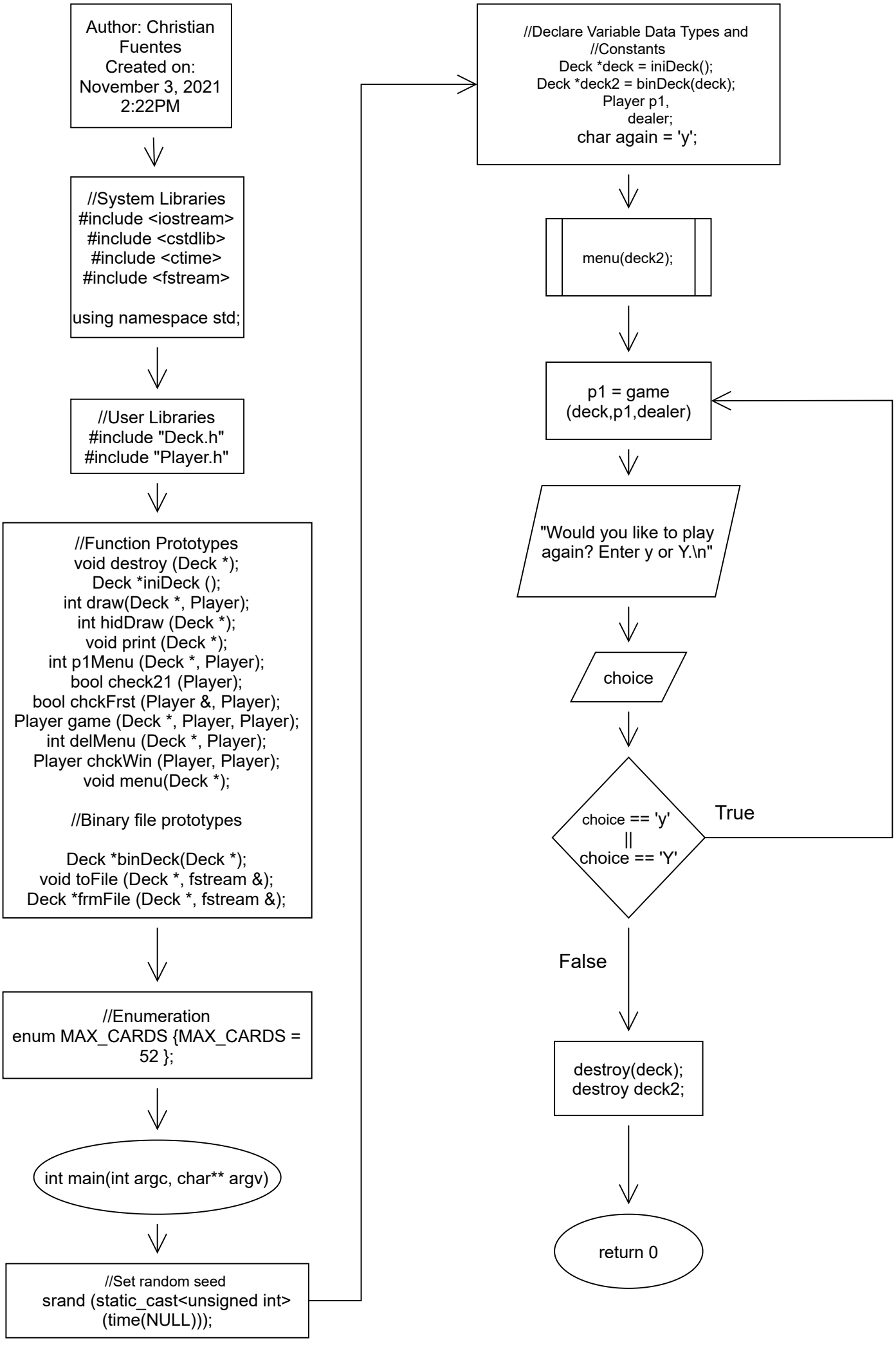
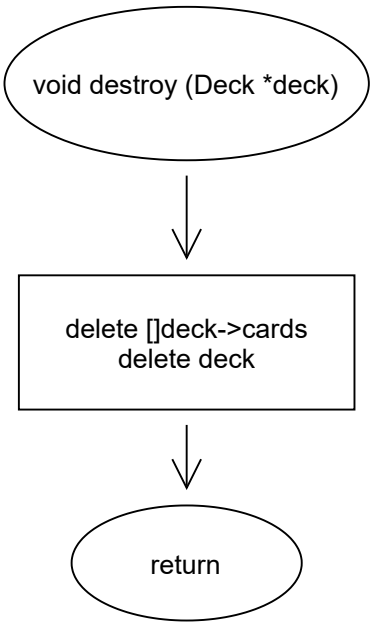
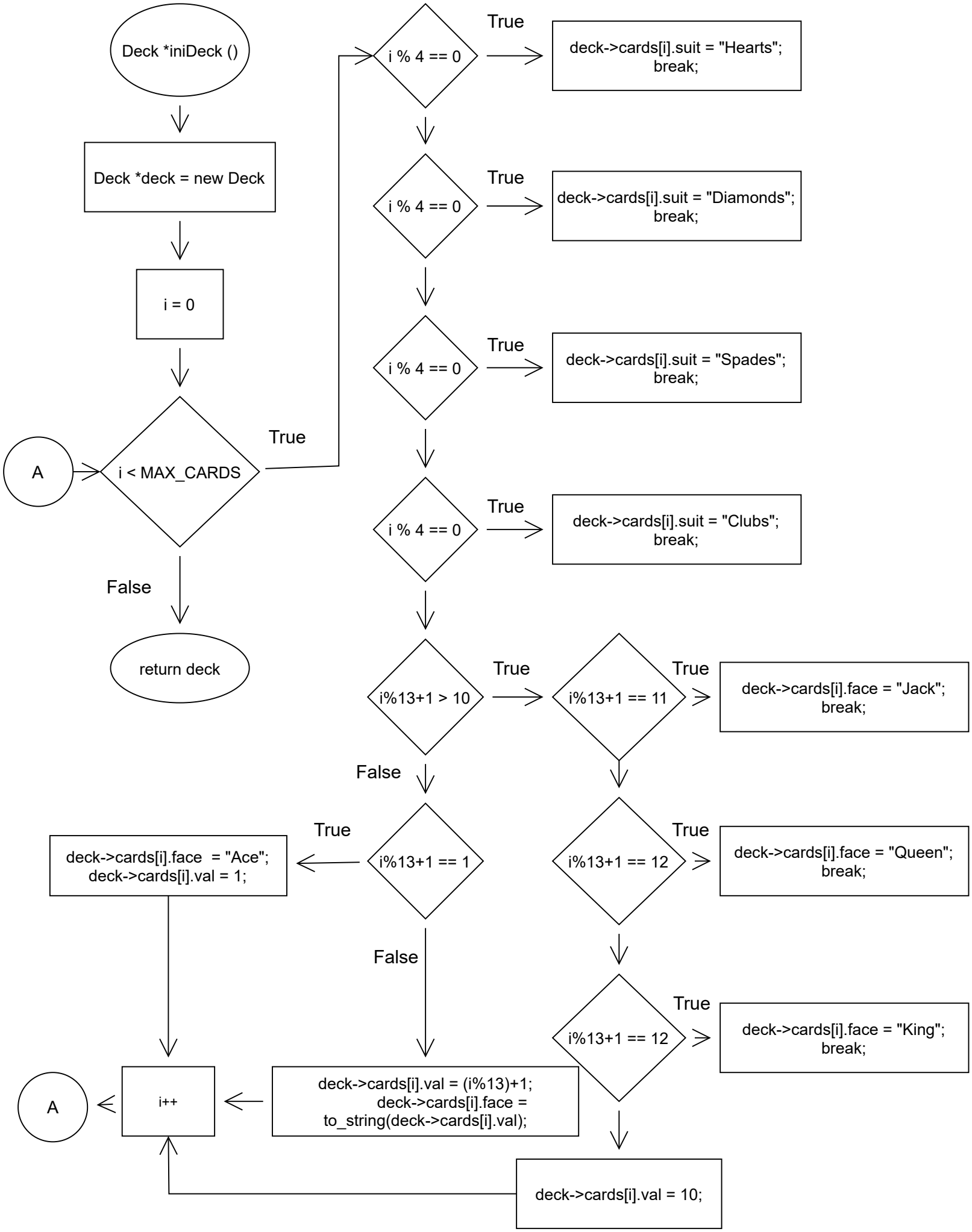
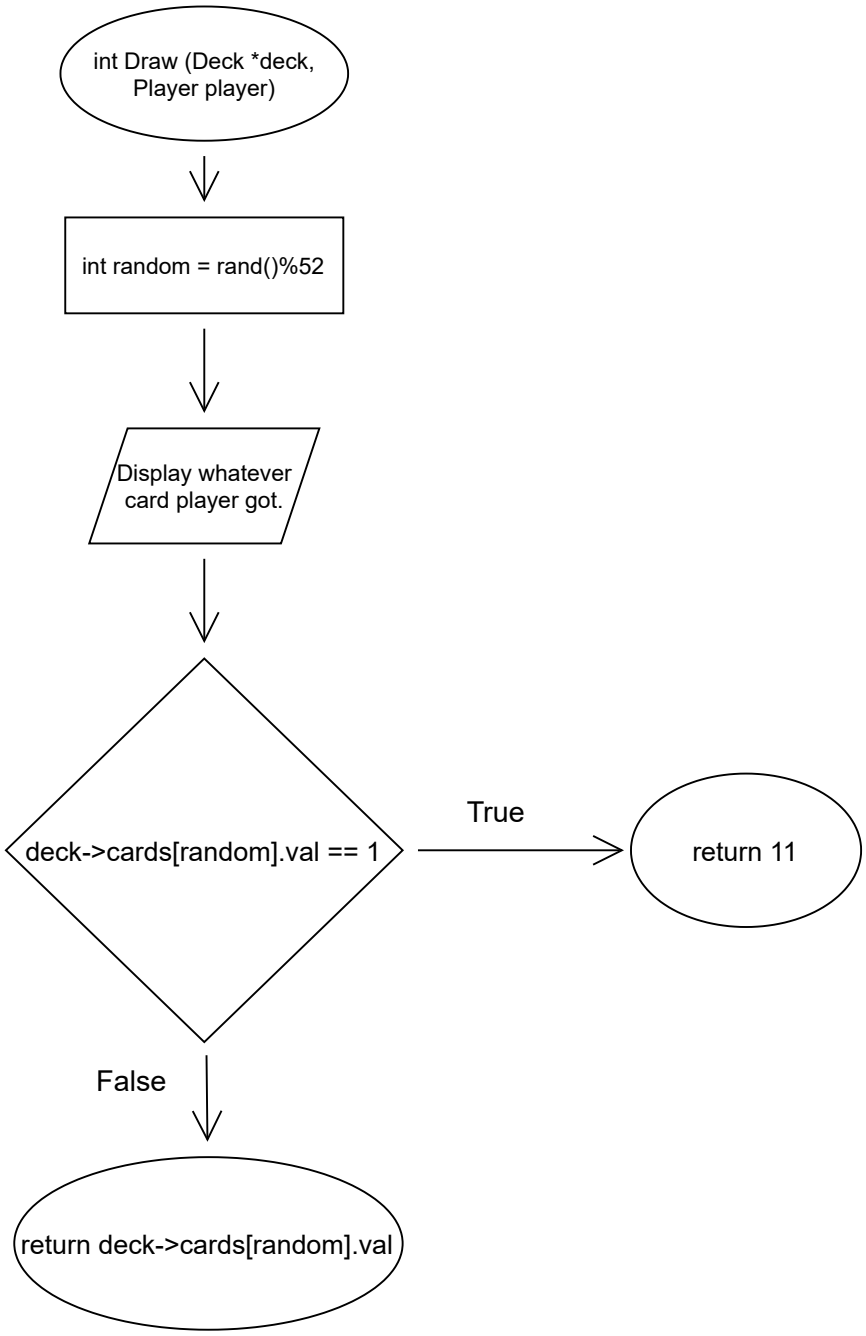
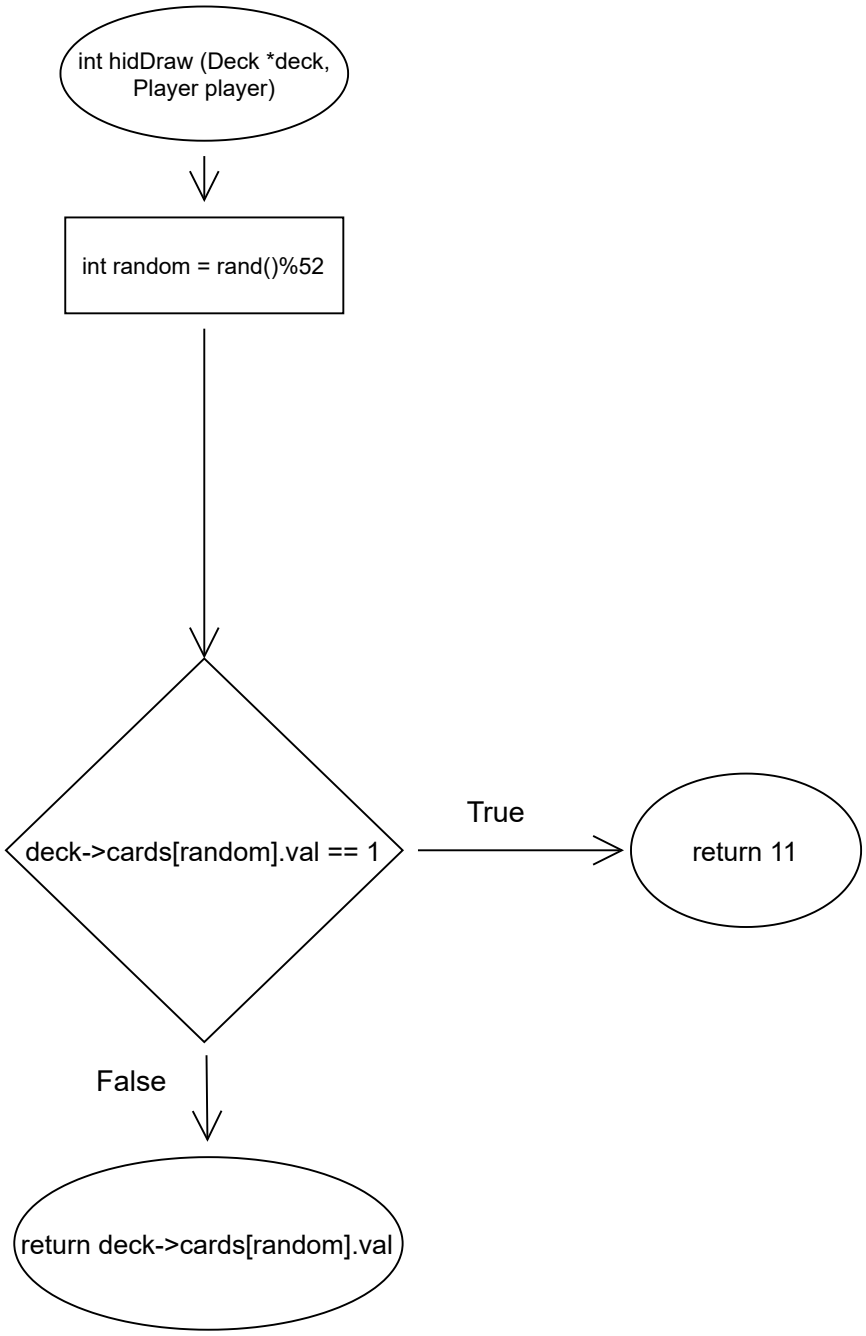
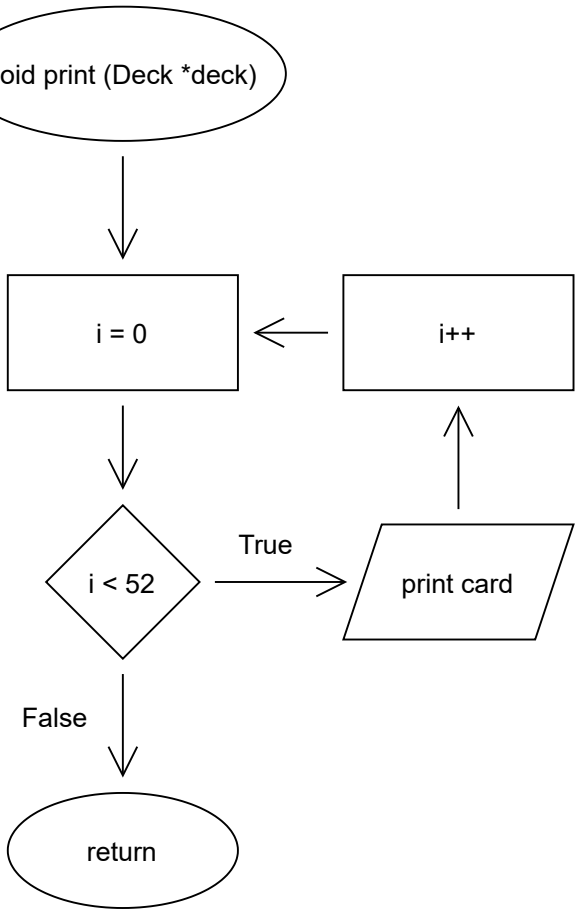

Author: Christian Fuentes
Created on: November 3, 2021 2:22PM
//System Libraries
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <fstream>
using namespace std;
//User Libraries
#include "Deck.h"
#include "Player.h"
//Function Prototypes
void destroy (Deck *);
Deck *iniDeck ();
int draw(Deck *, Player);
int hidDraw (Deck *);
void print (Deck *);
int p1Menu (Deck *, Player);
bool check21 (Player);
bool chckFrst (Player &, Player);
Player game (Deck *, Player, Player);
int delMenu (Deck *, Player);
Player chckWin (Player, Player);
void menu(Deck *);
//Binary file prototypes
Deck *binDeck(Deck *);
void toFile (Deck *, fstream &);
Deck *frmFile (Deck *, fstream &);
//Enumeration
enum MAX_CARDS {MAX_CARDS = 52 };
int main(int argc, char** argv)
//Set random seed
srand (static_cast<unsigned int> (time(NULL)));
//Declare Variable Data Types and //Constants
Deck *deck = iniDeck();
Deck *deck2 = binDeck(deck);
Player p1,
dealer;
char again = 'y';
menu(deck2);
p1 = game (deck,p1,dealer)
"Would you like to play again? Enter y or Y.\n"
choice
choice == 'y' || choice == 'Y'
True
False
destroy(deck);
destroy deck2;
return 0

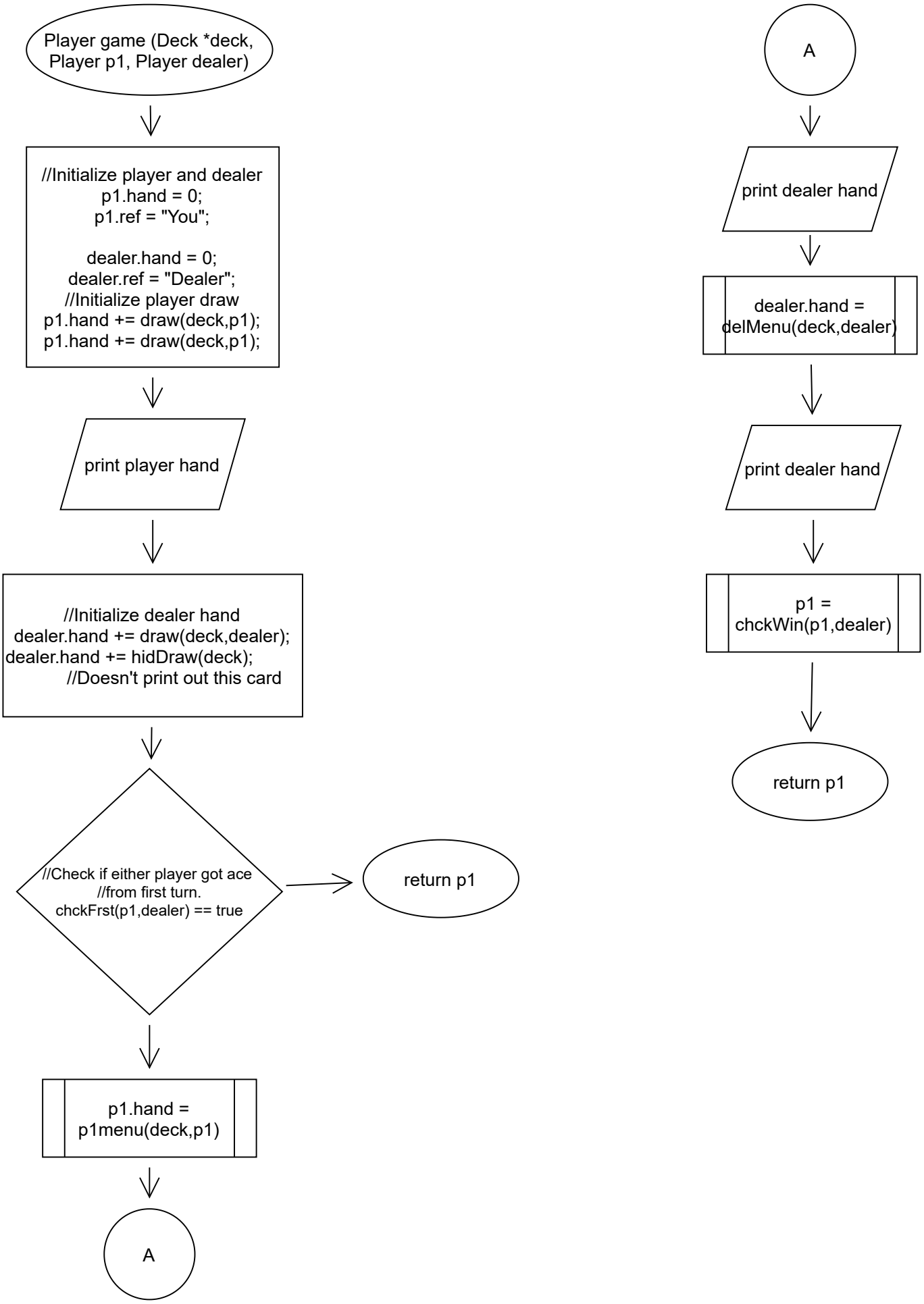

void destroy (Deck *deck)
delete []deck->cards
delete deck
return

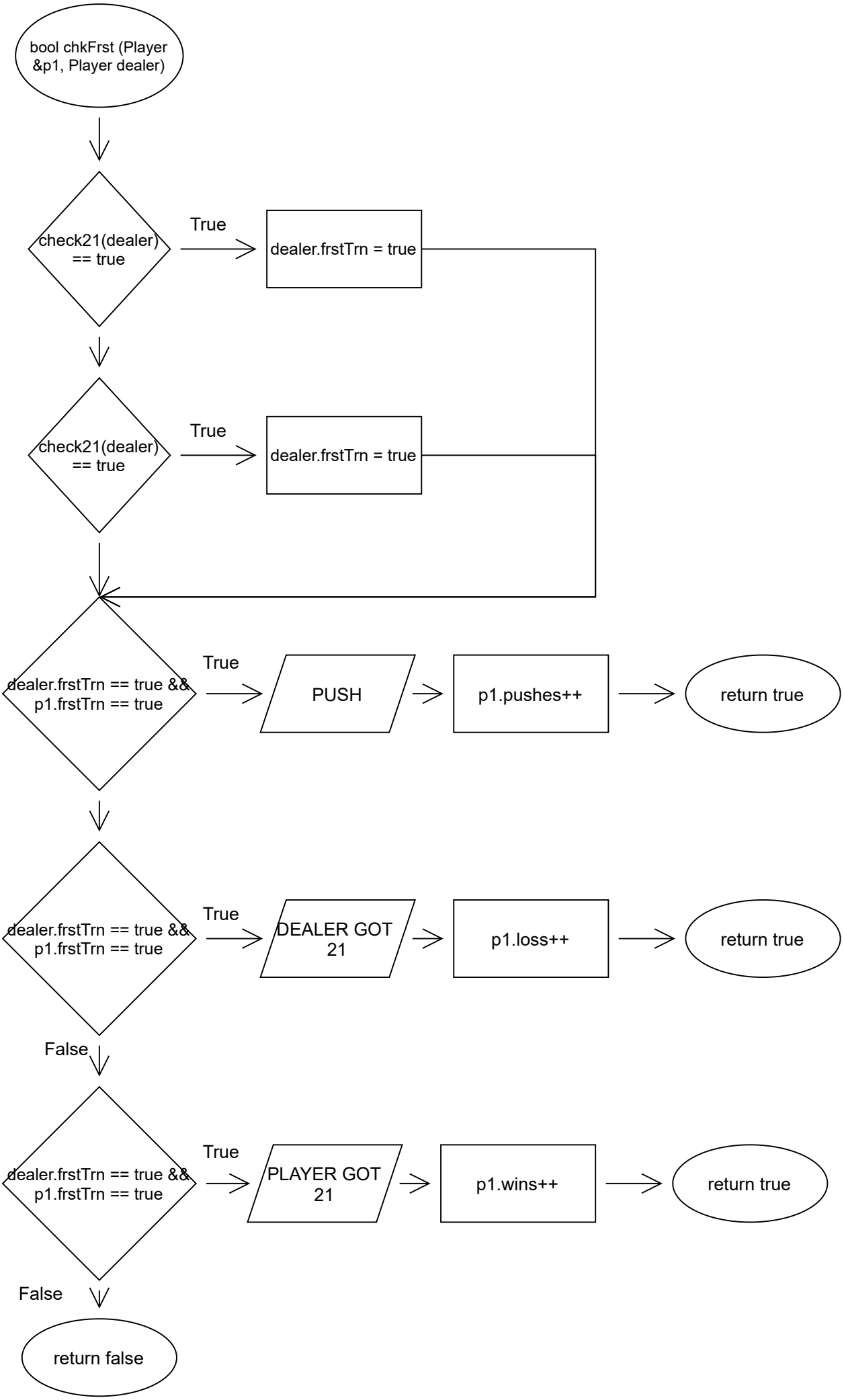

Deck *iniDeck ()
Deck *deck = new Deck
i = 0
A
i < MAX_CARDS
True
False
return deck
i % 4 == 0
True
deck->cards[i].suit = "Hearts";
break;
i % 4 == 0
True
deck->cards[i].suit = "Diamonds";
break;
i % 4 == 0
True
deck->cards[i].suit = "Spades";
break;
i % 4 == 0
True
deck->cards[i].suit = "Clubs";
break;
i%13+1 > 10
True
i%13+1 == 11
True
deck->cards[i].face = "Jack";
break;
False
i%13+1 == 1
True
deck->cards[i].face = "Ace";
deck->cards[i].val = 1;
i%13+1 == 12
True
deck->cards[i].face = "Queen";
break;
False
i%13+1 == 12
True
deck->cards[i].face = "King";
break;
deck->cards[i].val = (i%13)+1;
deck->cards[i].face =
to_string(deck->cards[i].val);
deck->cards[i].val = 10;
i++
A

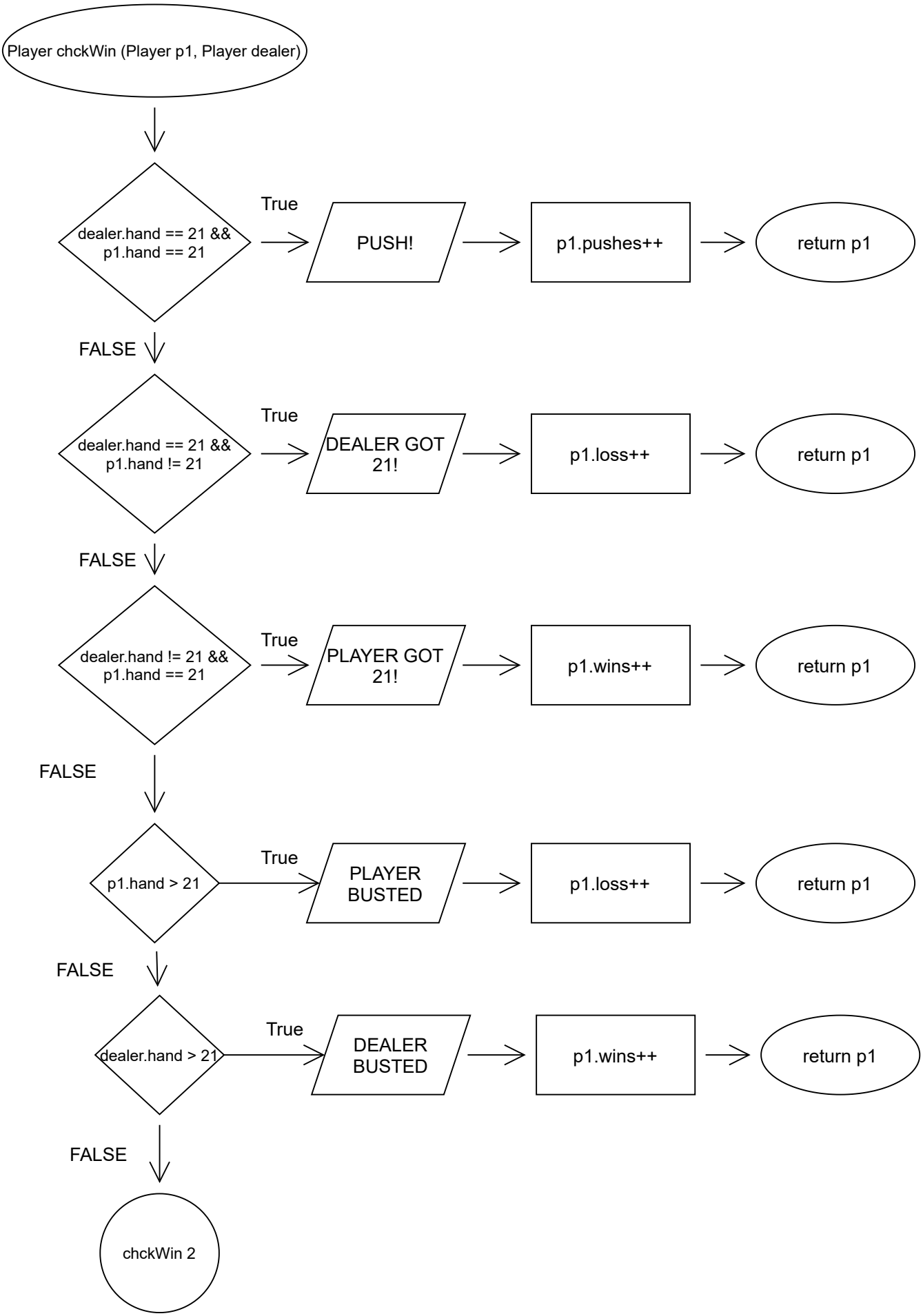

int Draw (Deck *deck, Player player)
int random = rand()%52
Display whatever card player got.
deck->cards[random].val == 1
True
return 11
False
return deck->cards[random].val

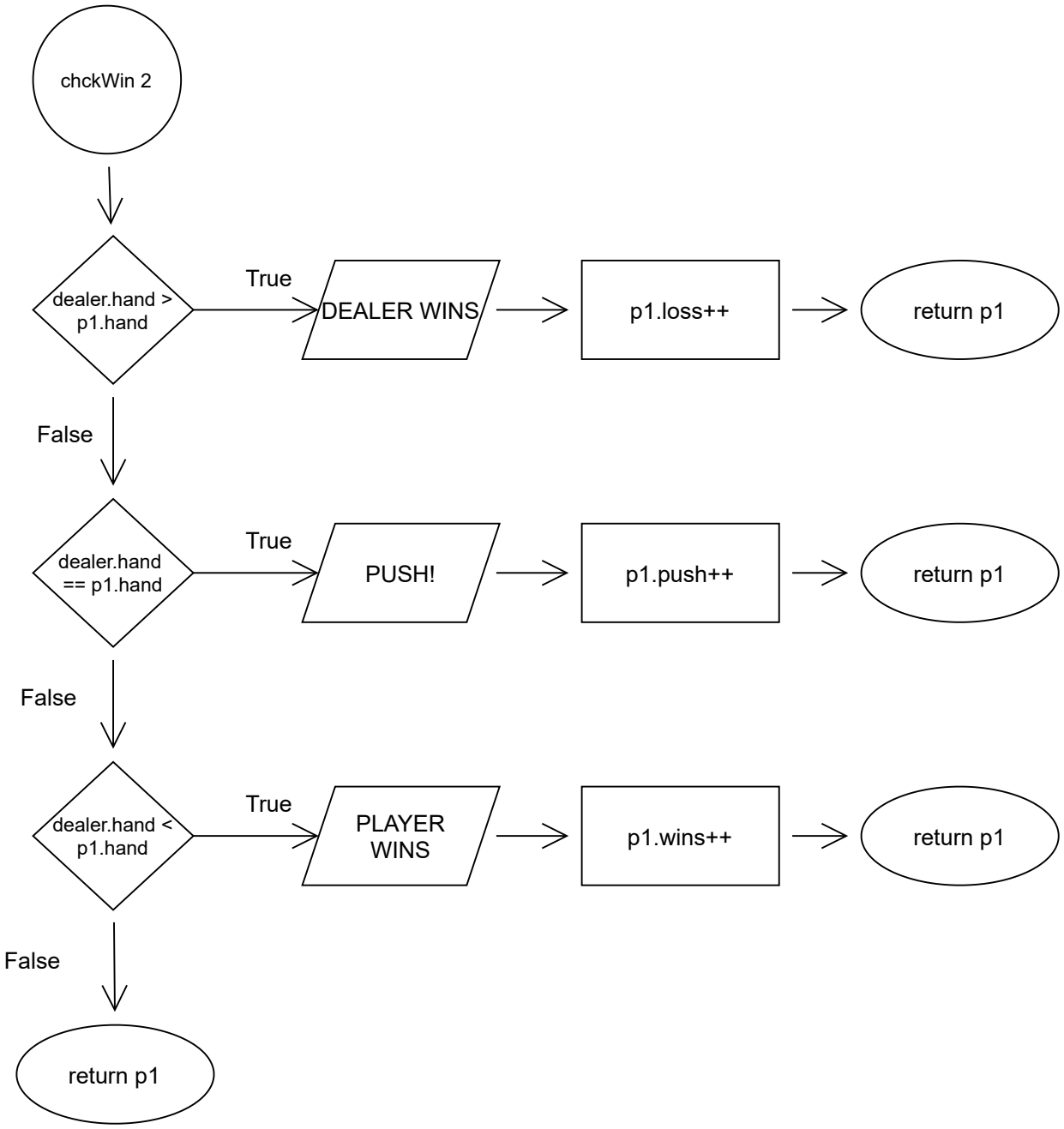

int hidDraw (Deck *deck,
Player player)
int random = rand()%52
deck->cards[random].val == 1
True
return 11
False
return deck->cards[random].val

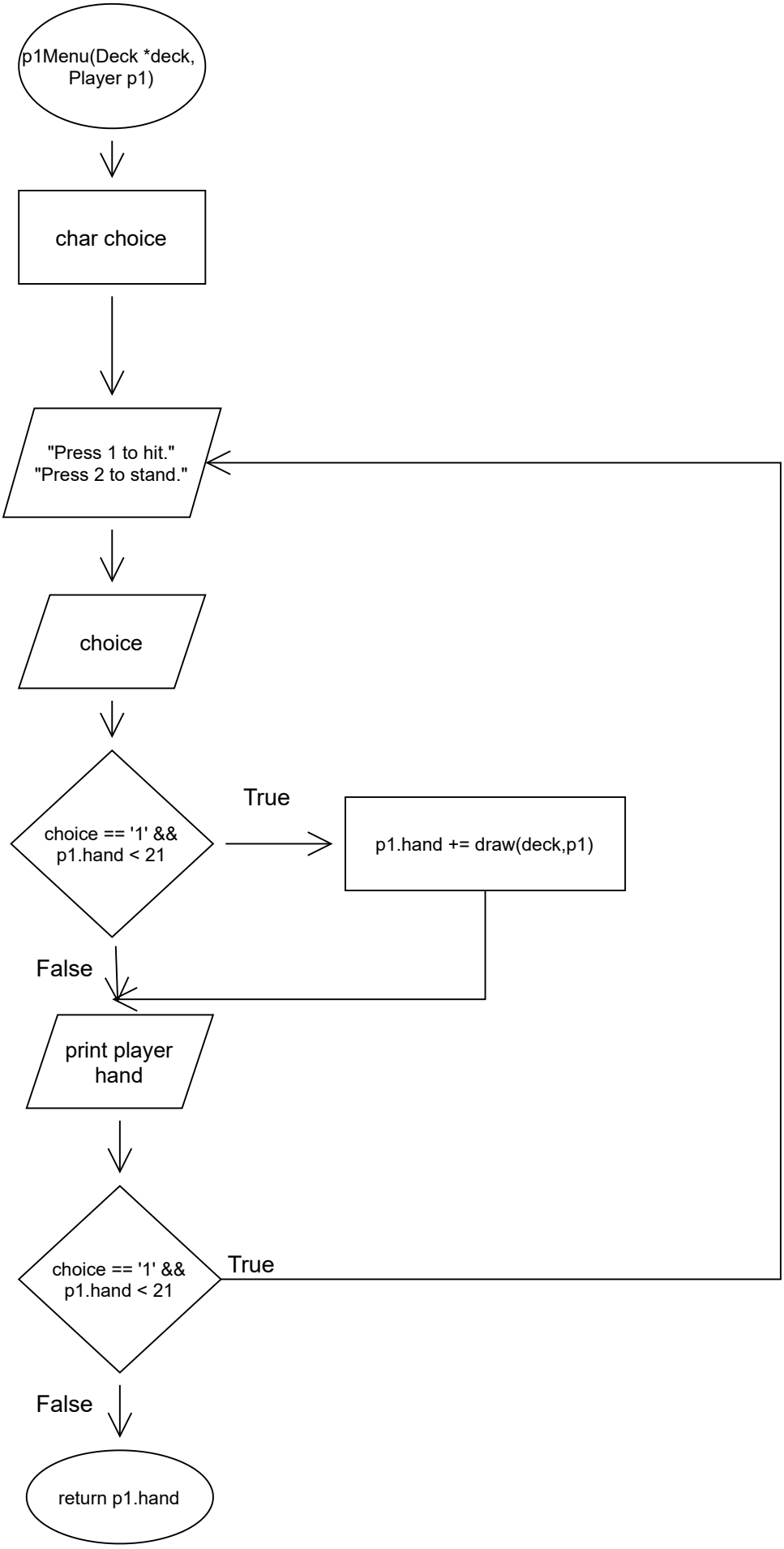

void print (Deck *deck)
i = 0
i++
i < 52
True
print card
False
return

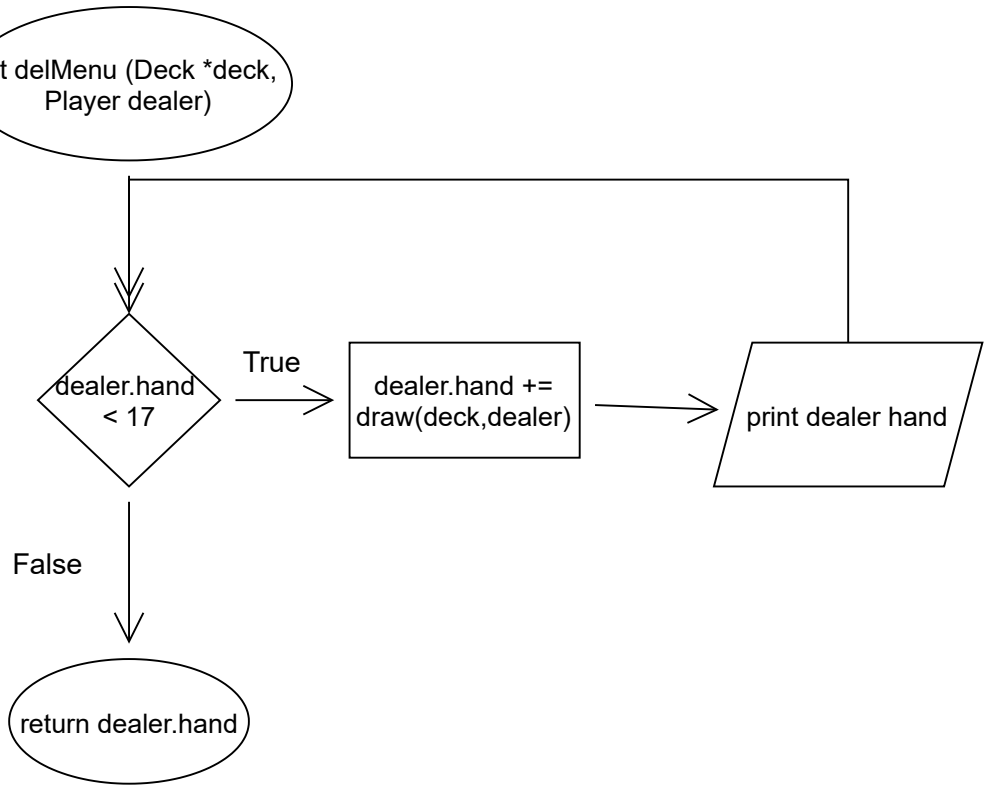

Player game (Deck *deck, Player p1, Player dealer)
//Initialize player and dealer
p1.hand = 0;
p1.ref = "You";
dealer.hand = 0;
dealer.ref = "Dealer";
//Initialize player draw
p1.hand += draw(deck,p1);
p1.hand += draw(deck,p1);
print player hand
//Initialize dealer hand
dealer.hand += draw(deck,dealer);
dealer.hand += hidDraw(deck);
//Doesn't print out this card
//Check if either player got ace
//from first turn.
chckFrst(p1,dealer) == true
return p1
p1.hand =
p1menu(deck,p1)
A
A
print dealer hand
dealer.hand =
delMenu(deck,dealer)
print dealer hand
p1 =
chckWin(p1,dealer)
return p1

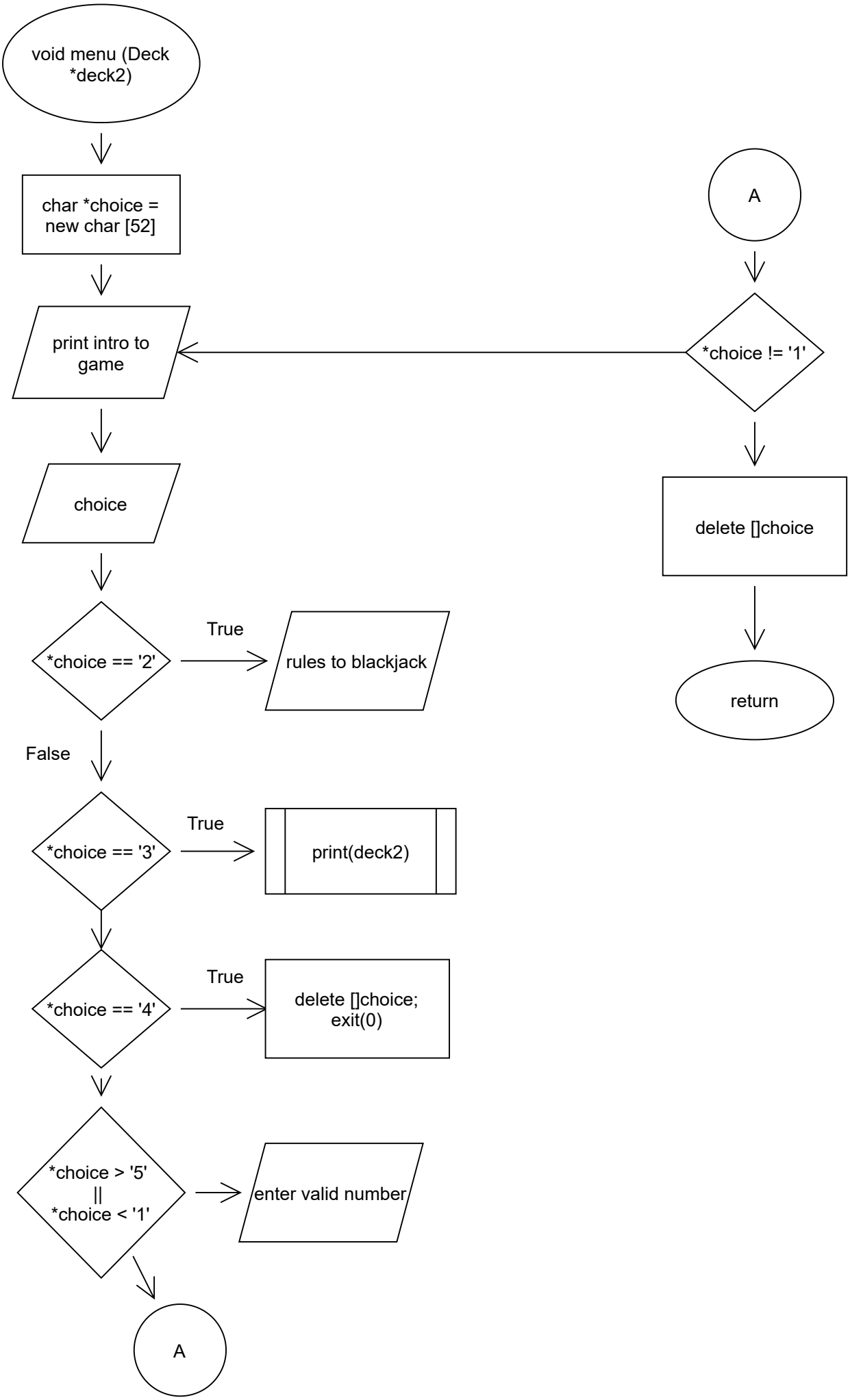

bool chkFrst (Player &p1, Player dealer)
check21(dealer) == true
True
dealer.frstTrn = true
check21(dealer) == true
True
dealer.frstTrn = true
dealer.frstTrn == true && p1.frstTrn == true
True
PUSH
p1.pushes++
return true
dealer.frstTrn == true && p1.frstTrn == true
True
DEALER GOT 21
p1.loss++
return true
False
dealer.frstTrn == true && p1.frstTrn == true
True
PLAYER GOT 21
p1.wins++
return true
False
return false

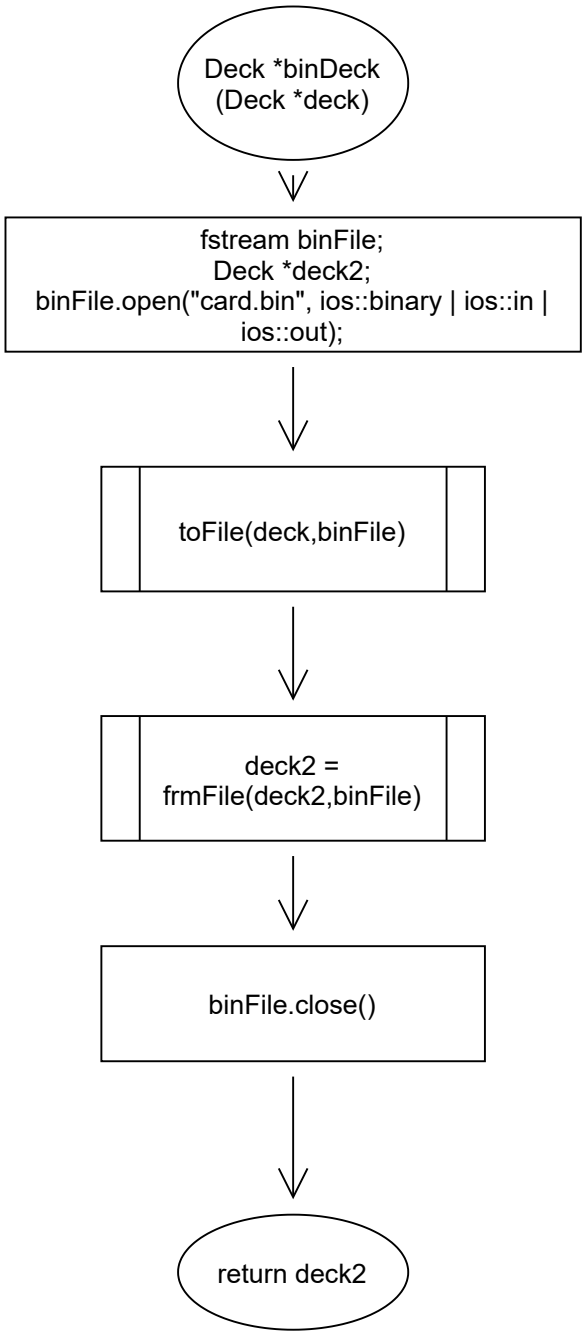

Player chckWin (Player p1, Player dealer)
dealer.hand == 21 && p1.hand == 21
True
PUSH!
p1.pushes++
return p1
FALSE
dealer.hand == 21 && p1.hand != 21
True
DEALER GOT 21!
p1.loss++
return p1
FALSE
dealer.hand != 21 && p1.hand == 21
True
PLAYER GOT 21!
p1.wins++
return p1
FALSE
p1.hand > 21
True
PLAYER BUSTED
p1.loss++
return p1
FALSE
dealer.hand > 21
True
DEALER BUSTED
p1.wins++
return p1
FALSE
chckWin 2

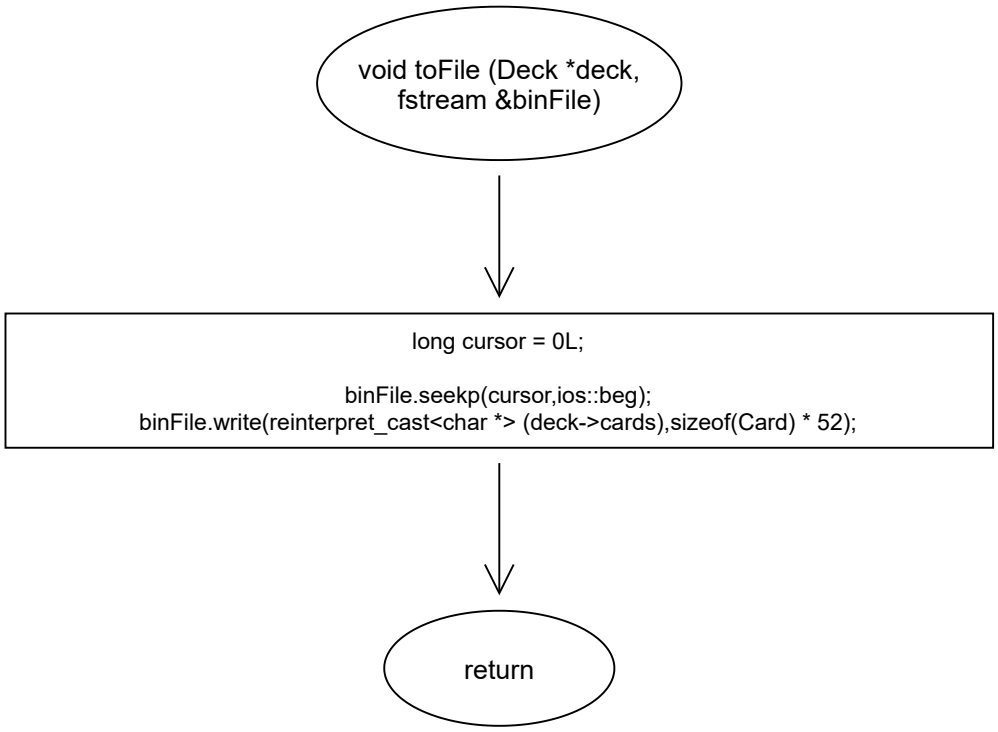

chckWin 2
dealer.hand > p1.hand
True
DEALER WINS
p1.loss++
return p1
False
dealer.hand == p1.hand
True
PUSH!
p1.push++
return p1
False
dealer.hand < p1.hand
True
PLAYER WINS
p1.wins++
return p1
False
return p1

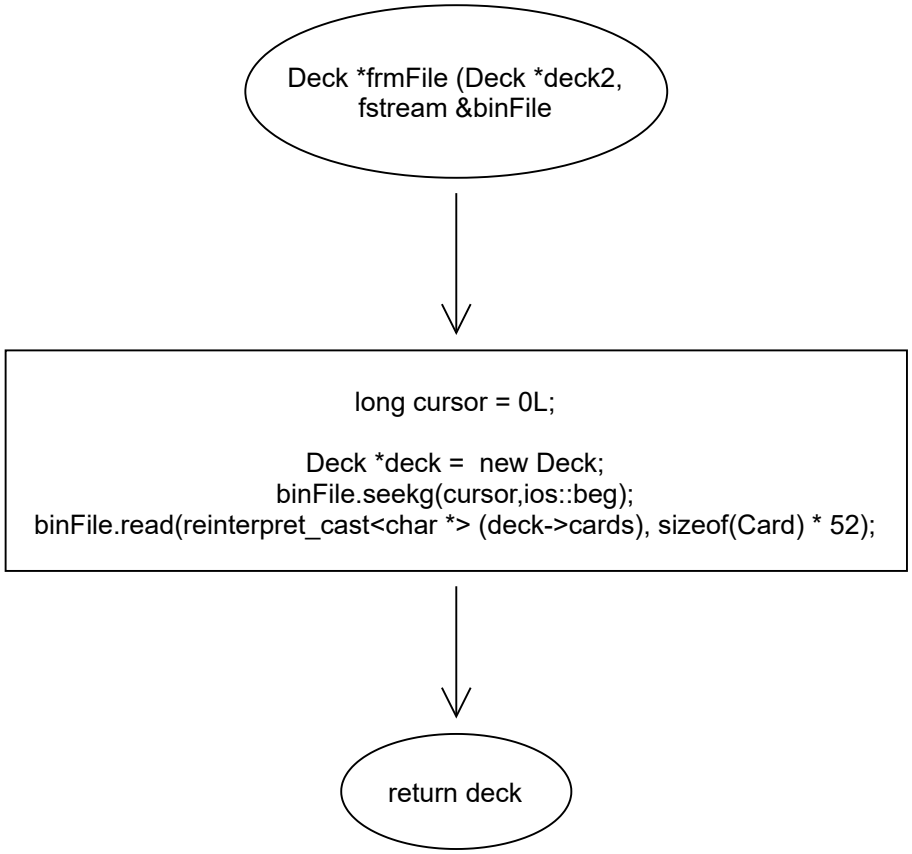

p1Menu(Deck *deck, Player p1)
char choice
"Press 1 to hit." "Press 2 to stand."
choice
choice == '1' && p1.hand < 21
True
p1.hand += draw(deck,p1)
False
print player hand
choice == '1' && p1.hand < 21
True
False
return p1.hand


int delMenu (Deck *deck,
Player dealer)
dealer.hand
< 17
True
dealer.hand +=
draw(deck,dealer)
print dealer hand
False
return dealer.hand


void menu (Deck *deck2)
char *choice = new char [52]
print intro to game
choice
*choice == '2'
True
rules to blackjack
False
*choice == '3'
True
print(deck2)
*choice == '4'
True
delete []choice; exit(0)
*choice > '5' || *choice < '1'
enter valid number
A
A
*choice != '1'
delete []choice
return


Deck *binDeck
(Deck *deck)
fstream binFile;
Deck *deck2;
binFile.open("card.bin", ios::binary | ios::in |
ios::out);
toFile(deck,binFile)
deck2 =
frmFile(deck2,binFile)
binFile.close()
return deck2


void toFile (Deck *deck,
fstream &binFile)
long cursor = 0L;
binFile.seekp(cursor,ios::beg);
binFile.write(reinterpret_cast<char *> (deck->cards),sizeof(Card) * 52);
return


Deck *frmFile (Deck *deck2,
fstream &binFile
long cursor = 0L;
Deck *deck = new Deck;
binFile.seekg(cursor,ios::beg);
binFile.read(reinterpret_cast<char *> (deck->cards), sizeof(Card) * 52);
return deck