the deck of cards between to players by creating two different string variables for each hand and using the substr function.

Version 5

In version 5 I decided to address a bug that I found when I split the deck. I noticed that I was splitting the original deck in numerical order. I created a shffld string variable to hold the shuffled deck. I decided to shrink the deck of cards so that I can apply the substr function more accurately. I added the functionality where players ask each other for a card and the functionality where players tell each other to "go fish".

Version 6

In version 6 I added the functionality of players moving a card from one player to the other using the erase built in function and string concatenation.

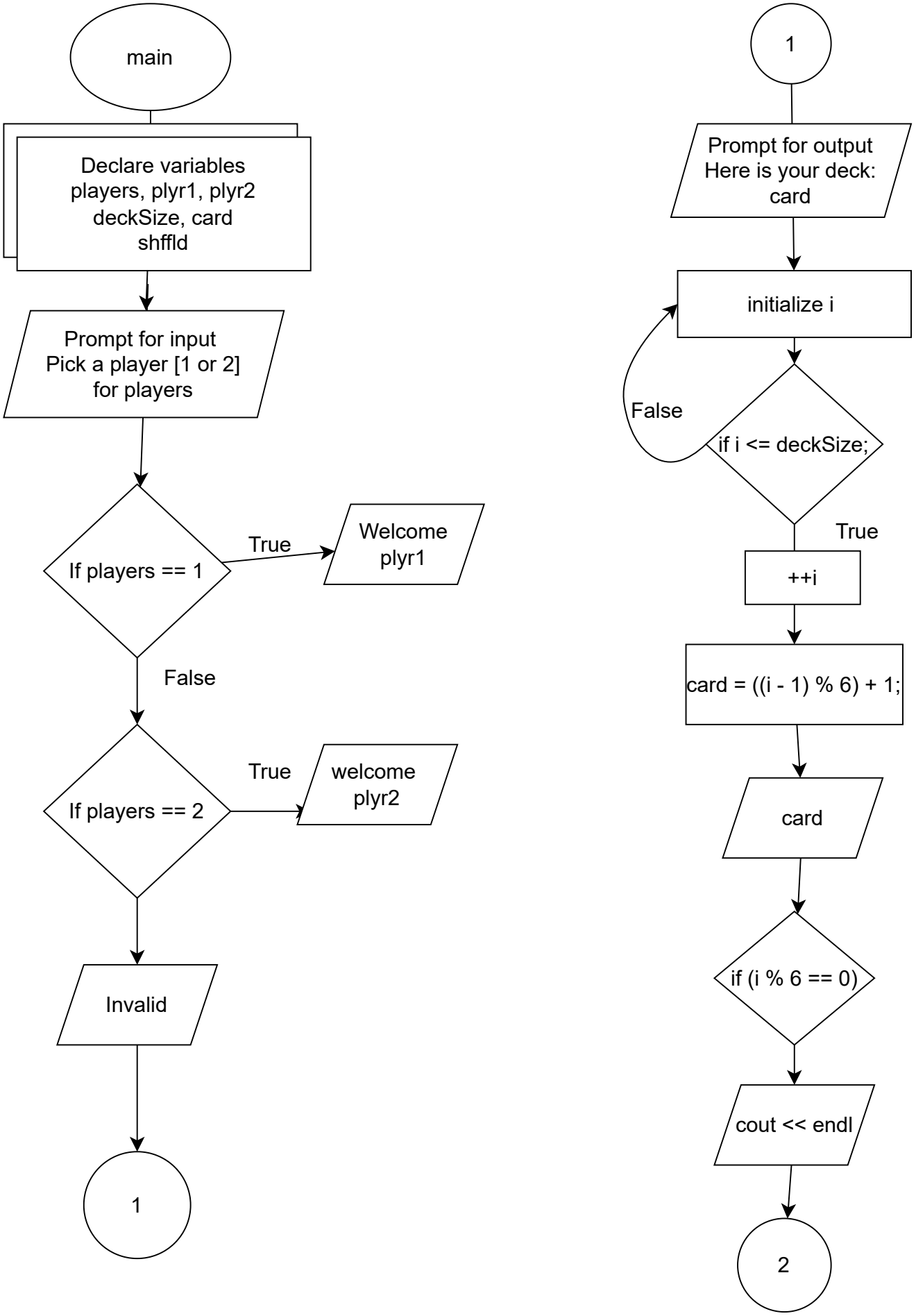
Version 7

In version 7 I implemented the functionality of locating matching cards from players hands. In this version I was still attempting to add the matching cards to a new string variable that would hold the matching pairs for each player. In this version I realized that I needed to add logic before the game llop to check for initial matches in the individual players hand.

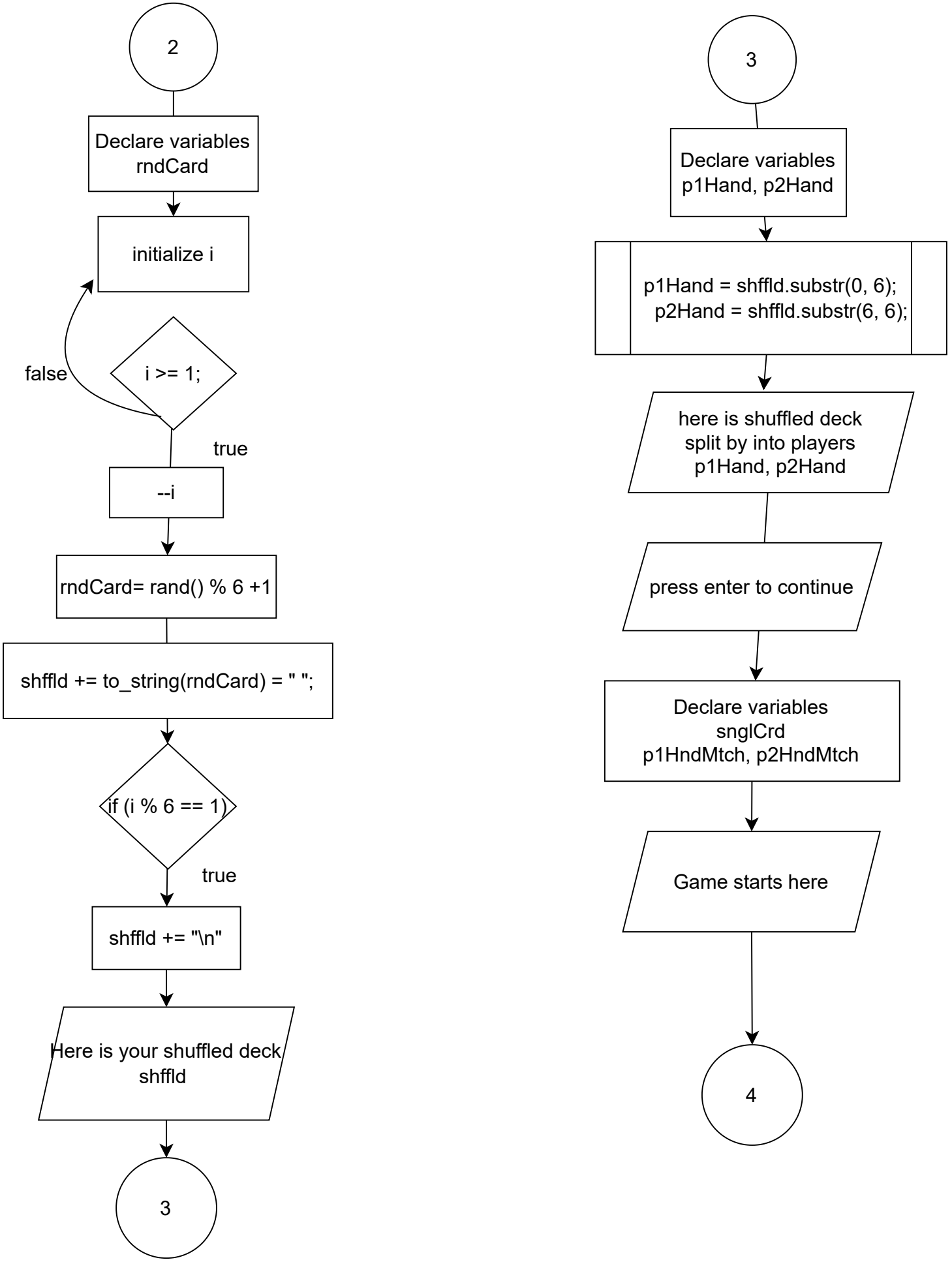
Version 8

In Version 8 I realized that it would be more productive to count the matches rather than concatinate the matching pairs in a new string variable. I attempted to write the logic for players picking a card from the shuffled deck but could not get it to work.

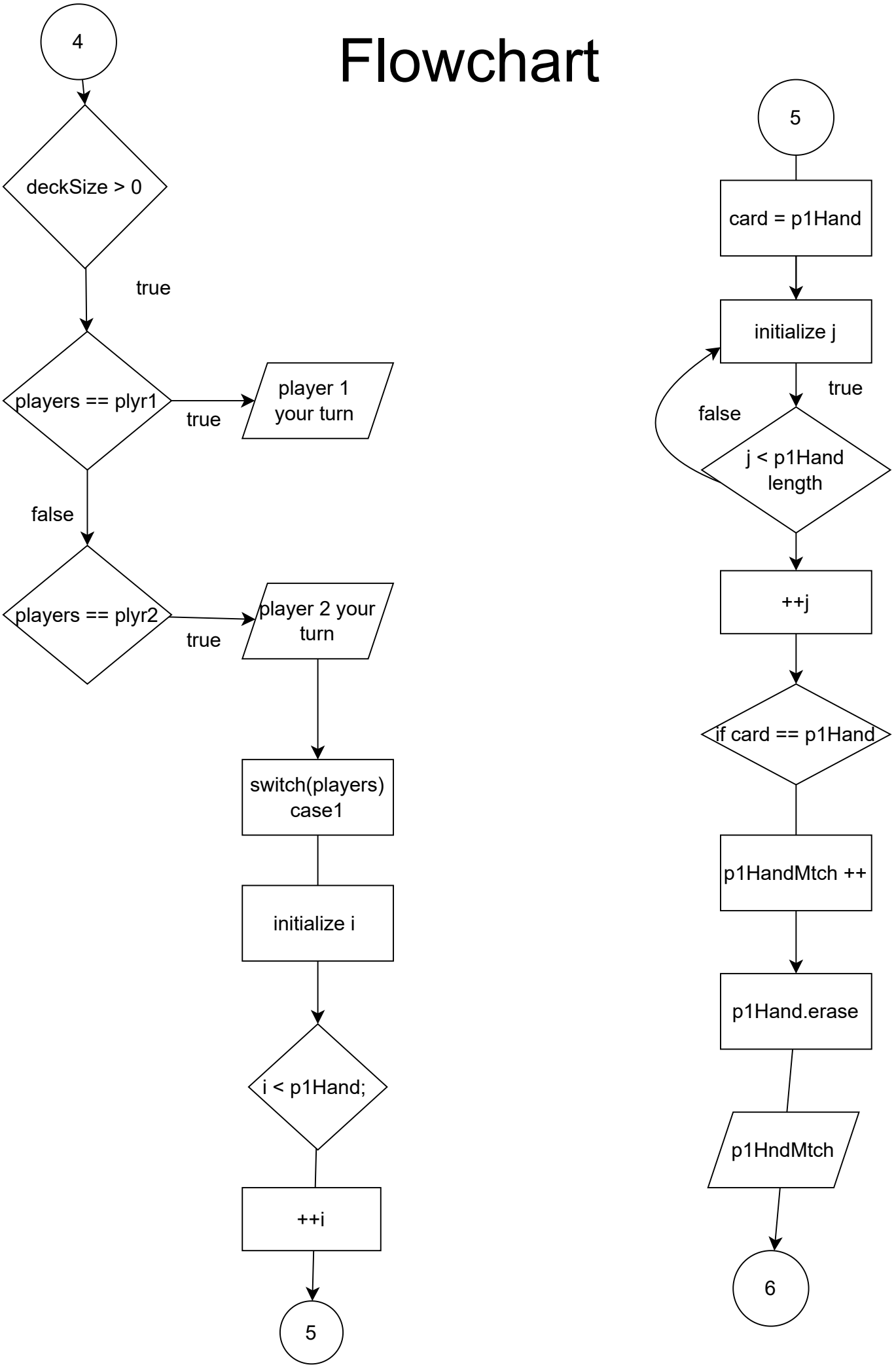
Flowchart


main
Declare variables
players, plyr1, plyr2
deckSize, card
shffld
Prompt for input
Pick a player [1 or 2]
for players
If players == 1
True
Welcome
plyr1
False
If players == 2
True
welcome
plyr2
Invalid
1
1
Prompt for output
Here is your deck:
card
initialize i
False
if i <= deckSize;
True
++i
card = ((i - 1) % 6) + 1;
card
if (i % 6 == 0)
cout << endl
2

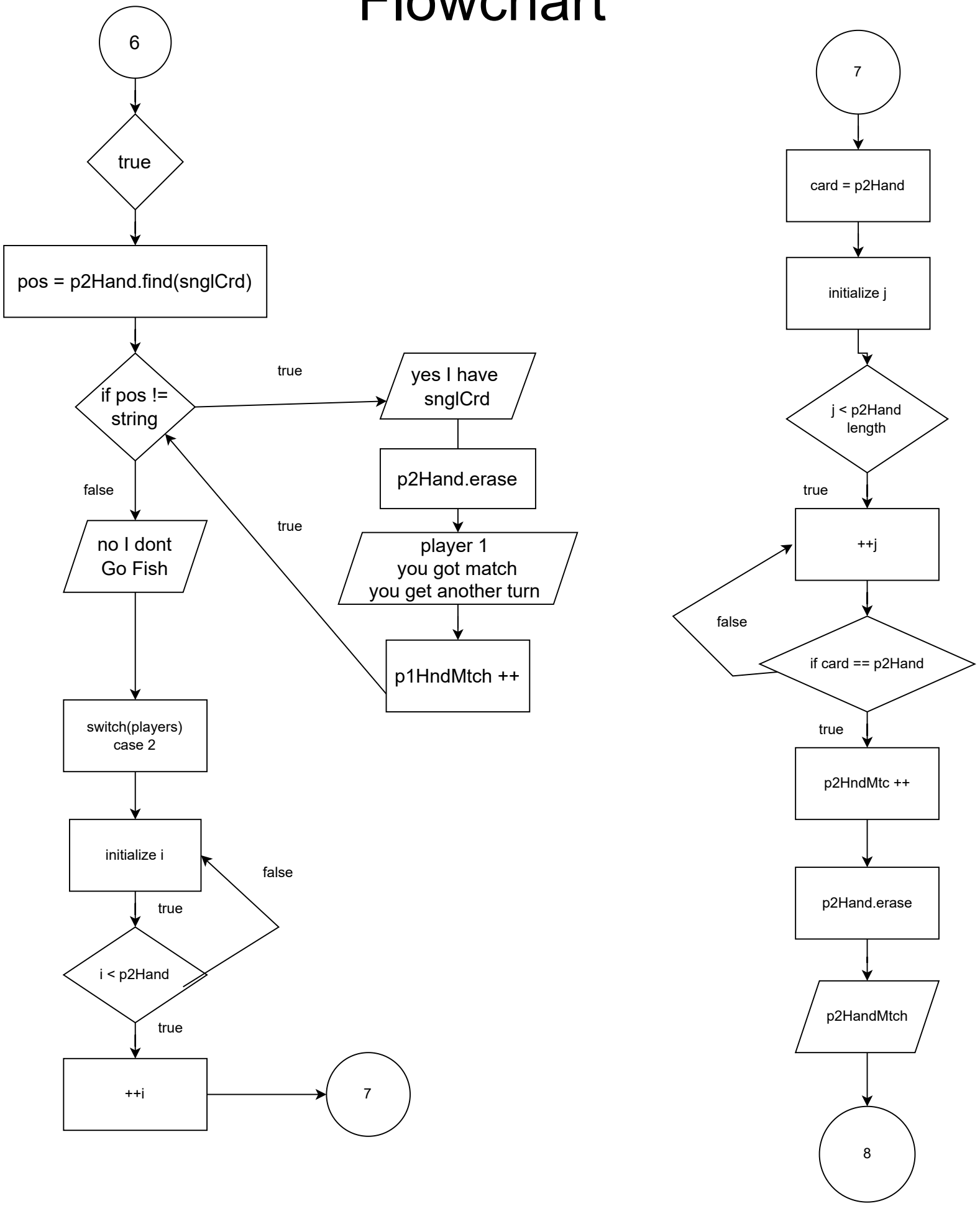
Flowchart


2
Declare variables
rndCard
initialize i
i >= 1;
false
true
--i
rndCard= rand() % 6 +1
shffld += to_string(rndCard) = " ";
if (i % 6 == 1)
true
shffld += "\n"
Here is your shuffled deck
shffld
3
3
Declare variables
p1Hand, p2Hand
p1Hand = shffld.substr(0, 6);
p2Hand = shffld.substr(6, 6);
here is shuffled deck
split by into players
p1Hand, p2Hand
press enter to continue
Declare variables
snglCrd
p1HndMtch, p2HndMtch
Game starts here
4

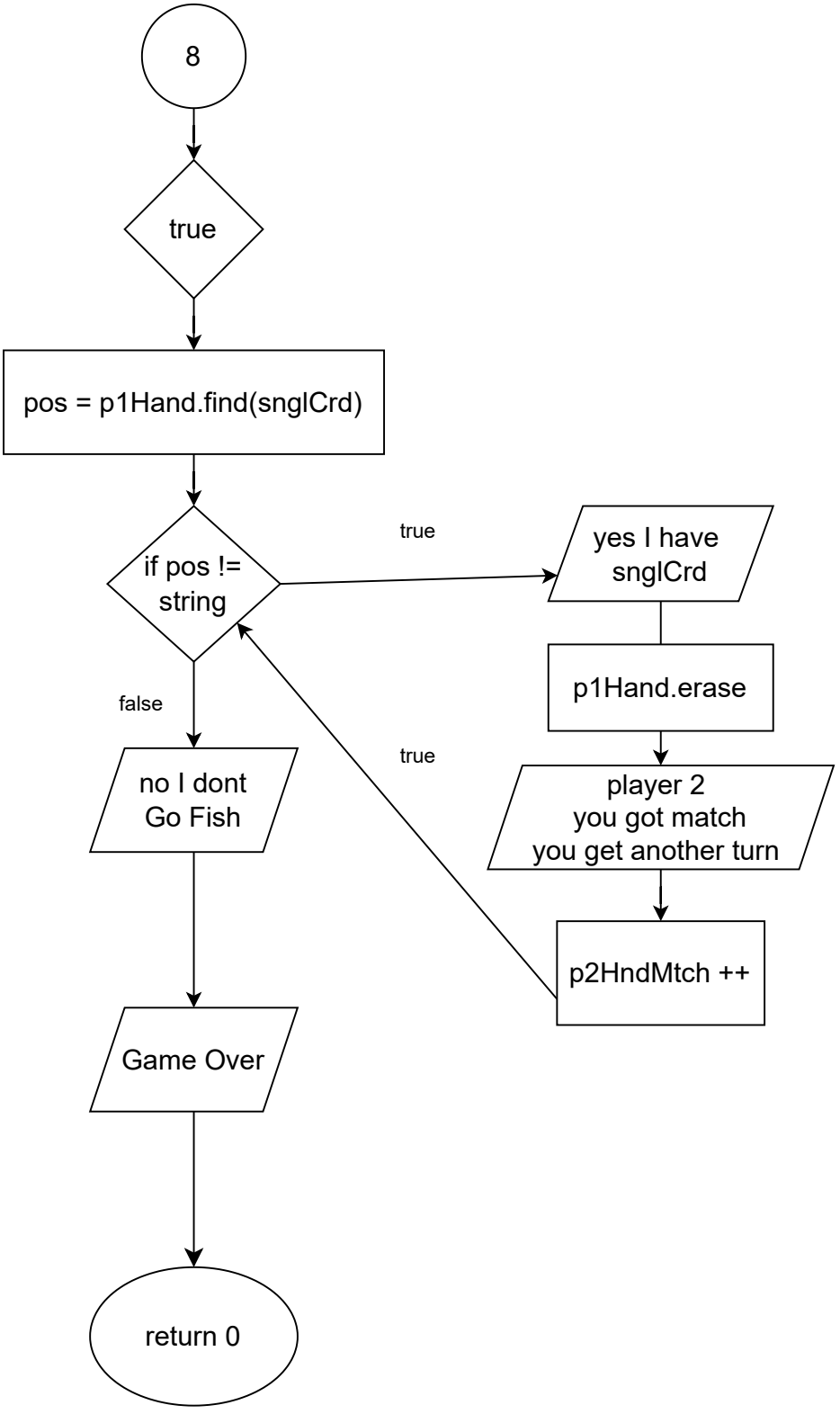
Flowchart


4
deckSize > 0
true
players == plyr1
true
player 1 your turn
false
players == plyr2
true
player 2 your turn
switch(players) case1
initialize i
i < p1Hand;
++i
5
5
card = p1Hand
initialize j
true
false
j < p1Hand length
++j
if card == p1Hand
p1HandMtch ++
p1Hand.erase
p1HndMtch
6

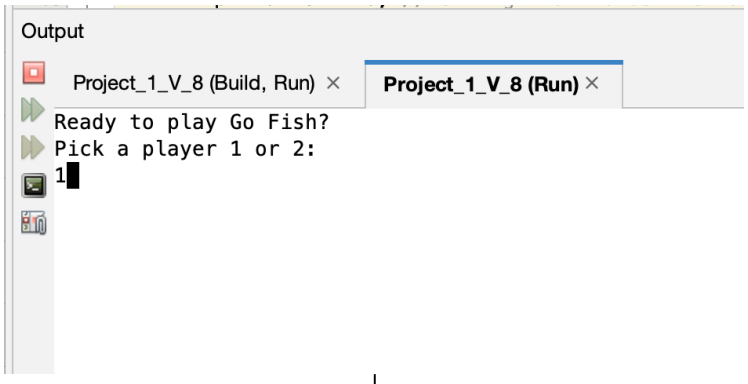
Flowchart

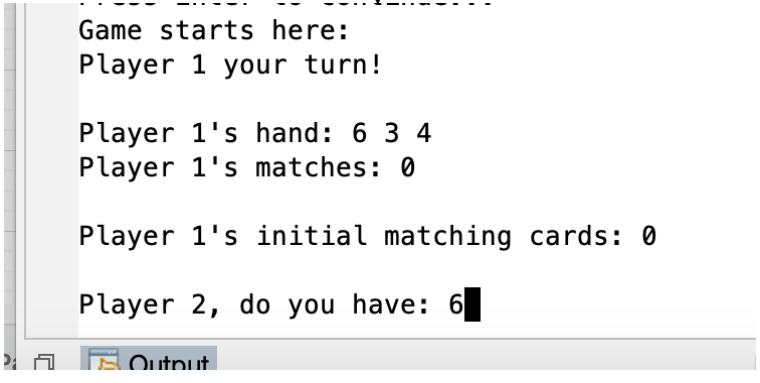

6
true
pos = p2Hand.find(snglCrd)
if pos != string
true
yes I have snglCrd
p2Hand.erase
player 1 you got match you get another turn
p1HndMtch ++
true
false
no I dont Go Fish
switch(players) case 2
initialize i
false
true
i < p2Hand
true
++i
7
7
card = p2Hand
initialize j
j < p2Hand length
true
++j
false
if card == p2Hand
true
p2HndMtc ++
p2Hand.erase
p2HandMtch
8

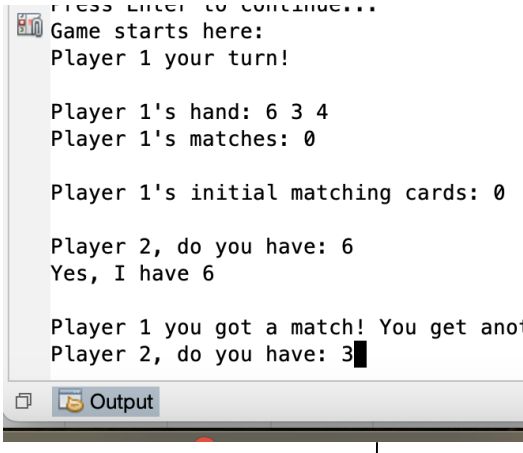
Flowchart

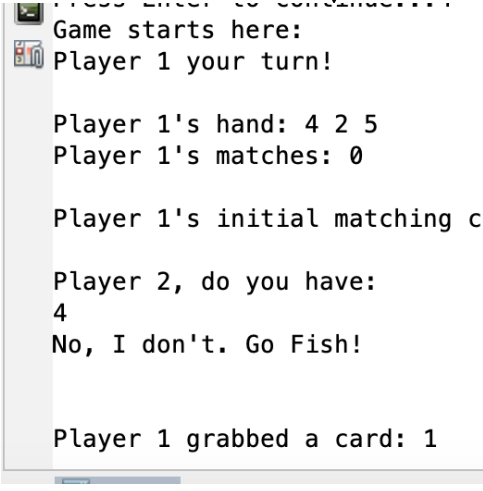

8
true
pos = p1Hand.find(sngICrd)
if pos != string
true
yes I have sngICrd
p1Hand.erase
player 2 you got match you get another turn
p2HndMtch ++
true
false
no I dont Go Fish
Game Over
return 0

Input Examples

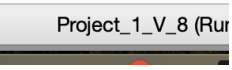








ther turn!



ards: 0