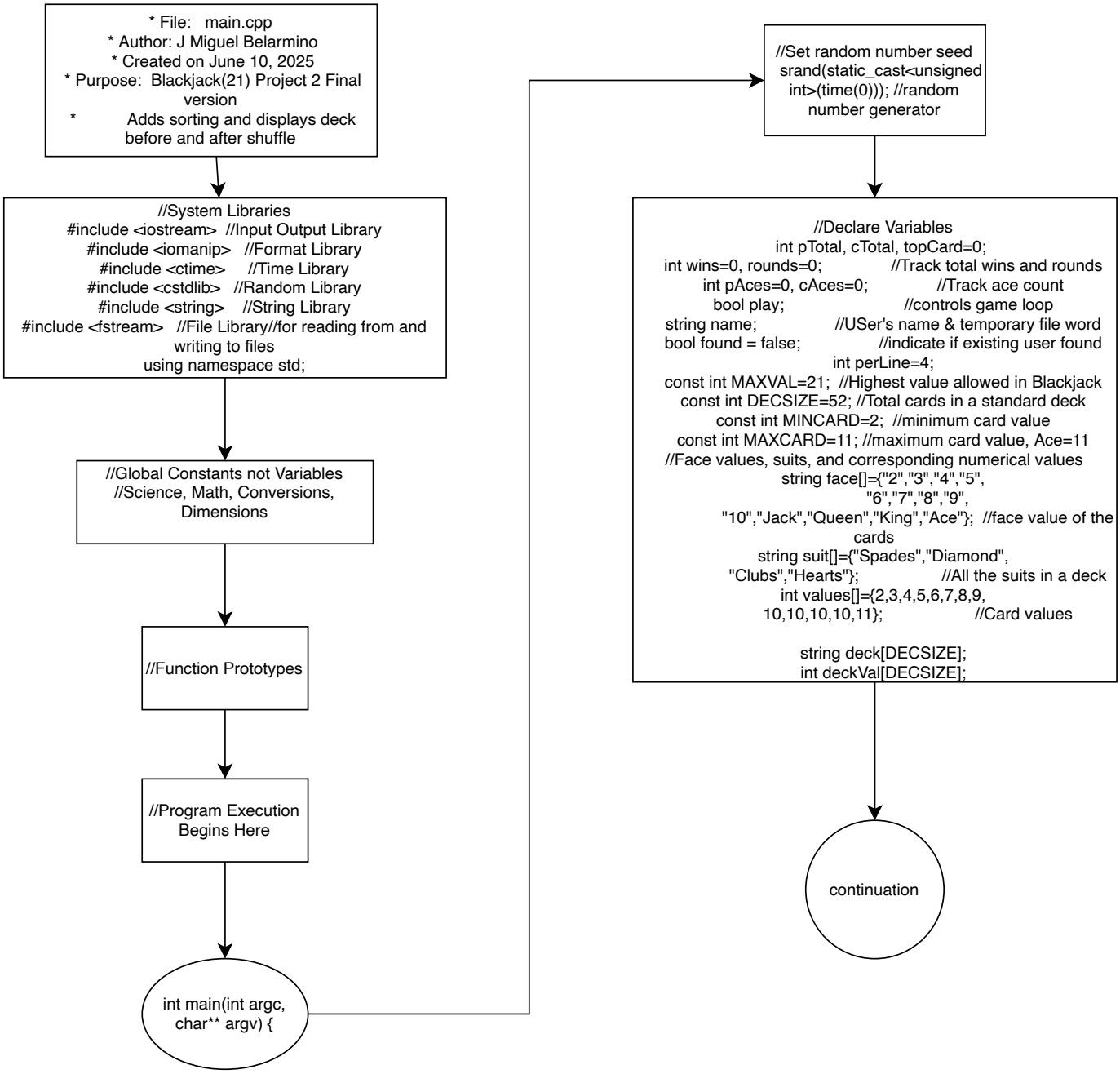
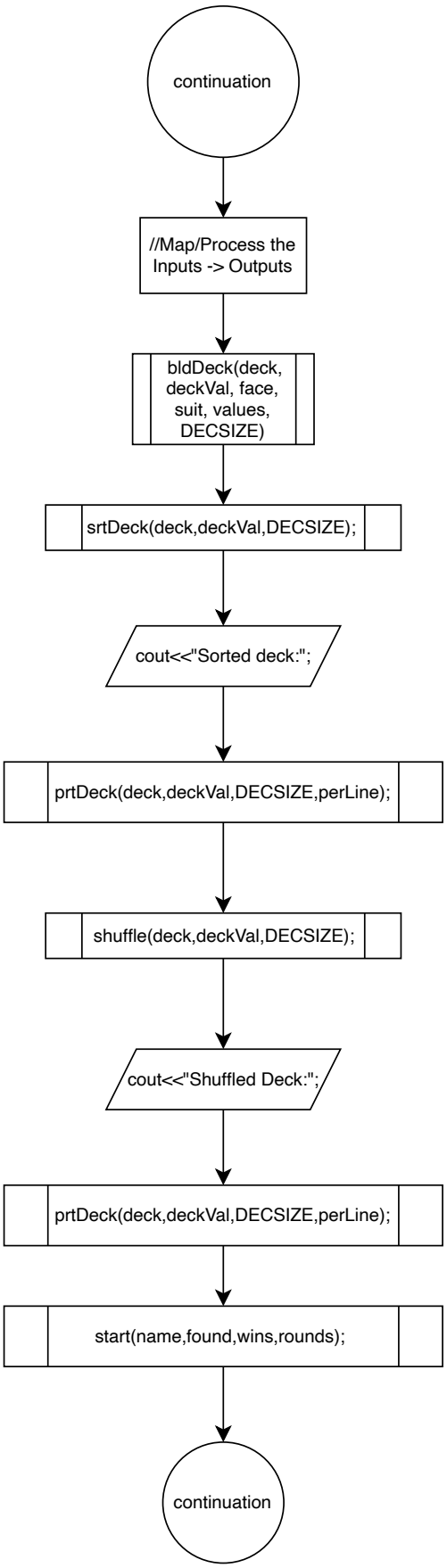
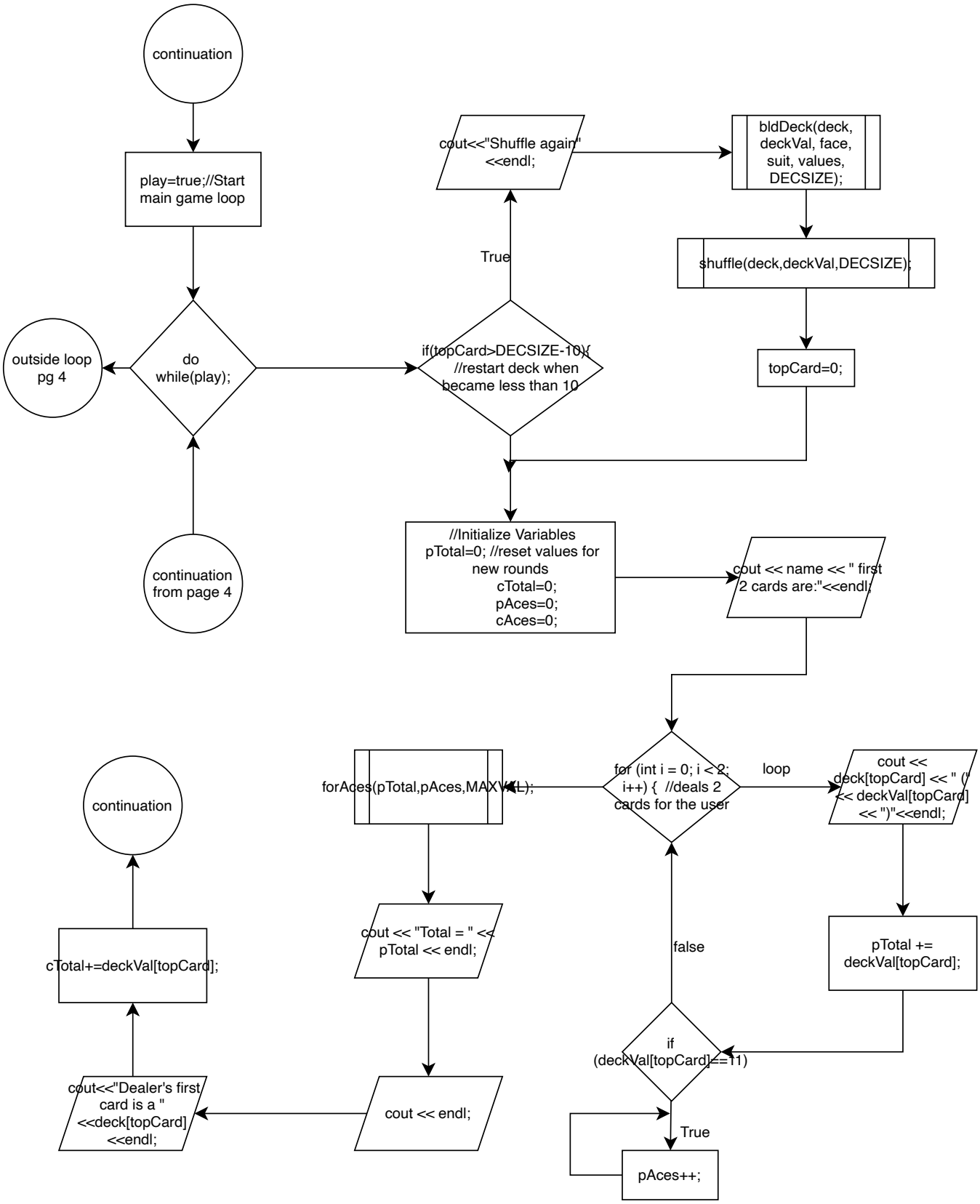
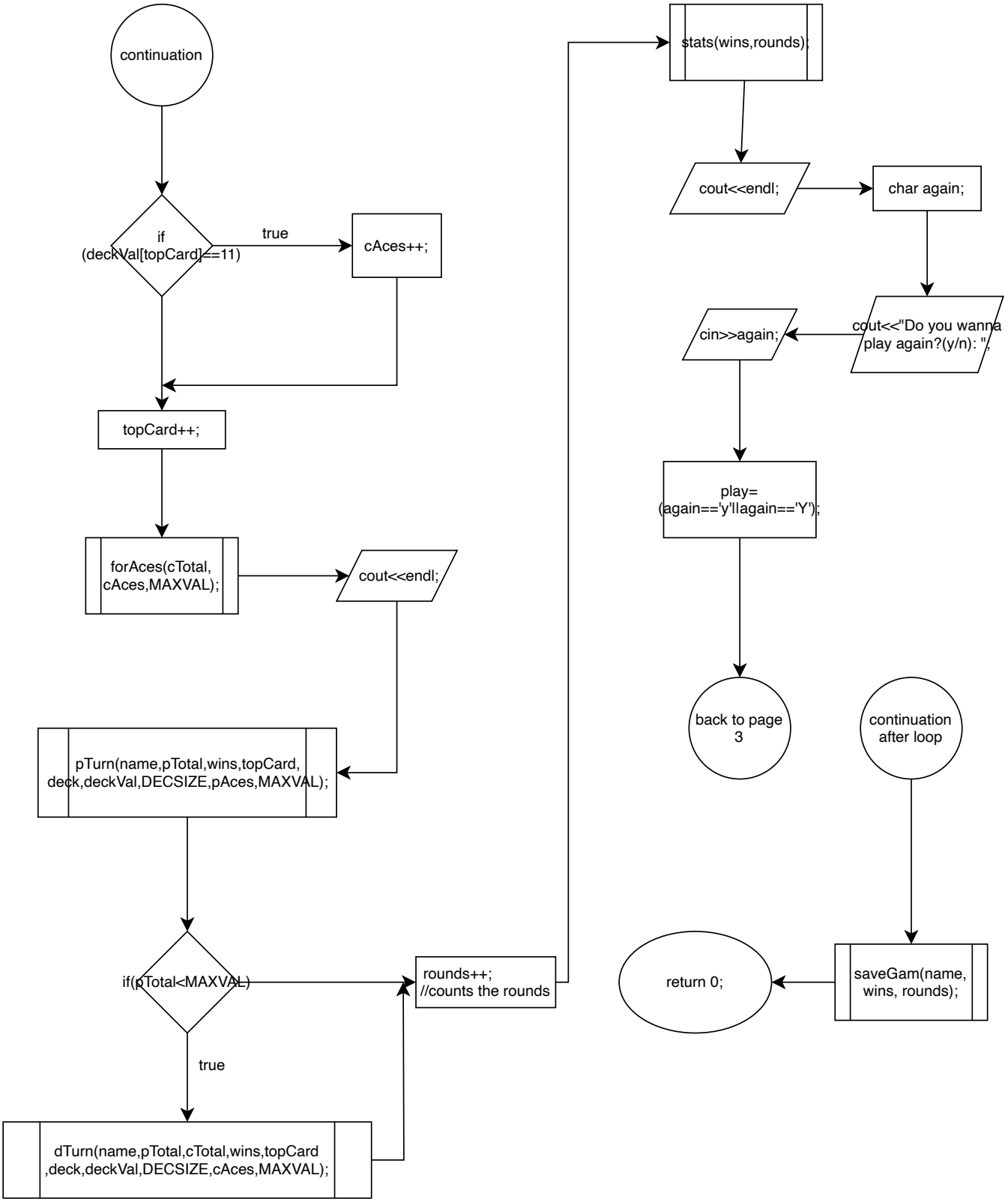
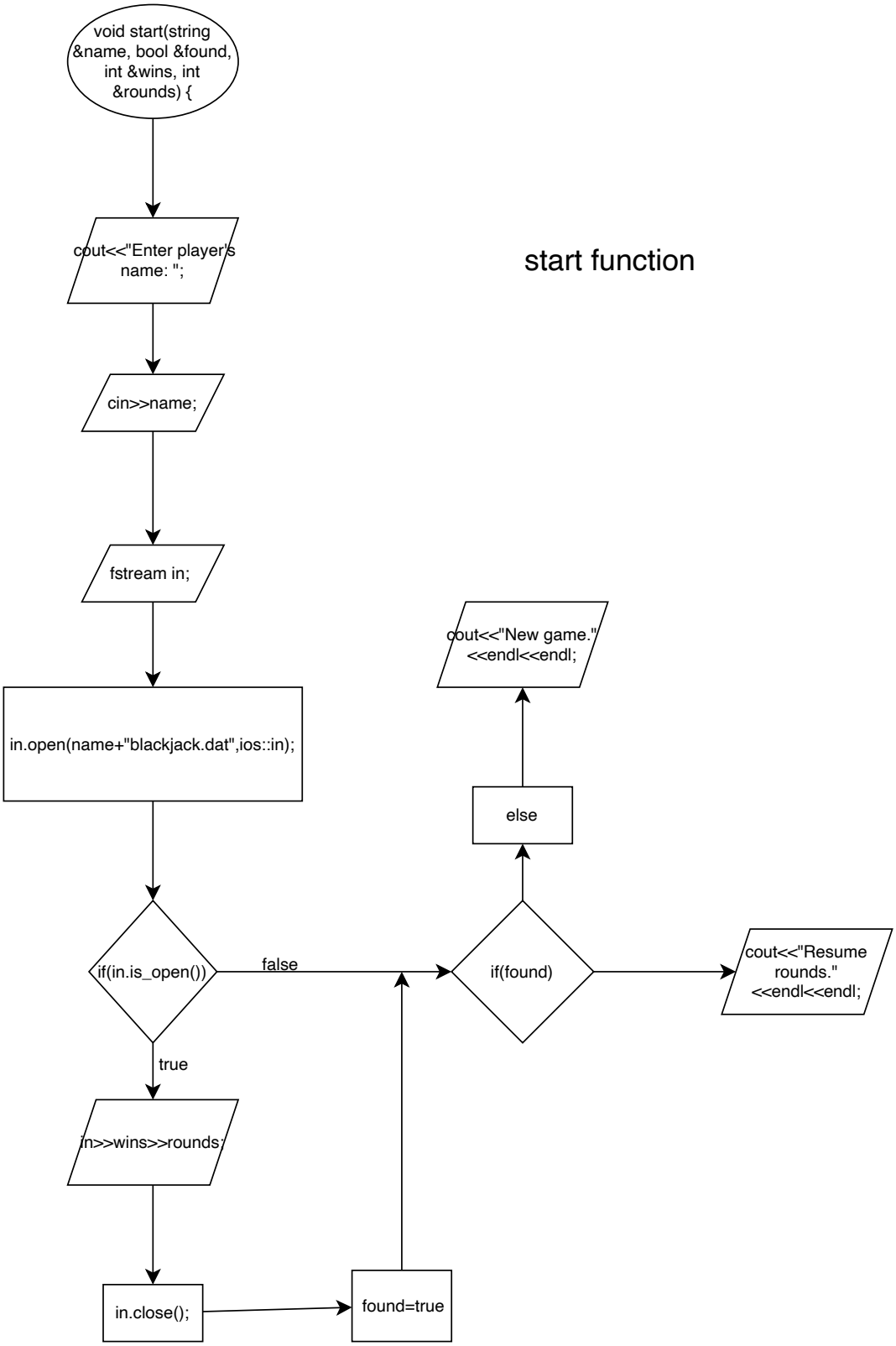

* File: main.cpp
* Author: J Miguel Belarmino
* Created on June 10, 2025
* Purpose: Blackjack(21) Project 2 Final version
* Adds sorting and displays deck before and after shuffle
//System Libraries
#include <iostream> //Input Output Library
#include <iomanip> //Format Library
#include <ctime> //Time Library
#include <cstdlib> //Random Library
#include <string> //String Library
#include <fstream> //File Library//for reading from and writing to files
using namespace std;
//Global Constants not Variables
//Science, Math, Conversions, Dimensions
//Function Prototypes
//Program Execution Begins Here
int main(int argc, char** argv) {
//Set random number seed
srand(static_cast<unsigned int>(time(0))); //random number generator
//Declare Variables
int pTotal, cTotal, topCard=0;
int wins=0, rounds=0; //Track total wins and rounds
int pAces=0, cAces=0; //Track ace count
bool play; //controls game loop
string name; //USer's name & temporary file word
bool found = false; //indicate if existing user found
int perLine=4;
const int MAXVAL=21; //Highest value allowed in Blackjack
const int DECSIZE=52; //Total cards in a standard deck
const int MINCARD=2; //minimum card value
const int MAXCARD=11; //maximum card value, Ace=11
//Face values, suits, and corresponding numerical values
string face[]={"2","3","4","5",
"6","7","8","9",
"10","Jack","Queen","King","Ace"}; //face value of the cards
string suit[]={"Spades","Diamond",
"Clubs","Hearts"}; //All the suits in a deck
int values[]={2,3,4,5,6,7,8,9,
10,10,10,10,11}; //Card values
string deck[DECSIZE];
int deckVal[DECSIZE];
continuation


continuation
//Map/Process the Inputs -> Outputs
bldDeck(deck, deckVal, face, suit, values, DECSIZE)
srtDeck(deck,deckVal,DECSIZE);
cout<<"Sorted deck:";
prtDeck(deck,deckVal,DECSIZE,perLine);
shuffle(deck,deckVal,DECSIZE);
cout<<"Shuffled Deck:";
prtDeck(deck,deckVal,DECSIZE,perLine);
start(name,found,wins,rounds);
continuation

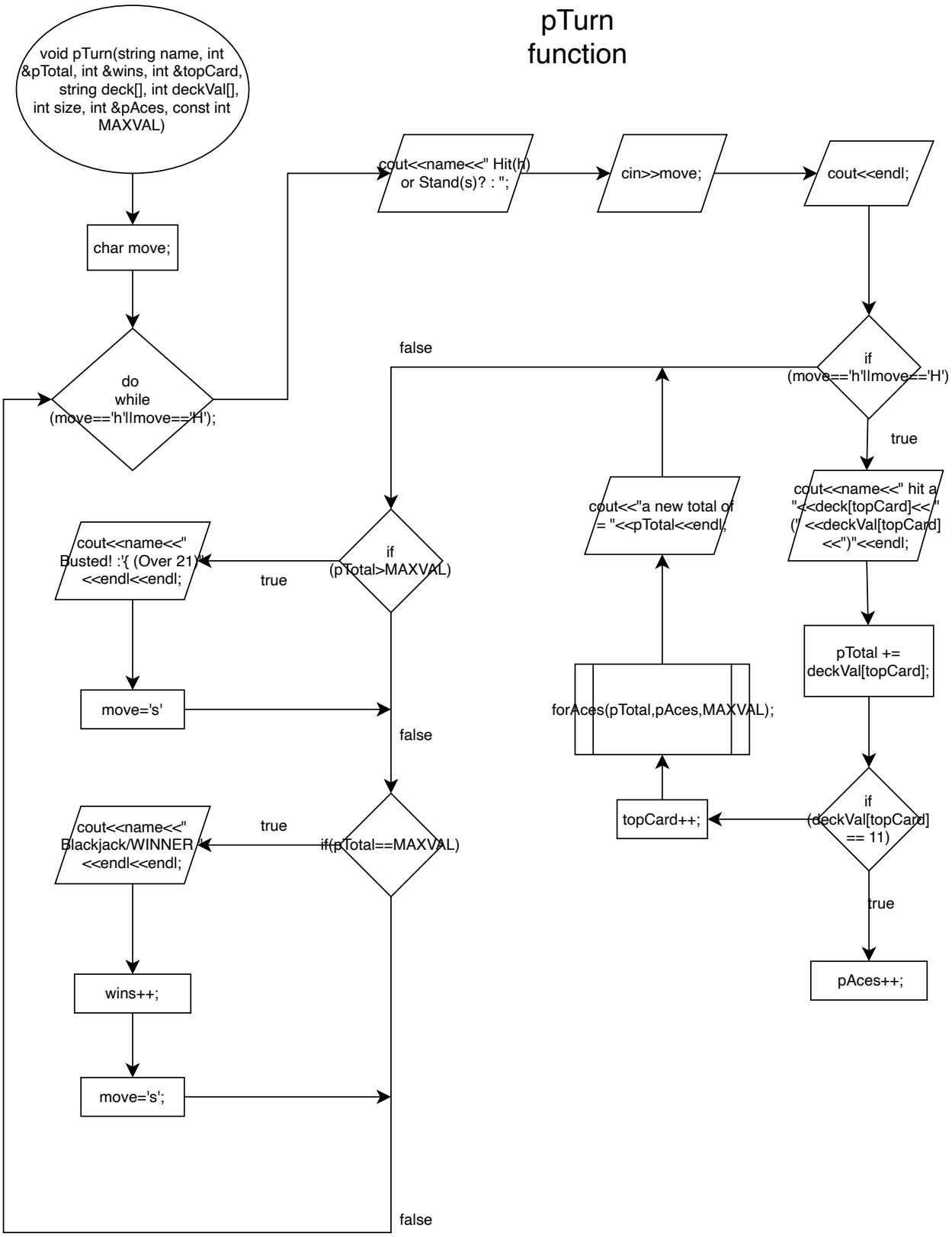

continuation
play=true;//Start main game loop
do while(play);
outside loop pg 4
continuation from page 4
cout<<"Shuffle again" <<endl;
True
if(topCard>DECSIZE-10){ //restart deck when became less than 10
bldDeck(deck, deckVal, face, suit, values, DECSIZE);
shuffle(deck,deckVal,DECSIZE);
topCard=0;
//Initialize Variables
pTotal=0; //reset values for new rounds
cTotal=0;
pAces=0;
cAces=0;
cout << name << " first 2 cards are:"<<endl;
for (int i = 0; i < 2; i++) { //deals 2 cards for the user
loop
cout << deck[topCard] << " (" << deckVal[topCard] << ")"<<endl;
pTotal += deckVal[topCard];
if (deckVal[topCard]==11)
false
True
pAces++;
forAces(pTotal,pAces,MAXVAL);
cout << "Total = " << pTotal << endl;
cout << endl;
cout<<"Dealer's first card is a " <<deck[topCard] <<endl;
cTotal+=deckVal[topCard];
continuation


continuation
if (deckVal[topCard]==11)
true
cAces++;
topCard++;
forAces(cTotal, cAces,MAXVAL);
cout<<endl;
pTurn(name,pTotal,wins,topCard, deck,deckVal,DECSIZE,pAces,MAXVAL);
if(pTotal<MAXVAL)
true
dTurn(name,pTotal,cTotal,wins,topCard ,deck,deckVal,DECSIZE,cAces,MAXVAL);
rounds++; //counts the rounds
stats(wins,rounds);
cout<<endl;
char again;
cout<<"Do you wanna play again?(y/n): ";
cin>>again;
play= (again=='y'||again=='Y');
back to page 3
continuation after loop
saveGam(name, wins, rounds);
return 0;

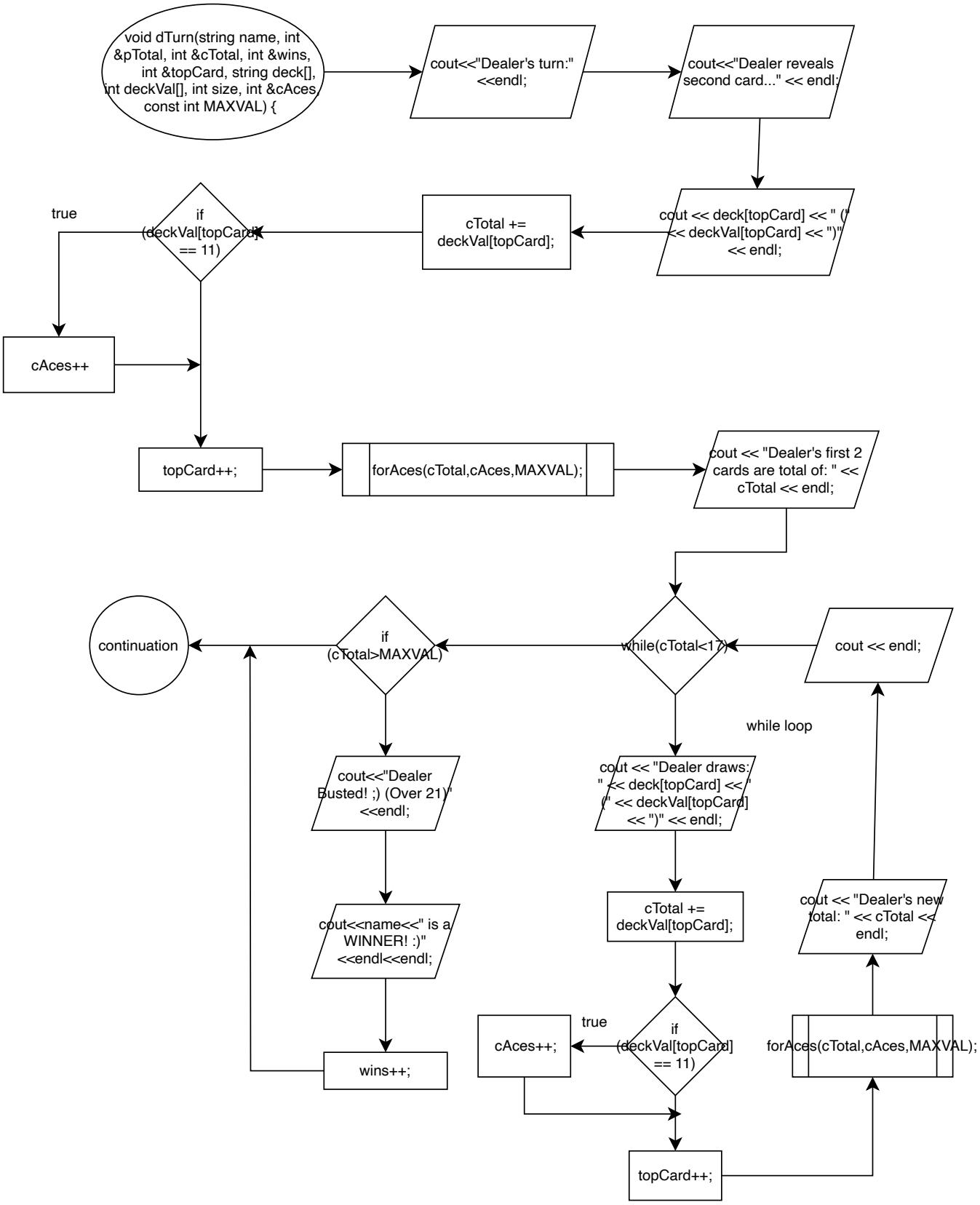
start function

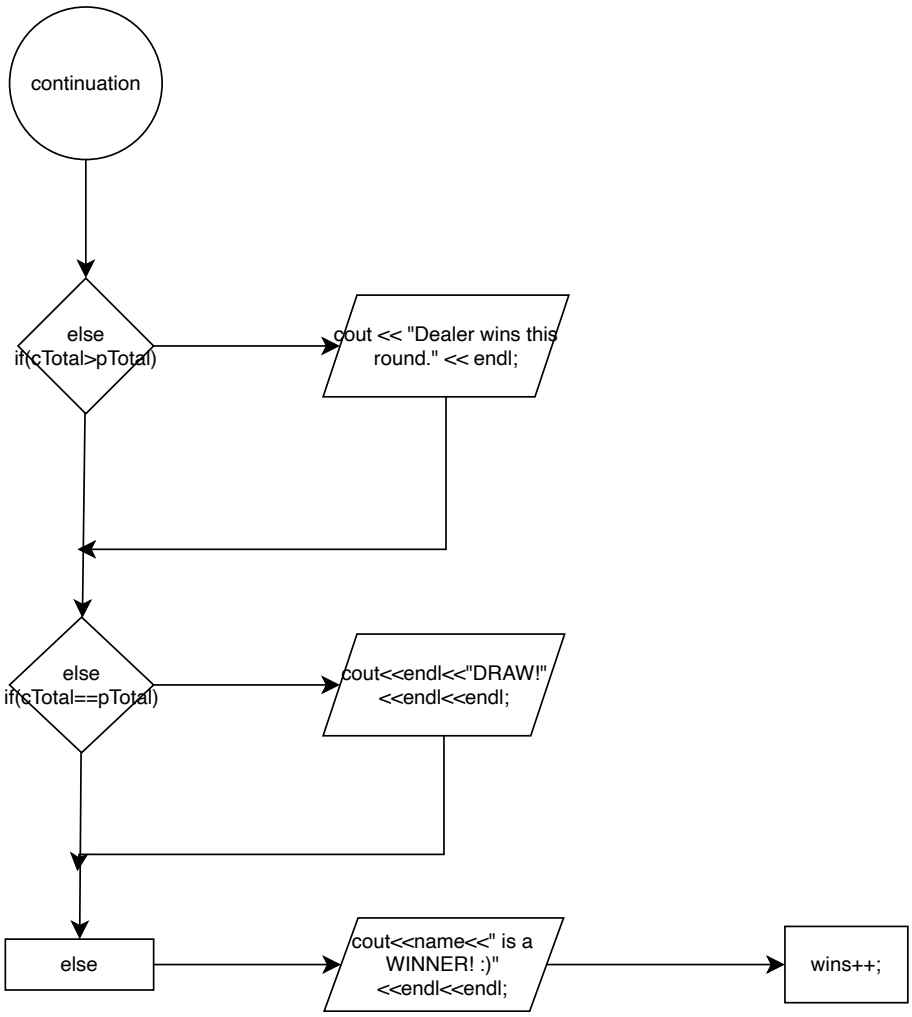


pTurn function

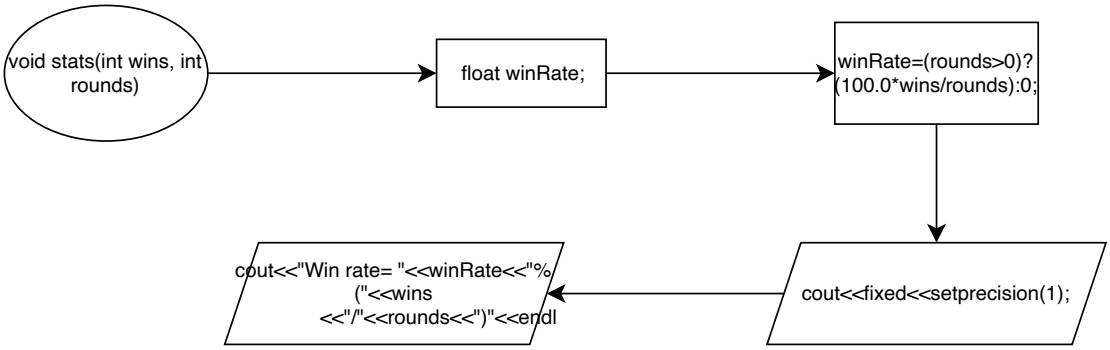


dTurn function

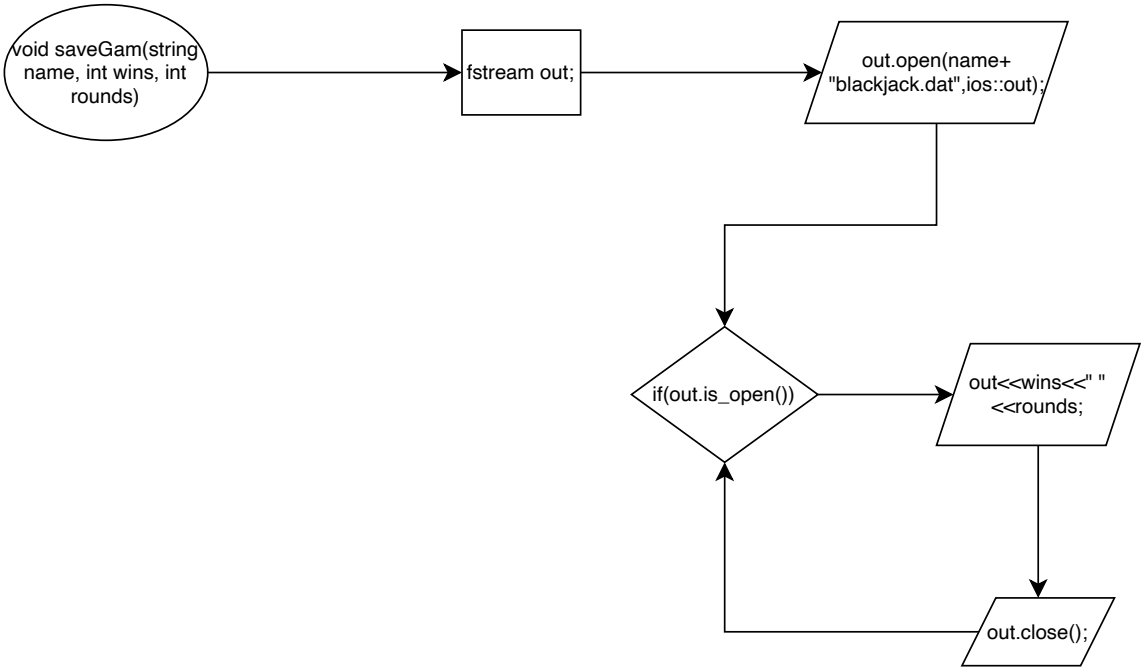



continuation
else if(cTotal>pTotal)
cout << "Dealer wins this round." << endl;
else if(cTotal==pTotal)
cout<<endl<<"DRAW!" <<endl<<endl;
else
cout<<name<<" is a WINNER! :)" <<endl<<endl;
wins++;

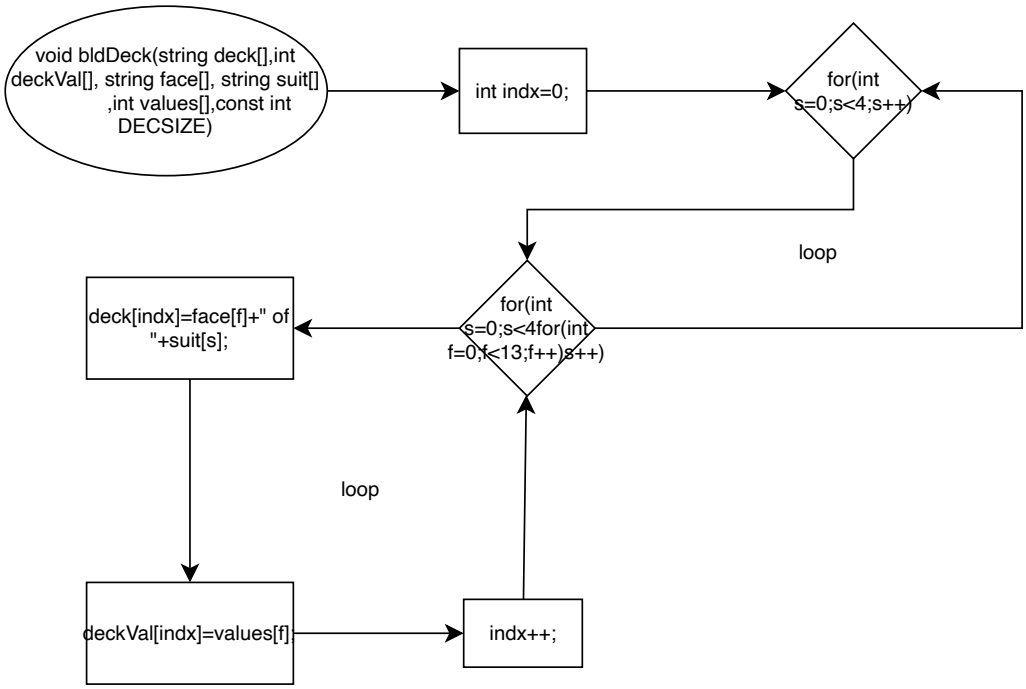
stats function

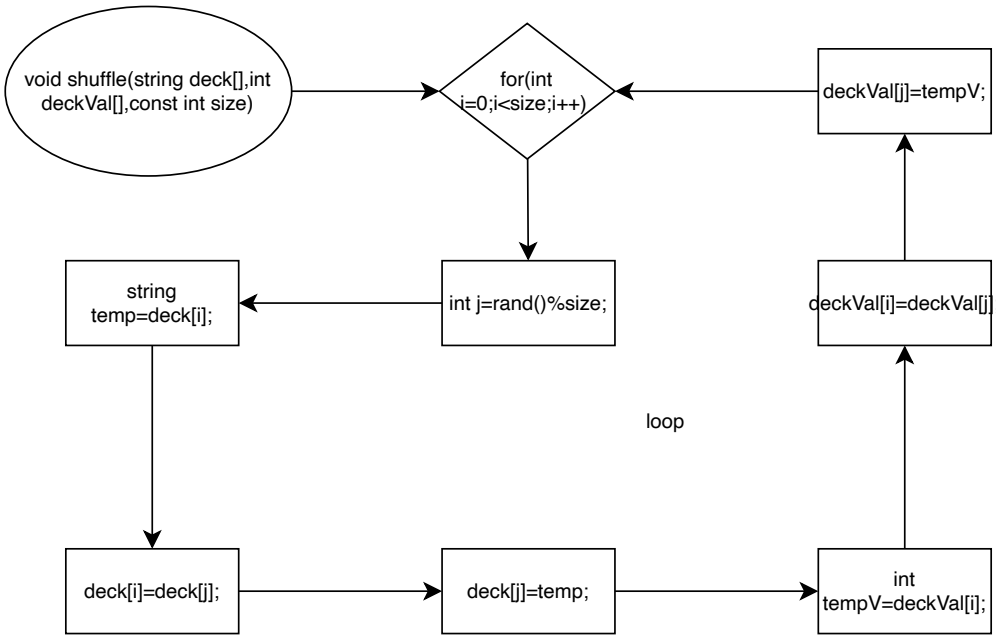


saveGam function

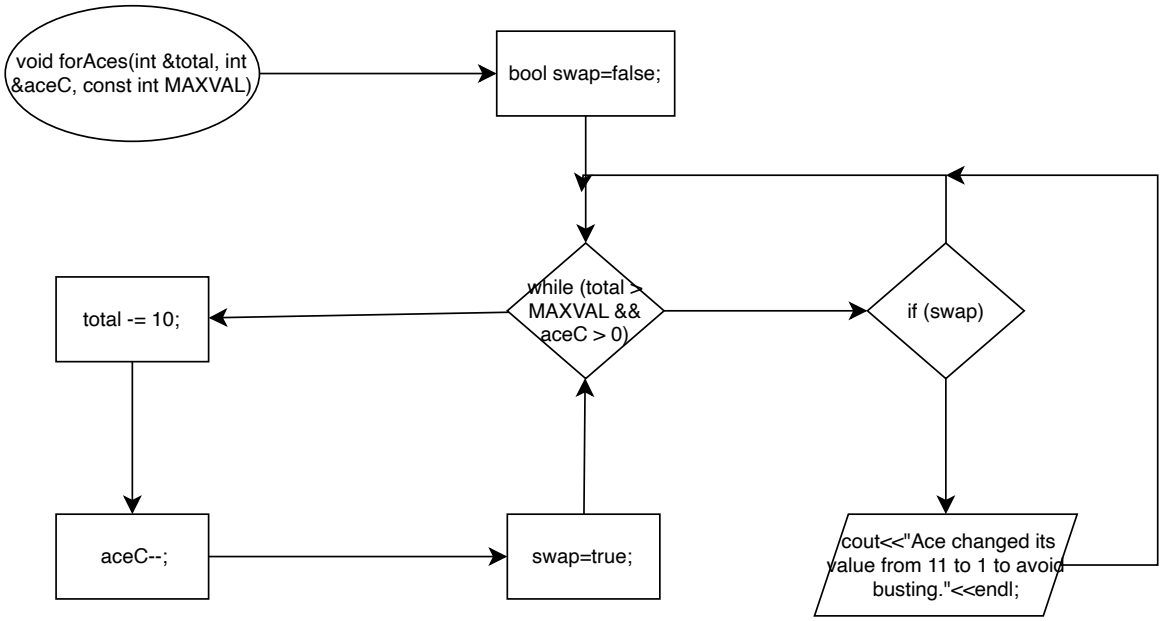


bldDeck function

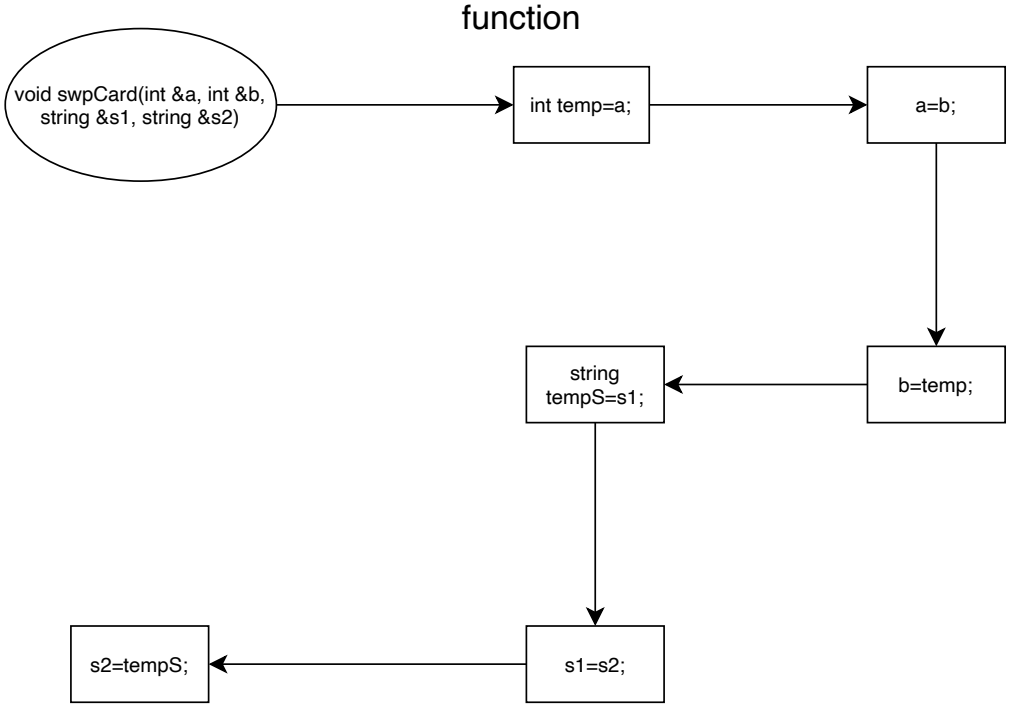


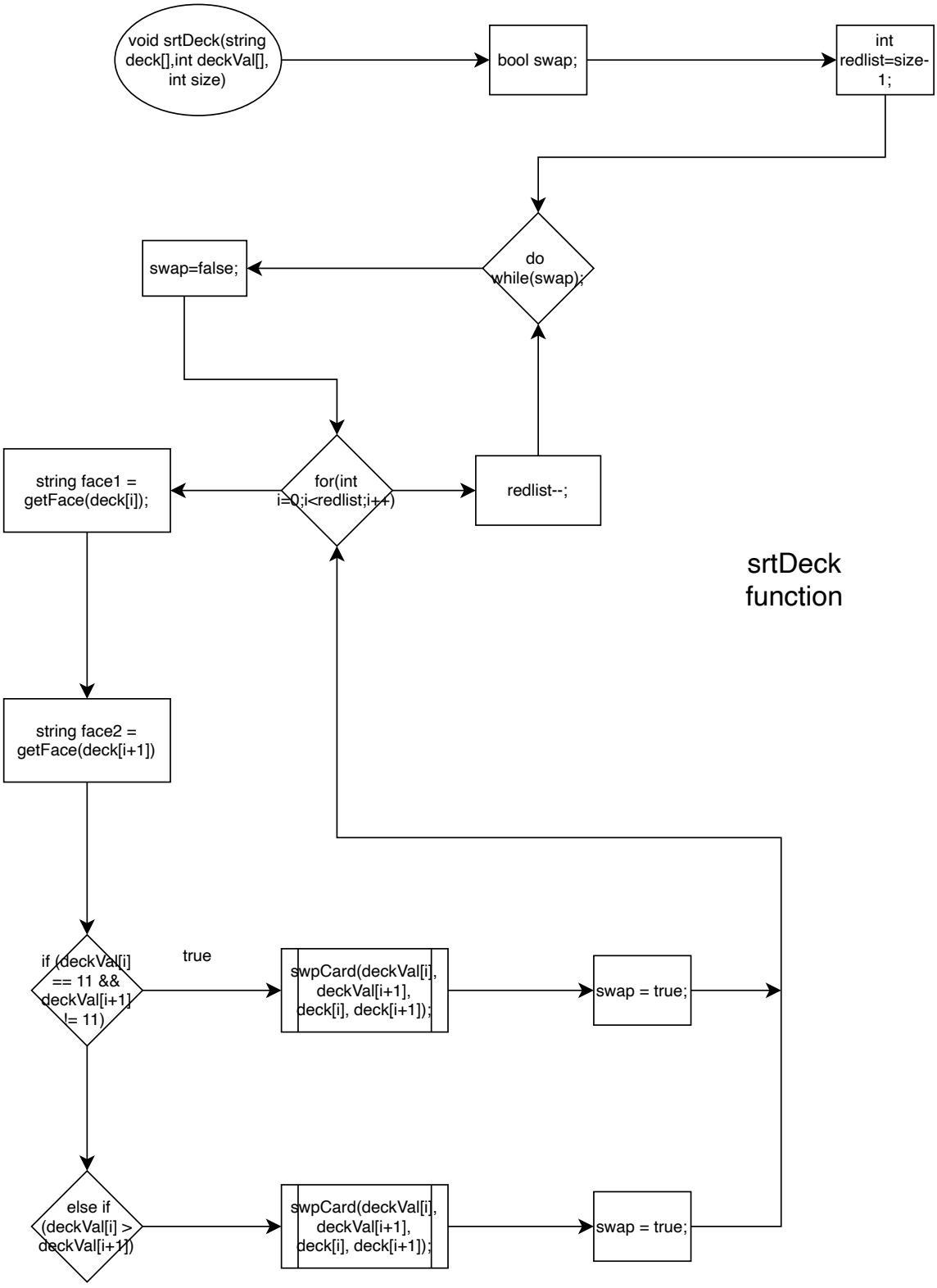


forAces fuction

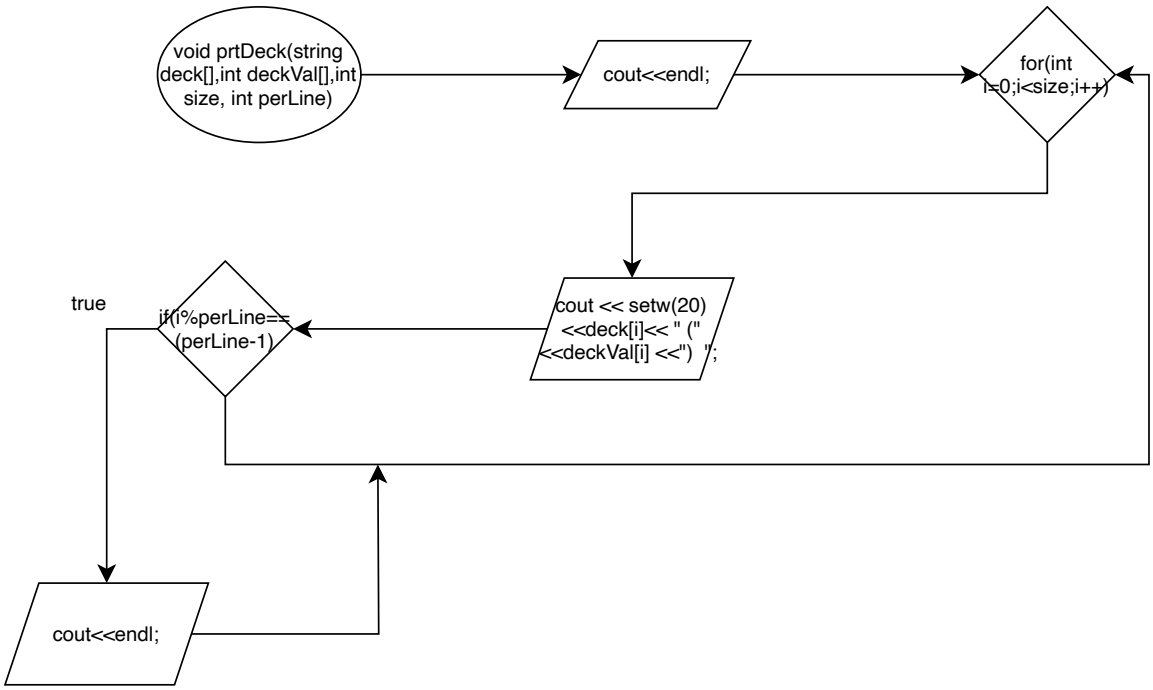
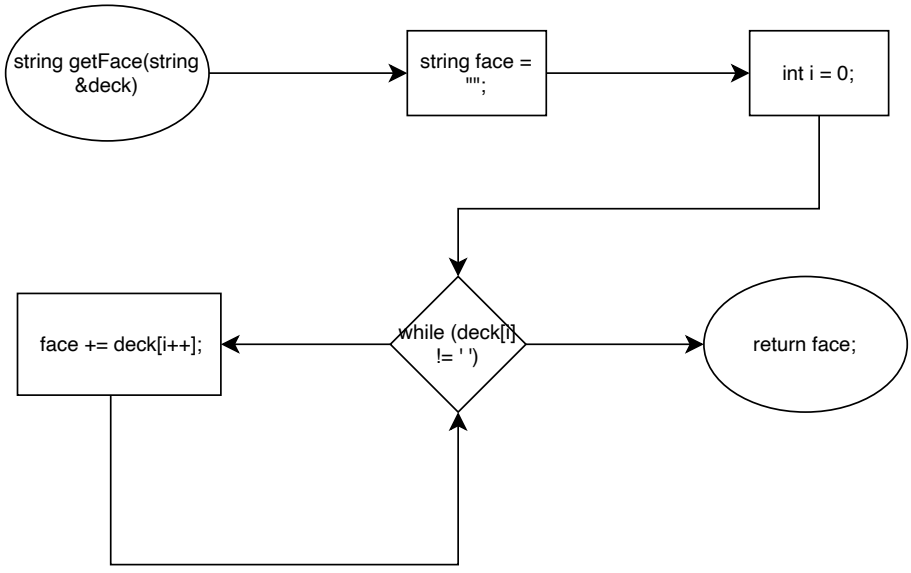


swpCard function




void srtDeck(string deck[],int deckVal[], int size)
bool swap;
int redlist=size-1;
do while(swap);
swap=false;
for(int i=0;i<redlist;i++)
redlist--;
string face1 = getFace(deck[i]);
string face2 = getFace(deck[i+1])
if (deckVal[i] == 11 && deckVal[i+1] != 11)
true
swpCard(deckVal[i], deckVal[i+1], deck[i], deck[i+1]);
swap = true;
else if (deckVal[i] > deckVal[i+1])
swpCard(deckVal[i], deckVal[i+1], deck[i], deck[i+1]);
swap = true;
srtDeck function

prtDeck function



getFace function