Project 1
<Snakes and Ladders>

CIS-5
Name: David Seitz
Date: 5/3/20

Introduction

Title: Snakes and Ladders

Snakes and Ladders is played on a board with a ten by ten grid. On the board, there are Snakes that connect to tiles lower on the board and Ladders that connect to a higher point on the board. It is a player with up to four players who take turns rolling a six-sided dice to move forward that many spaces. If they land on a Snake they will go back to the previous space that the Snake connects to. If they land on a Ladder they will progress to the higher space that the ladder connects to. The goal is to reach the 100 space first. If a player's role would put them on a space greater than 100, they do not move; they must land exactly on 100.

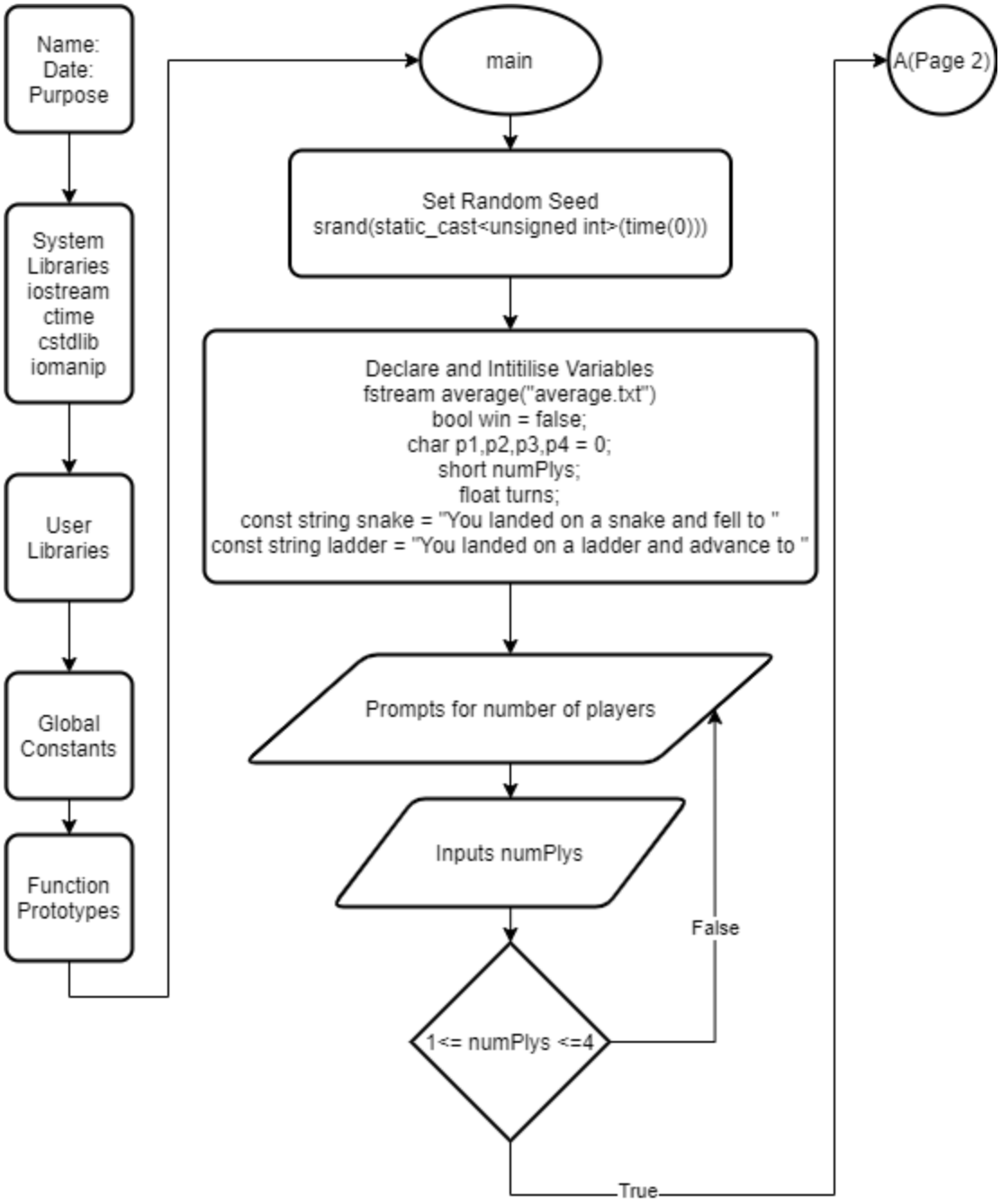
Summary

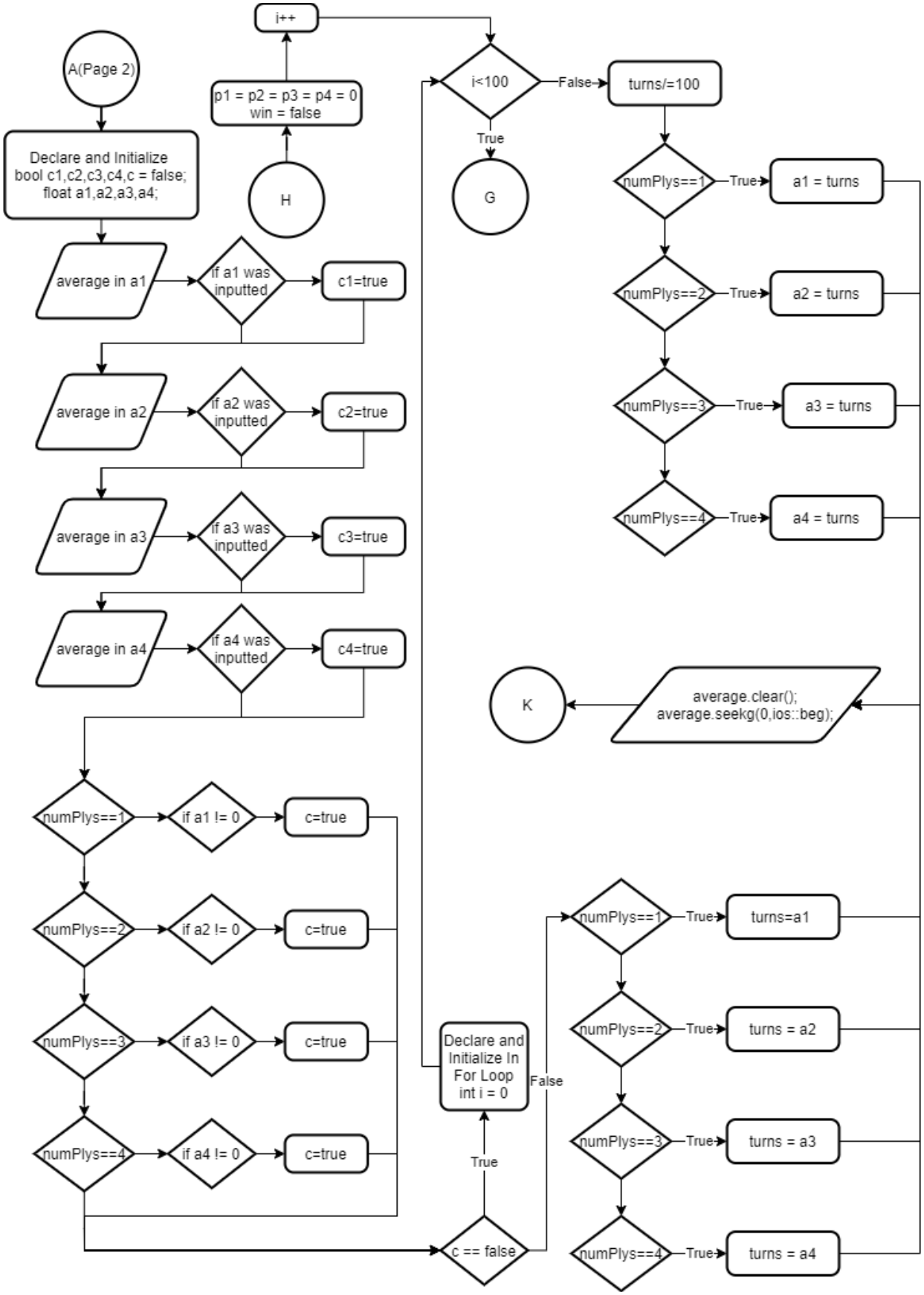
Project size: 386 Lines
The number of variables: about 23
The number of methods: 1?

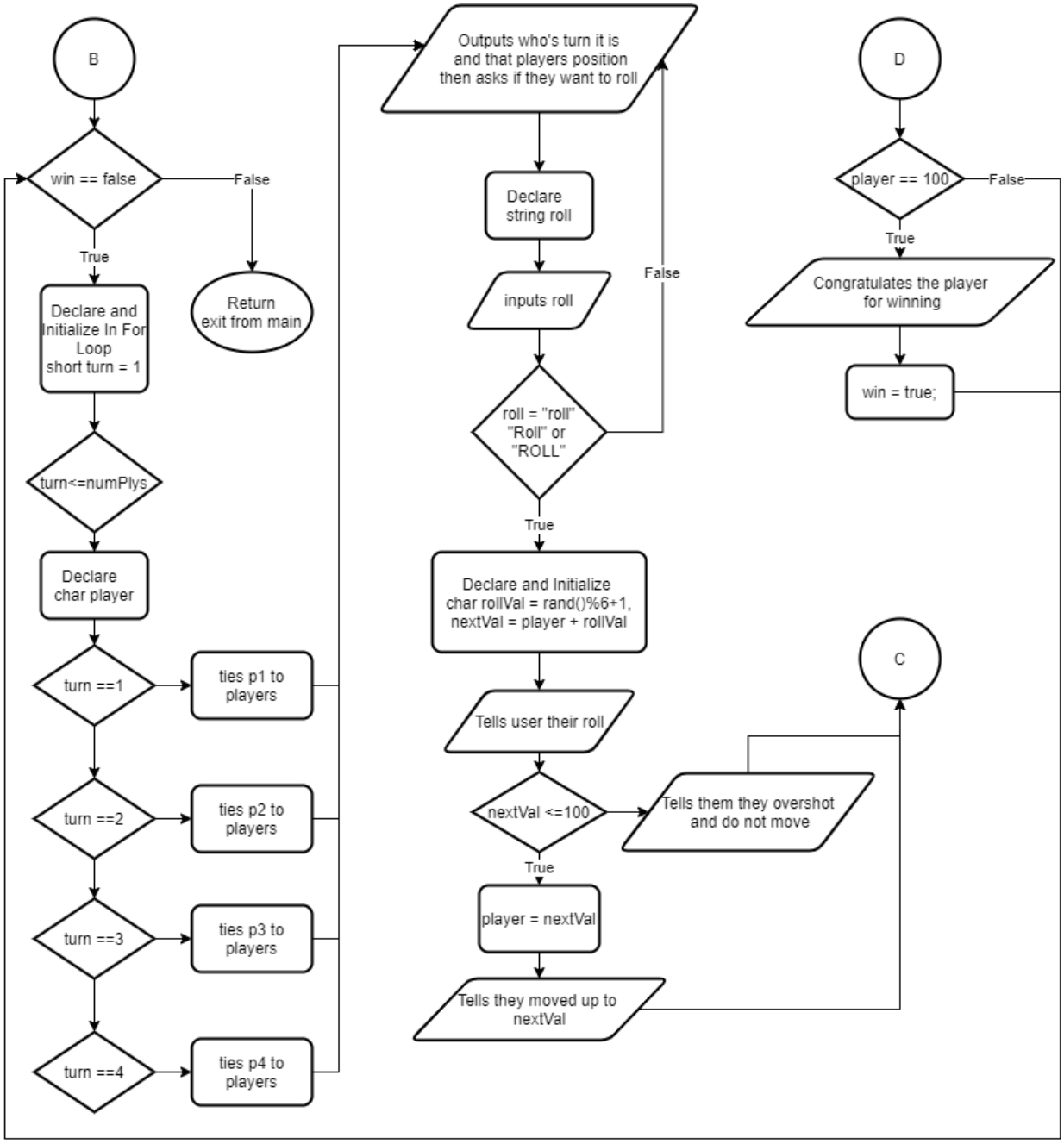
The project includes almost all concepts found in the first 5 chapters of Gaddis with the notable exception of cmath which I did not have a use for. I tried to implement the game in the best manner I could think of without using arrays and functions. I included the entire game as well as a system to estimate how many turns a game will last on average.

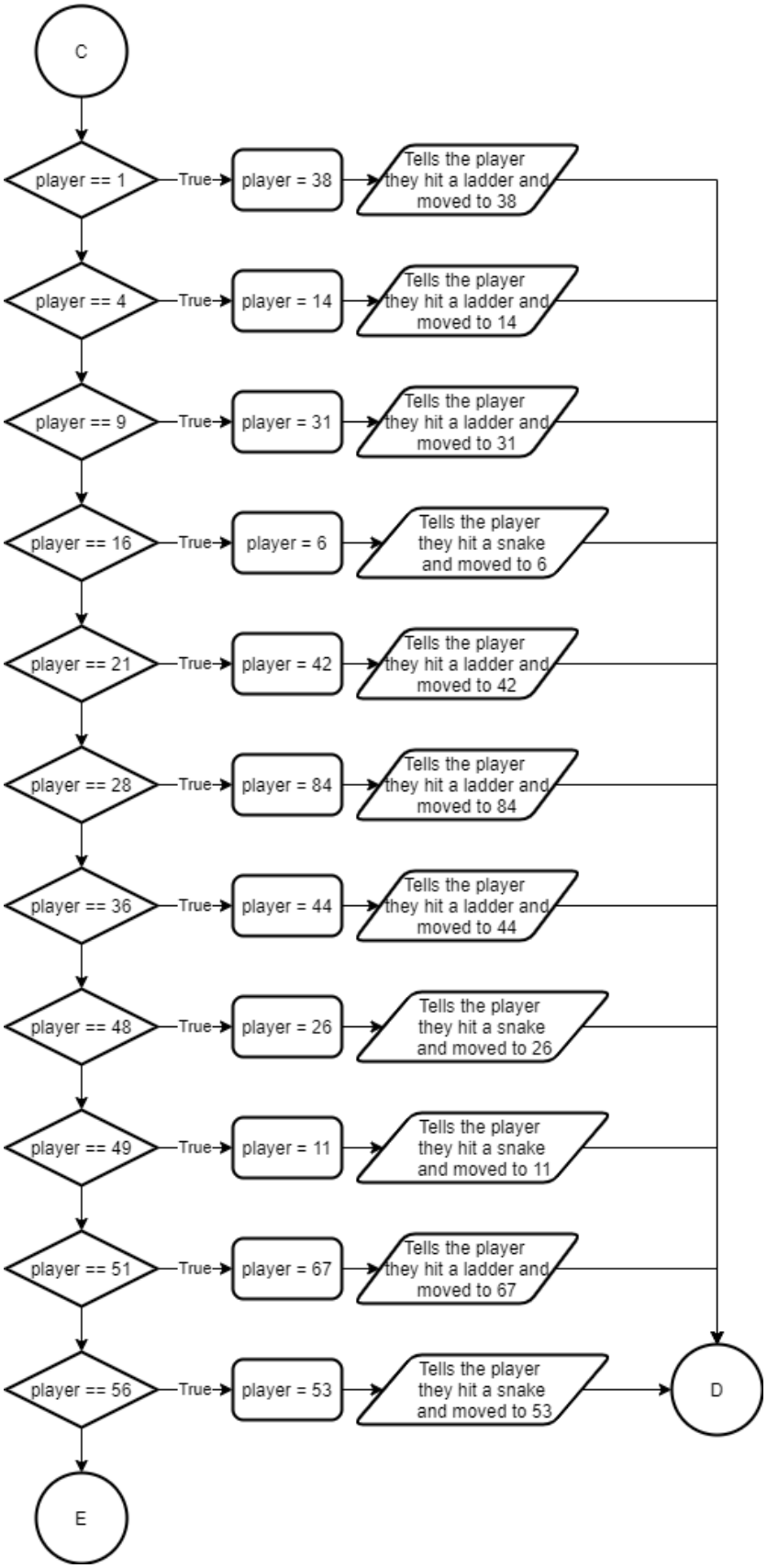
It took me three weeks but two of those were dedicated to creating the card game Coup, which I moved away from because it would have been too unwieldy to flowchart with its over 5k lines.
It was not hard because I have prior experience programing games in other languages.
I am not satisfied with Snakes and Ladders but I am proud of my work.

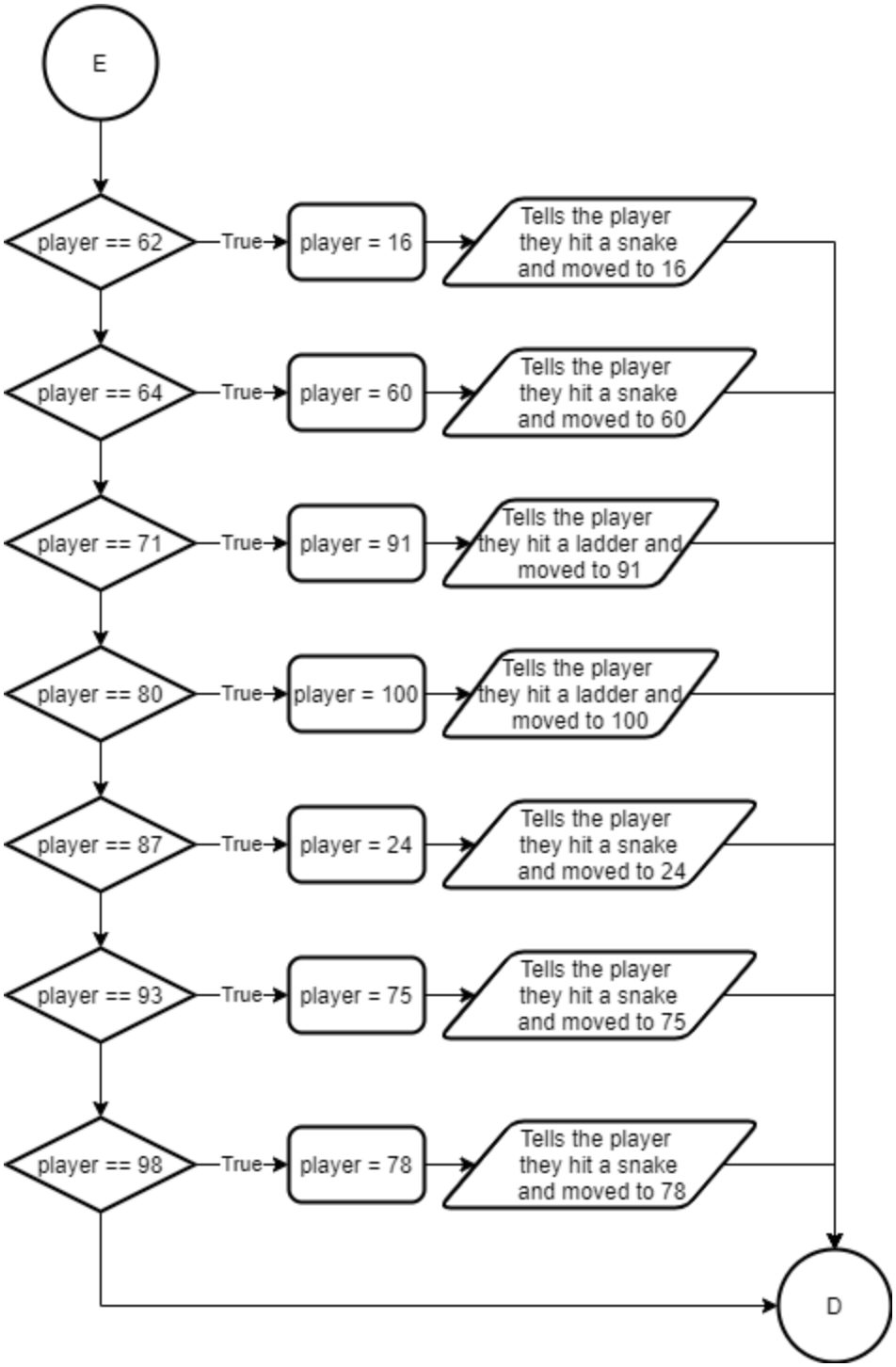
Flowchart

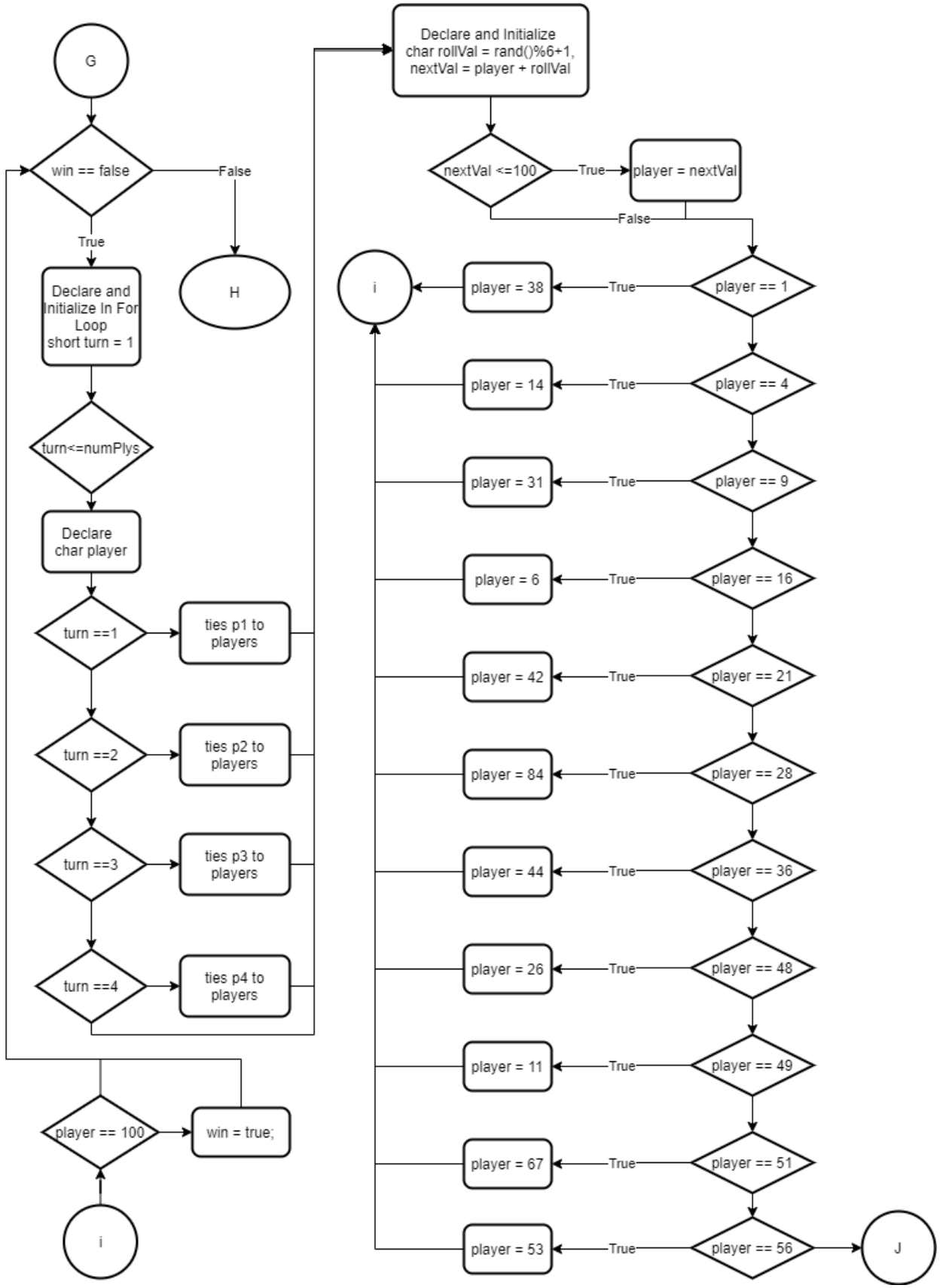

Name:
Date:
Purpose
System
Libraries
iostream
ctime
cstdlib
iomanip
User
Libraries
Global
Constants
Function
Prototypes
main
Set Random Seed
srand(static_cast<unsigned int>(time(0)))
Declare and Intitilise Variables
fstream average("average.txt")
bool win = false;
char p1,p2,p3,p4 = 0;
short numPlys;
float turns;
const string snake = "You landed on a snake and fell to "
const string ladder = "You landed on a ladder and advance to "
Prompts for number of players
Inputs numPlys
1<= numPlys <=4
False
True
A(Page 2)

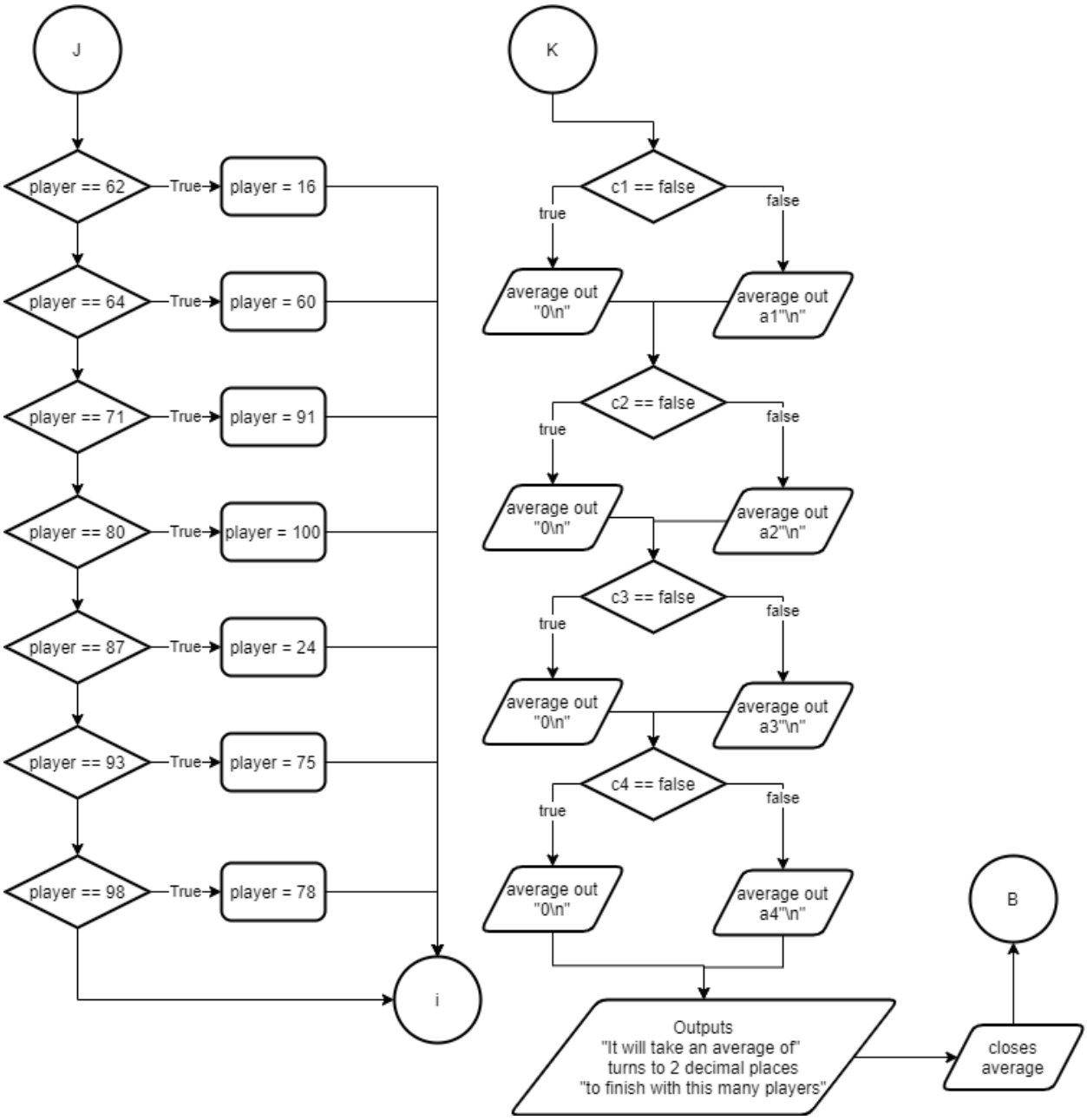

A(Page 2)
Declare and Initialize
bool c1,c2,c3,c4,c = false;
float a1,a2,a3,a4;
average in a1
if a1 was inputted
c1=true
average in a2
if a2 was inputted
c2=true
average in a3
if a3 was inputted
c3=true
average in a4
if a4 was inputted
c4=true
numPlys==1
if a1 != 0
c=true
numPlys==2
if a2 != 0
c=true
numPlys==3
if a3 != 0
c=true
numPlys==4
if a4 != 0
c=true
c == false
True
Declare and Initialize In For Loop
int i = 0
False
i<100
True
G
False
turns/=100
numPlys==1
True
a1 = turns
numPlys==2
True
a2 = turns
numPlys==3
True
a3 = turns
numPlys==4
True
a4 = turns
average.clear();
average.seekg(0,ios::beg);
K
numPlys==1
True
turns=a1
numPlys==2
True
turns = a2
numPlys==3
True
turns = a3
numPlys==4
True
turns = a4
i++
p1 = p2 = p3 = p4 = 0
win = false
H


B
win == false
False
True
Return
exit from main
Declare and
Initialize In For
Loop
short turn = 1
turn<=numPlys
Declare
char player
turn ==1
ties p1 to
players
turn ==2
ties p2 to
players
turn ==3
ties p3 to
players
turn ==4
ties p4 to
players
Outputs who's turn it is
and that players position
then asks if they want to roll
Declare
string roll
inputs roll
roll = "roll"
"Roll" or
"ROLL"
False
True
Declare and Initialize
char rollVal = rand()%6+1,
nextVal = player + rollVal
Tells user their roll
nextVal <=100
Tells them they overshot
and do not move
True
player = nextVal
Tells they moved up to
nextVal
C
D
player == 100
False
True
Congratulates the player
for winning
win = true;


C
player == 1
True
player = 38
Tells the player they hit a ladder and moved to 38
player == 4
True
player = 14
Tells the player they hit a ladder and moved to 14
player == 9
True
player = 31
Tells the player they hit a ladder and moved to 31
player == 16
True
player = 6
Tells the player they hit a snake and moved to 6
player == 21
True
player = 42
Tells the player they hit a ladder and moved to 42
player == 28
True
player = 84
Tells the player they hit a ladder and moved to 84
player == 36
True
player = 44
Tells the player they hit a ladder and moved to 44
player == 48
True
player = 26
Tells the player they hit a snake and moved to 26
player == 49
True
player = 11
Tells the player they hit a snake and moved to 11
player == 51
True
player = 67
Tells the player they hit a ladder and moved to 67
player == 56
True
player = 53
Tells the player they hit a snake and moved to 53
D
E


E
player == 62
True
player = 16
Tells the player they hit a snake and moved to 16
player == 64
True
player = 60
Tells the player they hit a snake and moved to 60
player == 71
True
player = 91
Tells the player they hit a ladder and moved to 91
player == 80
True
player = 100
Tells the player they hit a ladder and moved to 100
player == 87
True
player = 24
Tells the player they hit a snake and moved to 24
player == 93
True
player = 75
Tells the player they hit a snake and moved to 75
player == 98
True
player = 78
Tells the player they hit a snake and moved to 78
D


G
win == false
False
True
H
Declare and Initialize In For Loop short turn = 1
turn<=numPlys
Declare char player
turn ==1
ties p1 to players
turn ==2
ties p2 to players
turn ==3
ties p3 to players
turn ==4
ties p4 to players
player == 100
win = true;
i
Declare and Initialize char rollVal = rand()%6+1, nextVal = player + rollVal
nextVal <=100
True
player = nextVal
False
player == 1
True
player = 38
i
player == 4
True
player = 14
player == 9
True
player = 31
player == 16
True
player = 6
player == 21
True
player = 42
player == 28
True
player = 84
player == 36
True
player = 44
player == 48
True
player = 26
player == 49
True
player = 11
player == 51
True
player = 67
player == 56
True
player = 53
J


J
player == 62
True
player = 16
player == 64
True
player = 60
player == 71
True
player = 91
player == 80
True
player = 100
player == 87
True
player = 24
player == 93
True
player = 75
player == 98
True
player = 78
i
K
c1 == false
true
false
average out "0\n"
average out a1"\n"
c2 == false
true
false
average out "0\n"
average out a2"\n"
c3 == false
true
false
average out "0\n"
average out a3"\n"
c4 == false
true
false
average out "0\n"
average out a4"\n"
Outputs "It will take an average of" turns to 2 decimal places "to finish with this many players"
closes average
B

Sample console:

```
How many players are there? Enter a number between 1 and 4: 3
It will take an average of 23.53 turns to finish with this many players.



Its player 1's turn.
You are on space 0, type roll when you are ready to roll: roll
You rolled a 2
You move up too 2


Its player 2's turn.
You are on space 0, type roll when you are ready to roll:
```

This is the first turn. The user inputed that there are 3 players. The first player took their turn by typing roll. The second player's turn started but they did not roll yet.