Project 1

<Mexico Game>

Course: CIS-5 47948

Author: Amos Lopez

Date: 11-1-18

1 Introduction

Title: Mexico

This game is a dice game that is usually played with a minimum of 2 people. Players each take a turn to roll the dice. The player with the higher ranked roll wins. The highest ranked roll is 21, also known as “coming up Mexico” which is where the game get its name. 2nd best roll is rolling doubles with 66 being second best roll after 21, and 11 being the lowest double. However, rolling 11 is still better than 65 or any other non-double roll. After doubles, the best roll is whichever roll is the greater number like 45 beats 23.

Below is the table that shows how each roll is ranked from best to worst.

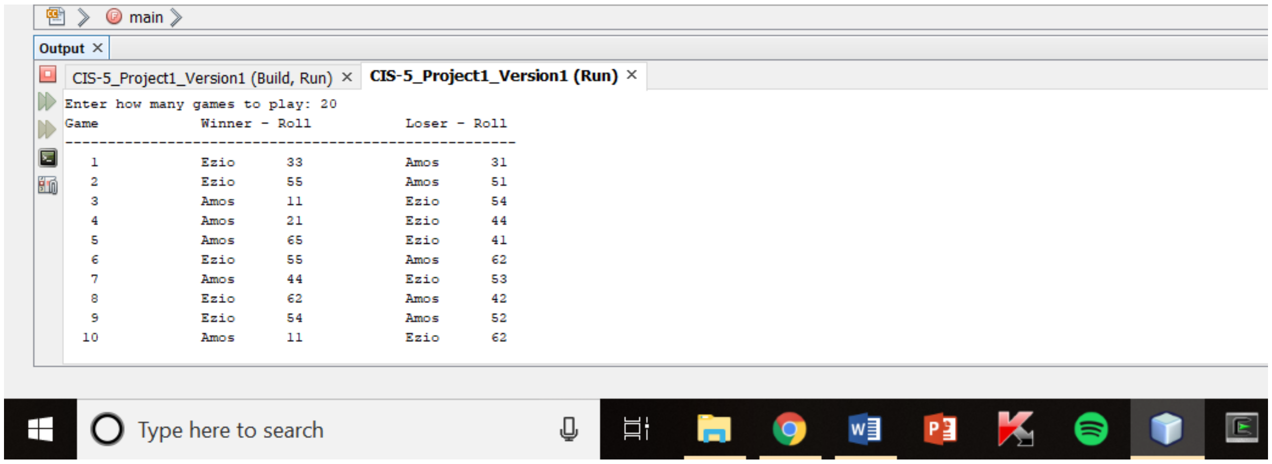
Best	1	21	Automatically win
	2	66	
	3	55	
	4	44	
	5	33	
	6	22	
	7	11	Still beats non doubles
Worst	8	Any non-double, non-21 roll	(65, 23, 45, etc…)

The game begins with the program asking for the name of each player then how many games or rounds to play. Once the user inputs the data, the game will display the results of each round showing the winner, loser, and what each player rolled for that game. If players get the same roll the program will automatically re-roll the dice until they have different outcomes in order to avoid a tie.

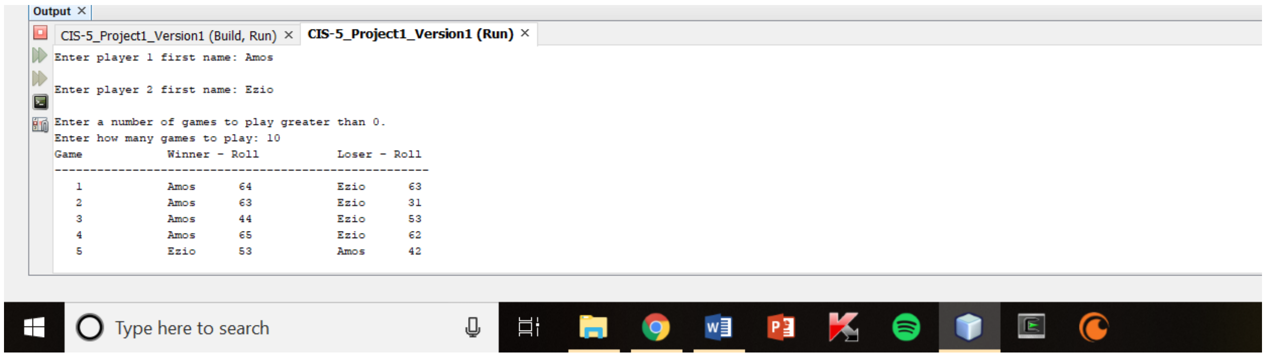
Once the amount of games entered at the start have been played out, the game will ask if you want to play again. If yes, then the game will ask you to enter ‘Y’ for yes or enter ‘N’ for no and the game will end.

Pseudo Code

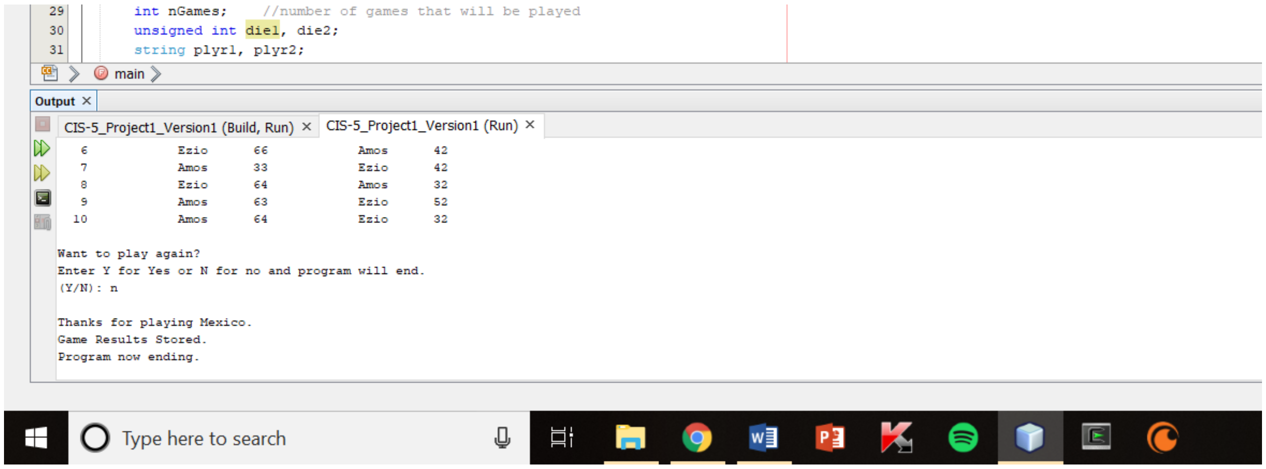
```
1. Get system time
2. Seed random number generator
3. Open output File
4. Prompt user for input
5.      Get names of players
6.      Get number of games to play
7. do this for each game:
8.      Roll dice for player 1
9.      Get sum of dice for player 1
10.     Roll dice for player 2
11.     Get sum of dice for player 2
12.     If dice both dice rolls are the same re-roll again
13.     if dice rolls are different determine winner
14.     output game number, winner, loser, both rolls to user
15.     store game results in output file then repeat step 8-15 until all games are done
16. when all games are done, ask user if they want to play again
17. repeat step 4-15 until user does not want to play anymore
18. display good-bye message to user
19. End Program.
```


main
Output
CIS-5_Project1_Version1 (Build, Run)
CIS-5_Project1_Version1 (Run)
Type here to search

```
Enter how many games to play: 20
Game              Winner - Roll              Loser - Roll
-----------------------------------------------------------
   1              Ezio      33               Amos      31
   2              Ezio      55               Amos      51
   3              Amos      11               Ezio      54
   4              Amos      21               Ezio      44
   5              Amos      65               Ezio      41
   6              Ezio      55               Amos      62
   7              Amos      44               Ezio      53
   8              Ezio      62               Amos      42
   9              Ezio      54               Amos      52
  10              Amos      11               Ezio      62
```

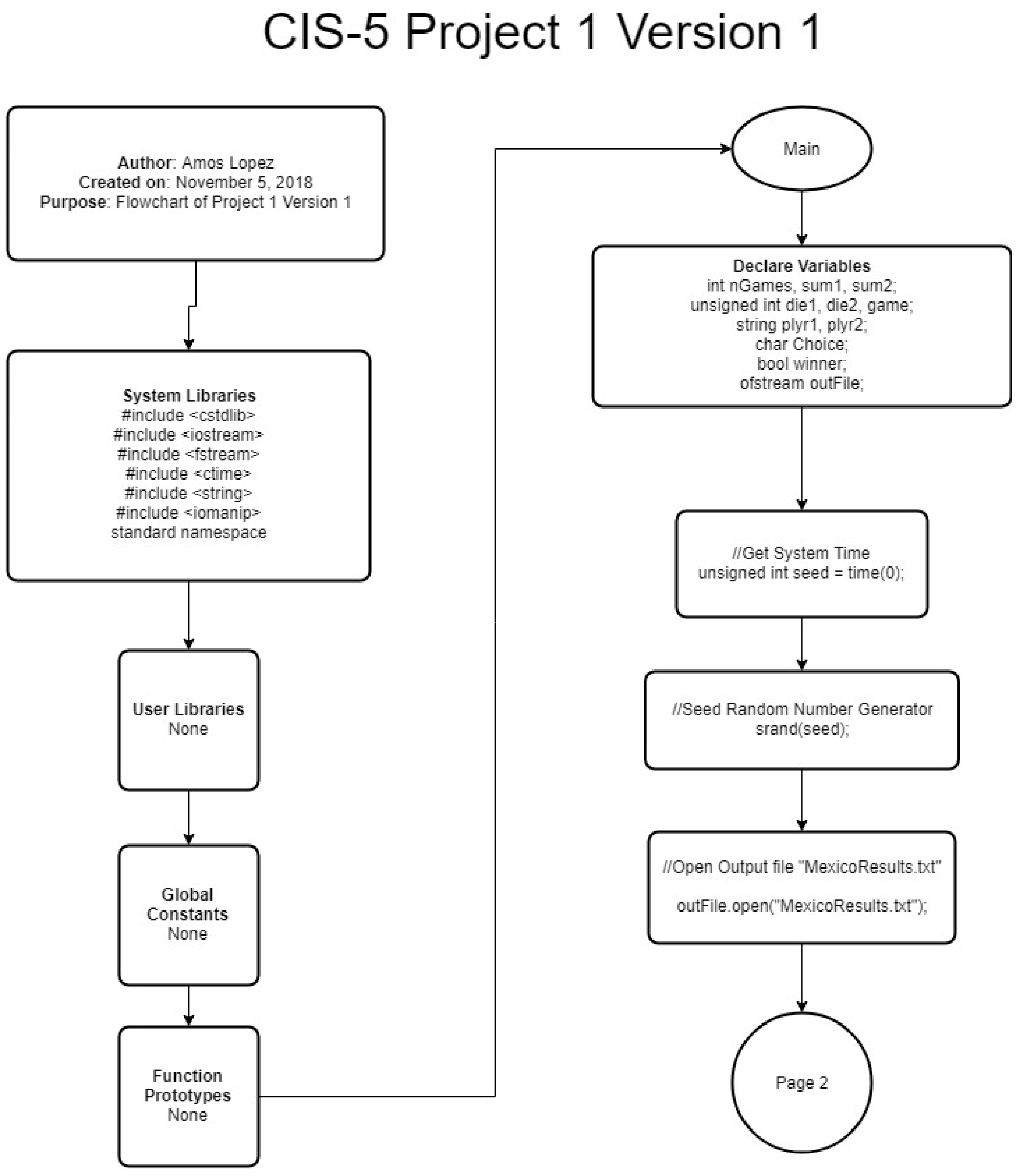

Output
CIS-5_Project1_Version1 (Build, Run)
CIS-5_Project1_Version1 (Run)
Type here to search

```
Enter player 1 first name: Amos

Enter player 2 first name: Ezio

Enter a number of games to play greater than 0.
Enter how many games to play: 10
Game              Winner - Roll              Loser - Roll
-----------------------------------------------------------
   1              Amos      64               Ezio      63
   2              Amos      63               Ezio      31
   3              Amos      44               Ezio      53
   4              Amos      65               Ezio      62
   5              Ezio      53               Amos      42
```

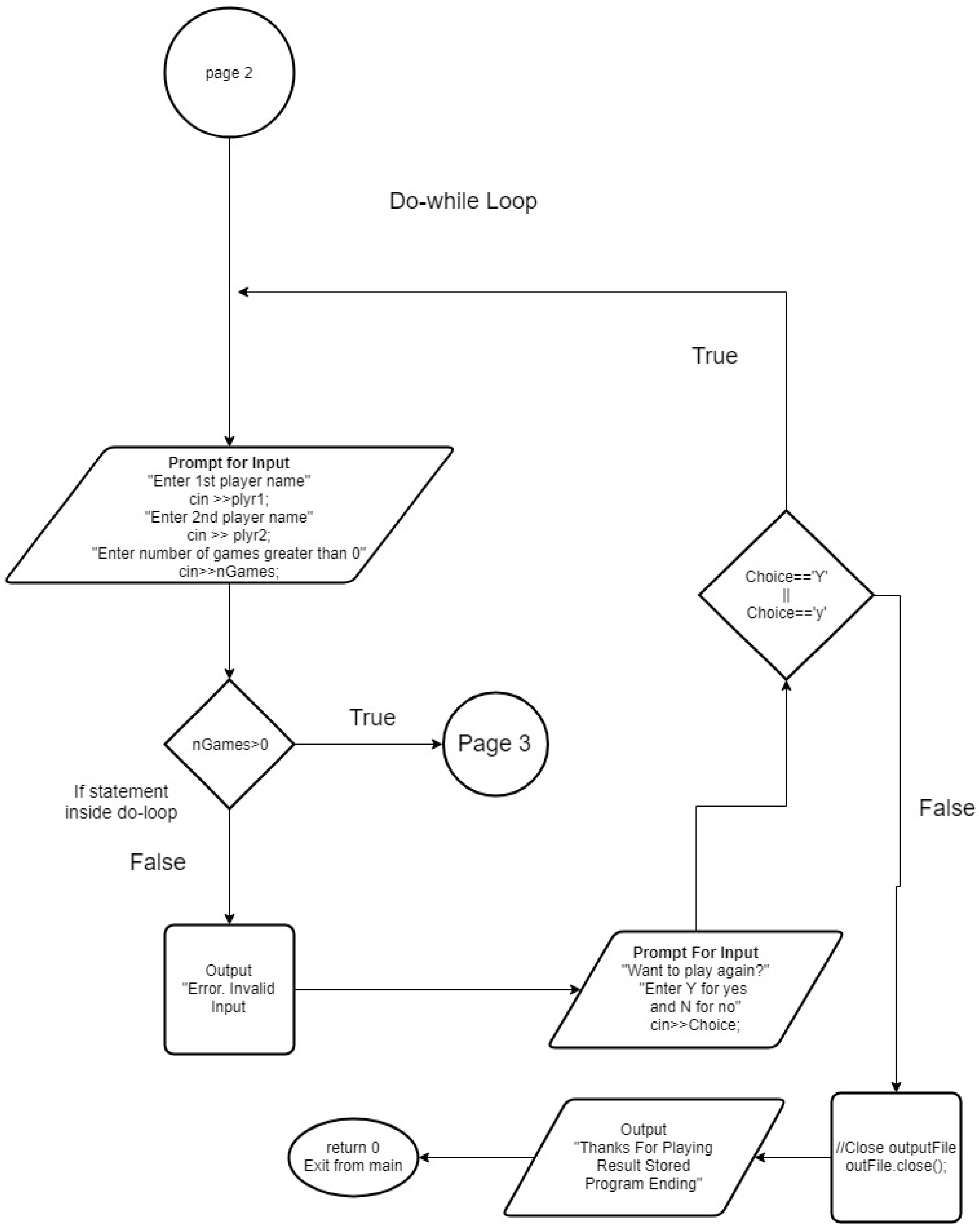

```
        int nGames;     //number of games that will be played
        unsigned int die1, die2;
        string plyr1, plyr2;
```
main
Output
CIS-5_Project1_Version1 (Build, Run)
CIS-5_Project1_Version1 (Run)
Type here to search

```
   6              Ezio      66               Amos      42
   7              Amos      33               Ezio      42
   8              Ezio      64               Amos      32
   9              Amos      63               Ezio      52
  10              Amos      64               Ezio      32

Want to play again?
Enter Y for Yes or N for no and program will end.
(Y/N): n

Thanks for playing Mexico.
Game Results Stored.
Program now ending.
```

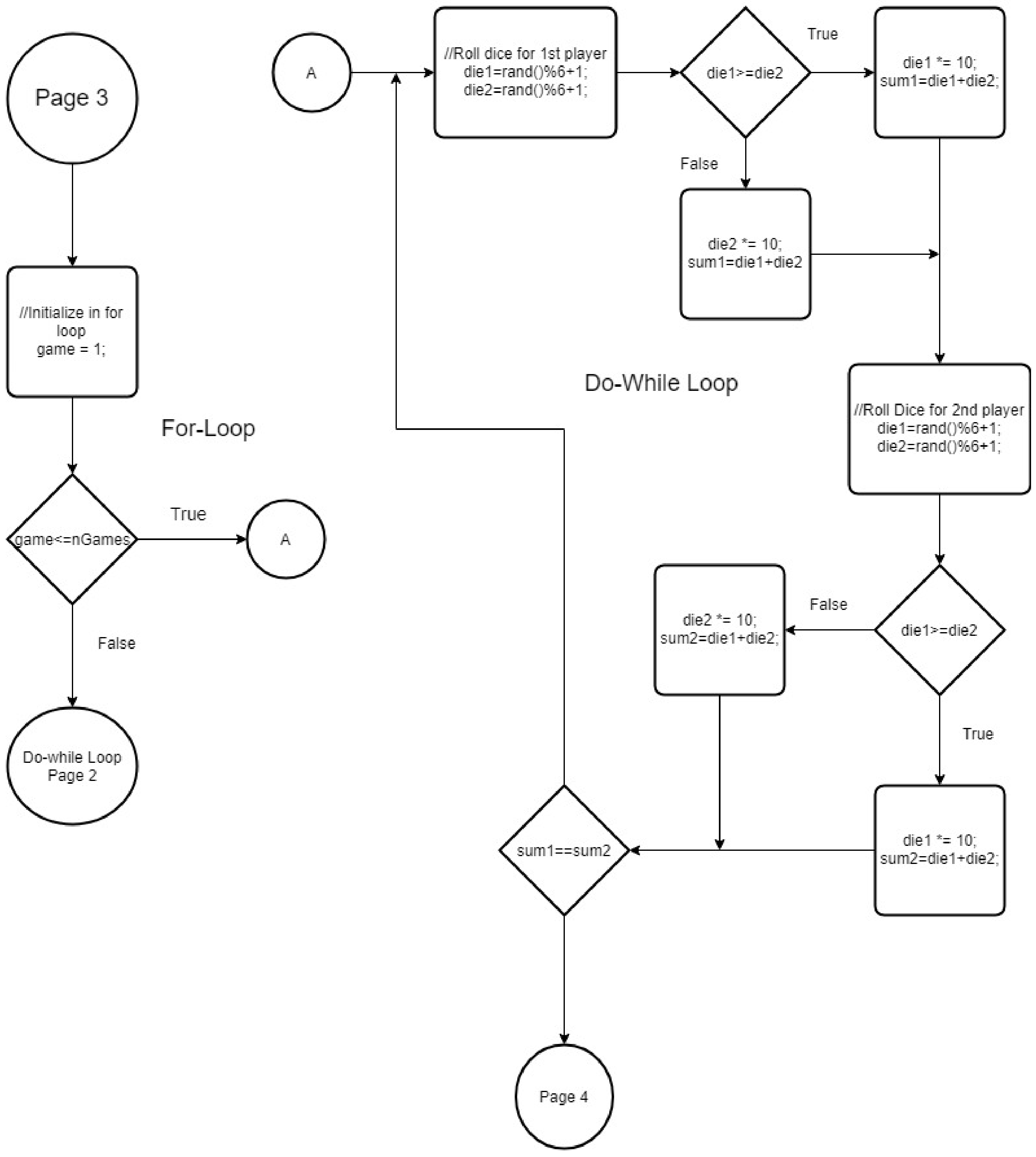
3 Flow Chart


CIS-5 Project 1 Version 1
Author: Amos Lopez
Created on: November 5, 2018
Purpose: Flowchart of Project 1 Version 1
System Libraries
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <ctime>
#include <string>
#include <iomanip>
standard namespace
User Libraries
None
Global Constants
None
Function Prototypes
None
Main
Declare Variables
int nGames, sum1, sum2;
unsigned int die1, die2, game;
string plyr1, plyr2;
char Choice;
bool winner;
ofstream outFile;
//Get System Time
unsigned int seed = time(0);
//Seed Random Number Generator
srand(seed);
//Open Output file "MexicoResults.txt"
outFile.open("MexicoResults.txt");
Page 2

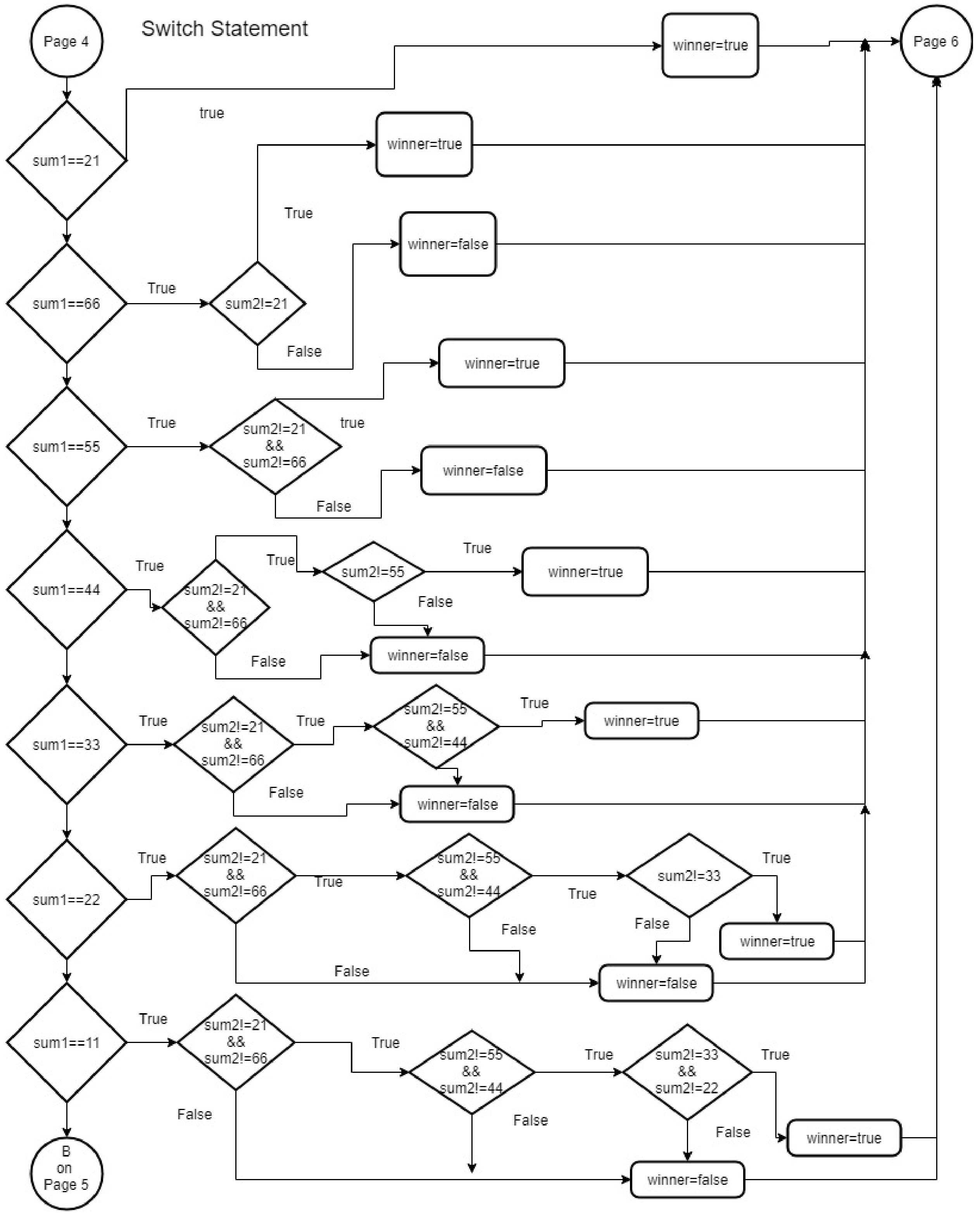
CIS-5 Project 1 Version 1



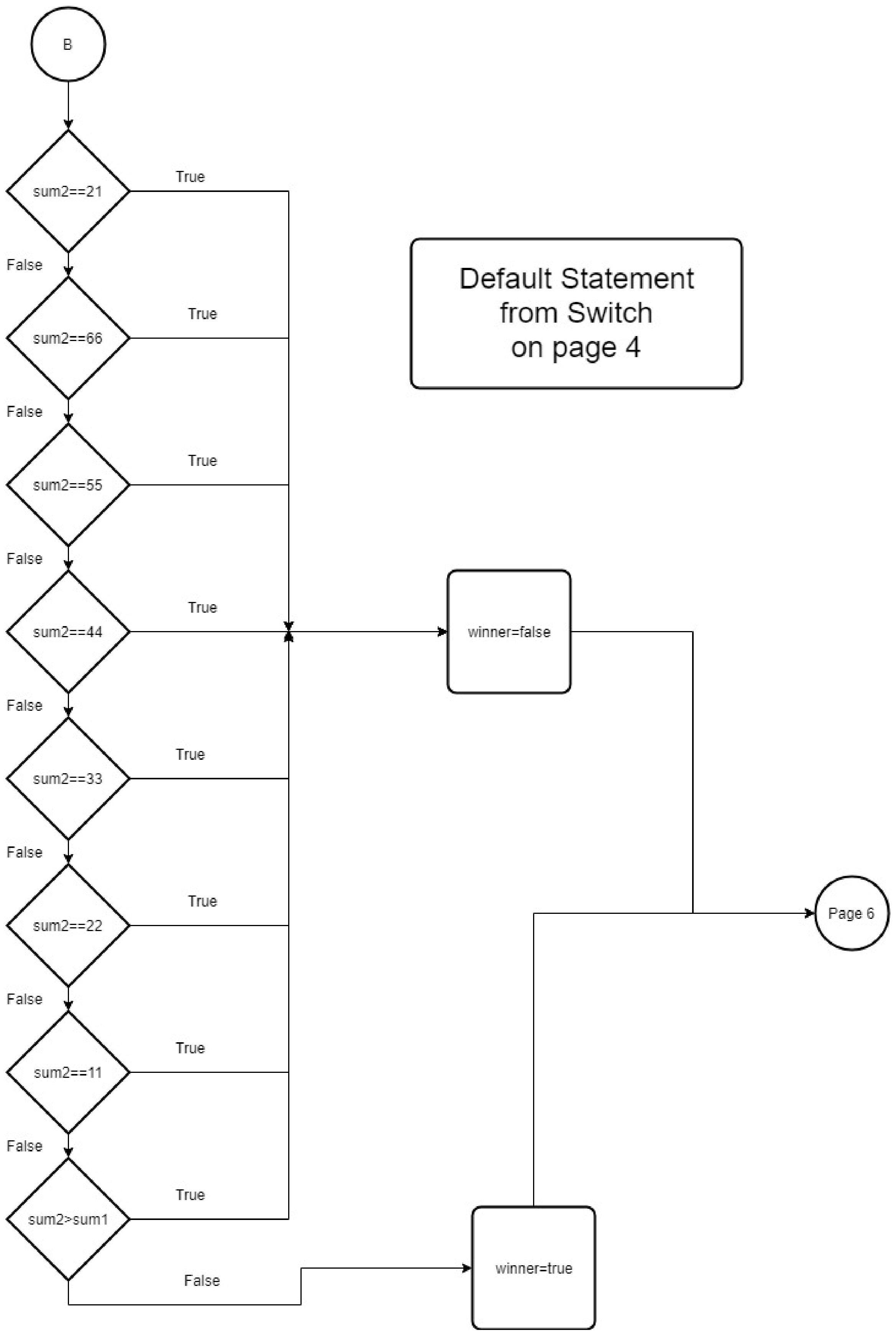
CIS-5 Project 1 Version 1

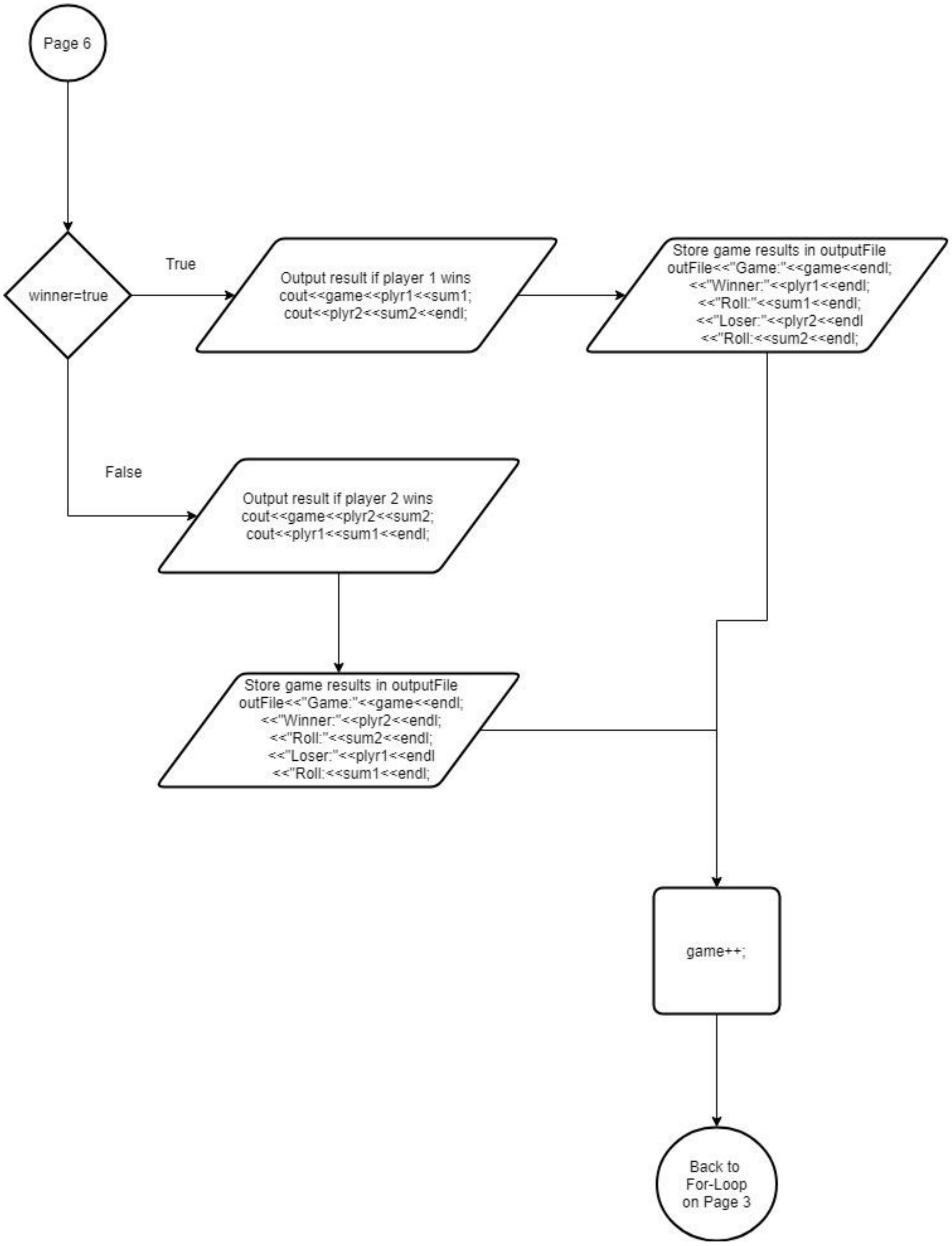


CIS-5 Project 1 Version 1



]


B
sum2==21
True
False
sum2==66
True
False
sum2==55
True
False
sum2==44
True
False
sum2==33
True
False
sum2==22
True
False
sum2==11
True
False
sum2>sum1
True
False
Default Statement
from Switch
on page 4
winner=false
winner=true
Page 6


Page 6
winner=true
True
False
Output result if player 1 wins
cout<<game<<plyr1<<sum1;
cout<<plyr2<<sum2<<endl;
Store game results in outputFile
outFile<<"Game:"<<game<<endl;
<<"Winner:"<<plyr1<<endl;
<<"Roll:"<<sum1<<endl;
<<"Loser:"<<plyr2<<endl
<<"Roll:<<sum2<<endl;
Output result if player 2 wins
cout<<game<<plyr2<<sum2;
cout<<plyr1<<sum1<<endl;
Store game results in outputFile
outFile<<"Game:"<<game<<endl;
<<"Winner:"<<plyr2<<endl;
<<"Roll:"<<sum2<<endl;
<<"Loser:"<<plyr1<<endl
<<"Roll:<<sum1<<endl;
game++;
Back to
For-Loop
on Page 3

4 Summary

Blank Lines (white spaces)	14	
Comment Lines	38	
Lines of Actual Code	170	
Total Lines in Source File	222	
Number of Variables	12	

This project includes many concepts that we have learned in the book. Although the project runs well enough, I am not nearly satisfied with it at all. There many improvements that can be made to the existing source file, and can hopefully be implemented in future versions of the project.

The main thing I spent the most time on was trying to implement the ranking system of the dice rolls into the program. At first I hit a snag when I would test out some input and would end up with the 1st player winning even if the 2nd player rolled 21 and the 1st player didn't.

Looking at the project as it stands now, I can definitely see where I can make improvements in order to make the program more efficient or compile faster. I know that I will be able to use concepts from future chapters in this project.

20181107_00001.pdf (Command Line)

Cross Reference for Project 1

You are to fill-in with where located in code

Chapter	Section	Topic	Where Line #'s	Pts	Notes
2	2	cout	50-57, 210-220		
	3	libraries	9, 10, 11, 12, 13	8	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals	28-35		No variables in global area, failed project!
	5	Identifiers	28-35		
	6	Integers	51, 64-87	3	
	7	Characters	204	3	
	8	Strings	53; 55	3	
	9	Floats No Doubles		3	Using doubles will fail the project, floats OK!
	10	Bools	96	4	
	11	Sizeof *****			
	12	Variables 7 characters or less	28-35		All variables <= 7 characters
	13	Scope ***** No Global Variables			
	14	Arithmetic operators	69-87		
	15	Comments 20%+	45-206	5	Model as pseudo code
	16	Named Constants			All Local, only Conversions/Physics/Math in Global area
	17	Programming Style ***** Emulate			Emulate style in book/in class repositiory
3	1	cin	51, 53, 55, 204		
	2	Math Expression	69-87		
	3	Mixing data types ****			
	4	Overflow/Underflow ****			
	5	Type Casting		4	
	6	Multiple assignment *****			
	7	Formatting output	176, 177, 189	4	
	8	Strings	53, 55	3	
	9	Math Library		4	All libraries included have to be used
	10	Hand tracing ******			
4	1	Relational Operators	67-205		
	2	if		4	Independent if
	4	If-else	57-150	4	
	5	Nesting	114-148	4	
	6	If-else-if	159-169	4	
	7	Flags *****			
	8	Logical operators	100-168	4	
	11	Validating user input	55	4	
	13	Conditional Operator	101	4	
	14	Switch	94	4	
5	1	Increment/Decrement	59	4	
	2	While		4	
	5	Do-while	47	4	
	6	For loop	59	4	
	11	Files input/output both	45	8	
	12	No breaks in loops ******			Failed Project if included
****** Not required to show			Total	100	