Project 1

Title: Yahtzee

Course: CIS 5

Section: 40652

Name: Allison Ohara

Date: 01/30/18

Introduction:

Title: Yahtzee

My project is the game of Yahtzee. This program allows one player to play Yahtzee in more or less the typical way. However, the player does not play against the computer, but instead tries to get a score of at least 200.

Rules of the Game:

The goal of the game is to get a score of at least 200.

After displaying the introduction message and the last score, the user is prompted to enter 'r' to roll the dice. Once the user does so, the five dice are displayed. The user is asked if he would like to reroll. If he enters 1, he may choose which dice to reroll. If he enters zero, then he is prompted to enter which category he would like to score in.

He chooses which dice to reroll by entering five 1's or 0's, with 1 meaning reroll the die and 0 meaning keep the die the same. The dice are displayed again, with some of them rerolled, and the user is asked if he would like to reroll. The process repeats itself once more.

The final numbers are displayed, along with the available category choices. There are 13 different categories: ones, twos, threes, fours, fives, sixes, three of a kind, four of a kind, small straight, large straight, full house, chance, and Yahtzee. The user chooses a category by typing and entering it, and his dice are scored for that category. Then, the next round starts. Each category is used once, and once all have been filled the game ends. If the user gets 63 or more points in the "upper" categories (ones, twos, threes, fours, fives, and sixes), then he gets a 35 point bonus. The sum of the scores in each of the categories, plus the bonus, is calculated. If the user gets a score of 200 or better, they win!

Here is how the scoring for each category works:

Ones: score is the sum of all the ones rolled

Twos: score is the sum of all the twos rolled

Threes: score is the sum of all the threes rolled

Fours: score is the sum of all the fours rolled

Fives: score is the sum of all the fives rolled

Sixes: score is the sum of all the sixes rolled

Three of a Kind: if there are three of a kind, score is the sum of all dice. Otherwise, the score is zero.

Four of a Kind: if there are four of a kind, score is sum of all dice. Otherwise, the score is zero.

Small Straight: if four consecutive numbers are rolled, score is 30. Otherwise, the score is zero.

Large Straight: if five consecutive numbers are rolled, score is 40. Otherwise, the score is zero.

Full House: if three of one number and two of another are rolled, score is 25. Otherwise, the score is zero.

Chance: score is sum of all dice

Yahtzee: if five of the same number is rolled, score is 50. Otherwise, the score is zero.

Summary of Development

Lines of Code: 351

This took about a week to develop. Since I did not use any functions or arrays, I wasn't able to completely do everything that I would have liked. For instance, there is no two player option (as of yet).

Using what I've read so far in the book, I developed the project in steps. I have five different versions of the project, each building on the previous one. For instance, version 2 has the user input the score for the category, while version 3 automatically calculates it. Likewise, version 3 only allows for the dice to be rolled once, while version 4 allows for the dice to be rerolled up to two times as the user chooses.

The hardest part of the project was simply working out various small bugs. It also took me a little while to figure out how to allow the user to reroll the dice up to twice, but to also allow them to not reroll the dice at all.

Overall, the project was fairly simple. It includes all of the concepts I have learned in the class before.

Example Inputs with Outputs

Output: Enter ‘r’ to roll the dice. Make sure you only enter one character.

Input: r

Output: You rolled:

Dice 1: 3

Dice 2: 6

Dice 3: 4

Dice 4: 3

Dice 5: 2

Would you like to reroll some dice? Enter 1 for yes and 0 for no.

Input: 1

Output: Which die would you like to reroll? Enter 1 for yes and 0 for no for each die, with a space between each.

i.e. to reroll dies 1, 2, and 5, but not 3 and 4, enter 1 1 0 0 1

make sure you only enter your answer is 0’s and 1’s

Input: 0 0 0 1 0

Output: Enter ‘r’ to roll the dice. Make sure you only enter one character

Input: r

Output: You rolled:

Dice 1: 3

Dice 2: 6

Dice 3: 4

Dice 4: 5

Dice 5: 2

Would you like to reroll some dice? Enter 1 for yes and 0 for no

Input: 0

Output: What category would you like to score in?

These are your available categories to score in:

ones

twos

threes

fours

fives

sixes

three of a kind

four of a kind

small straight

large straight

full house

chance

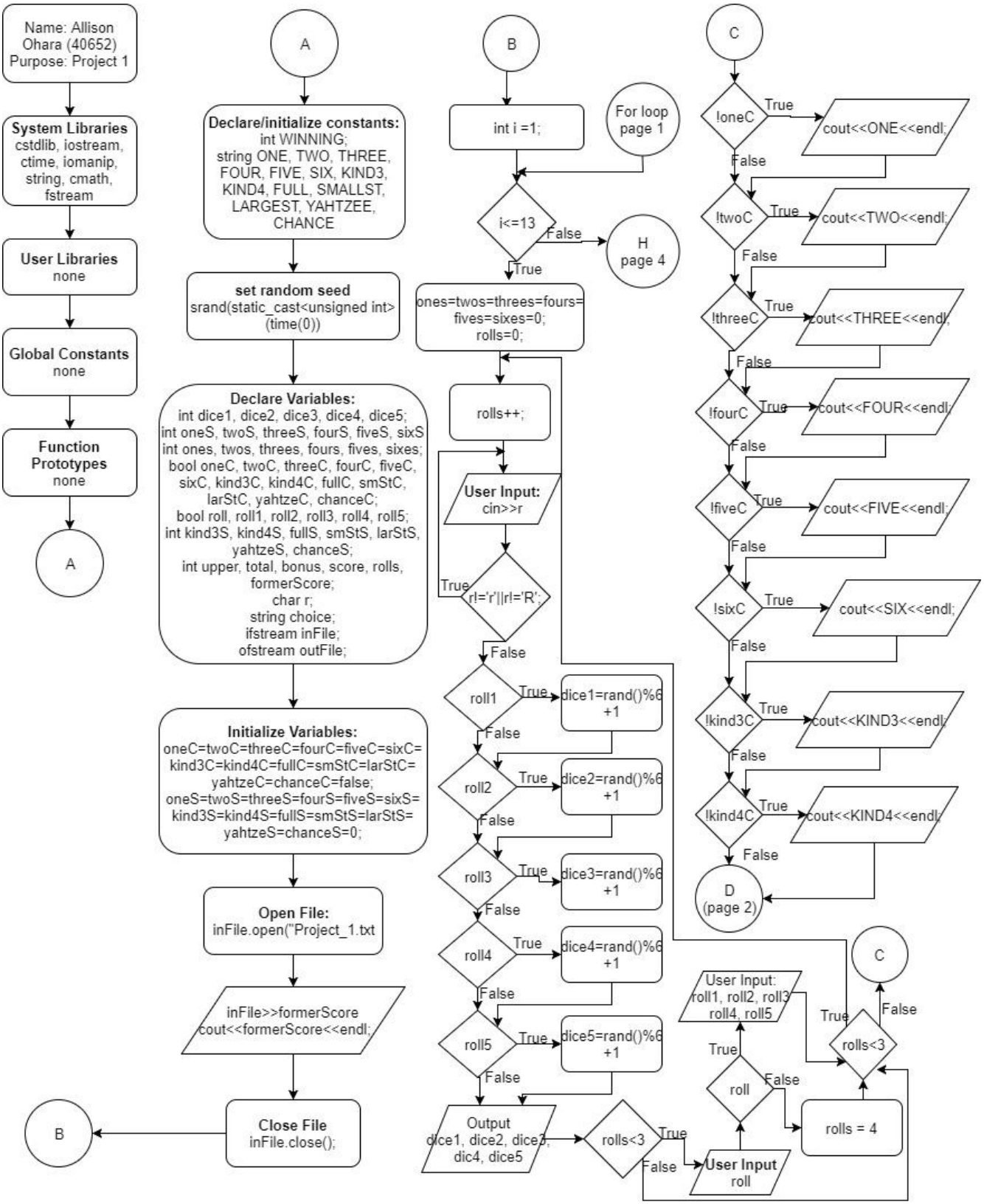
Yahtzee

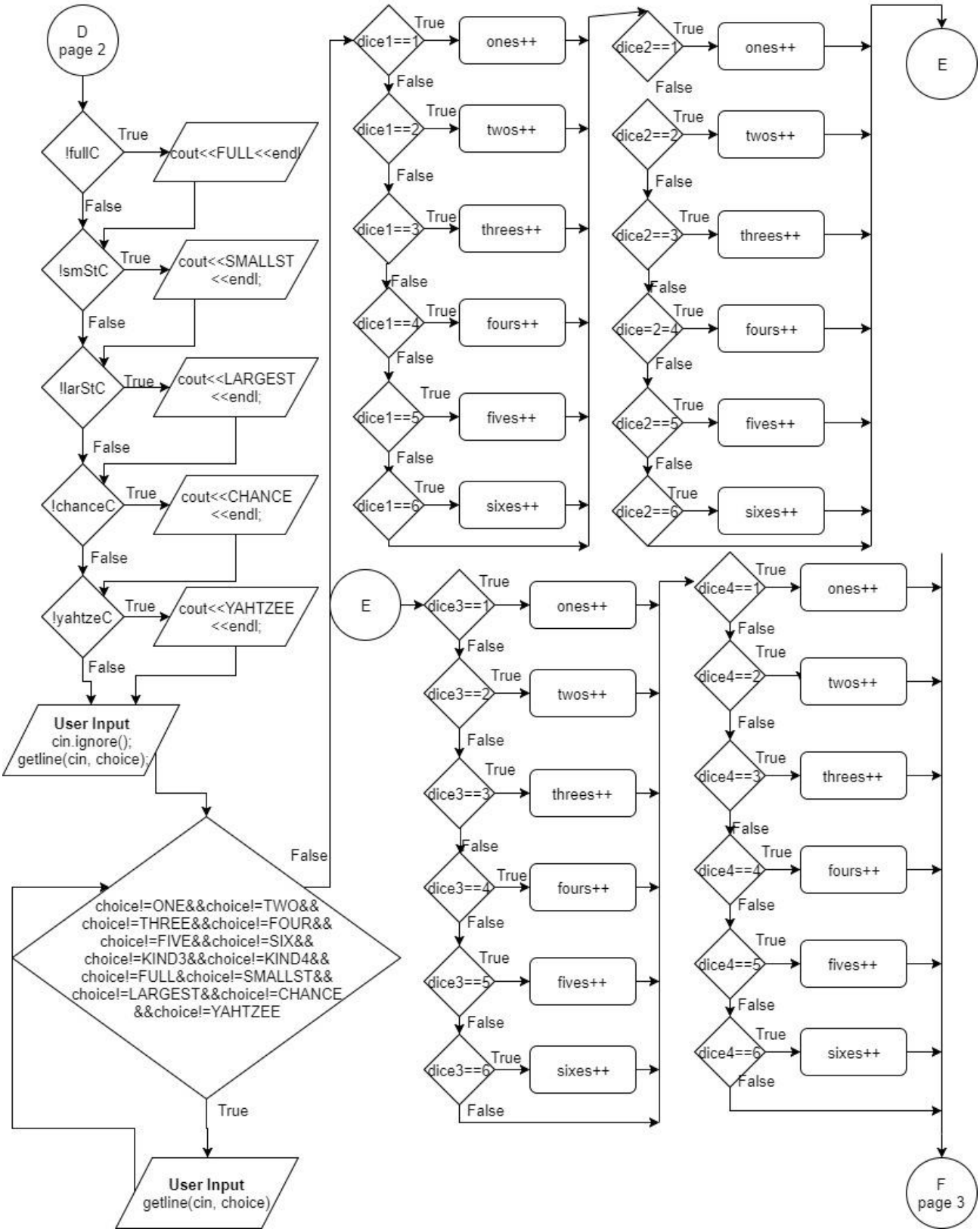
Input: large straight

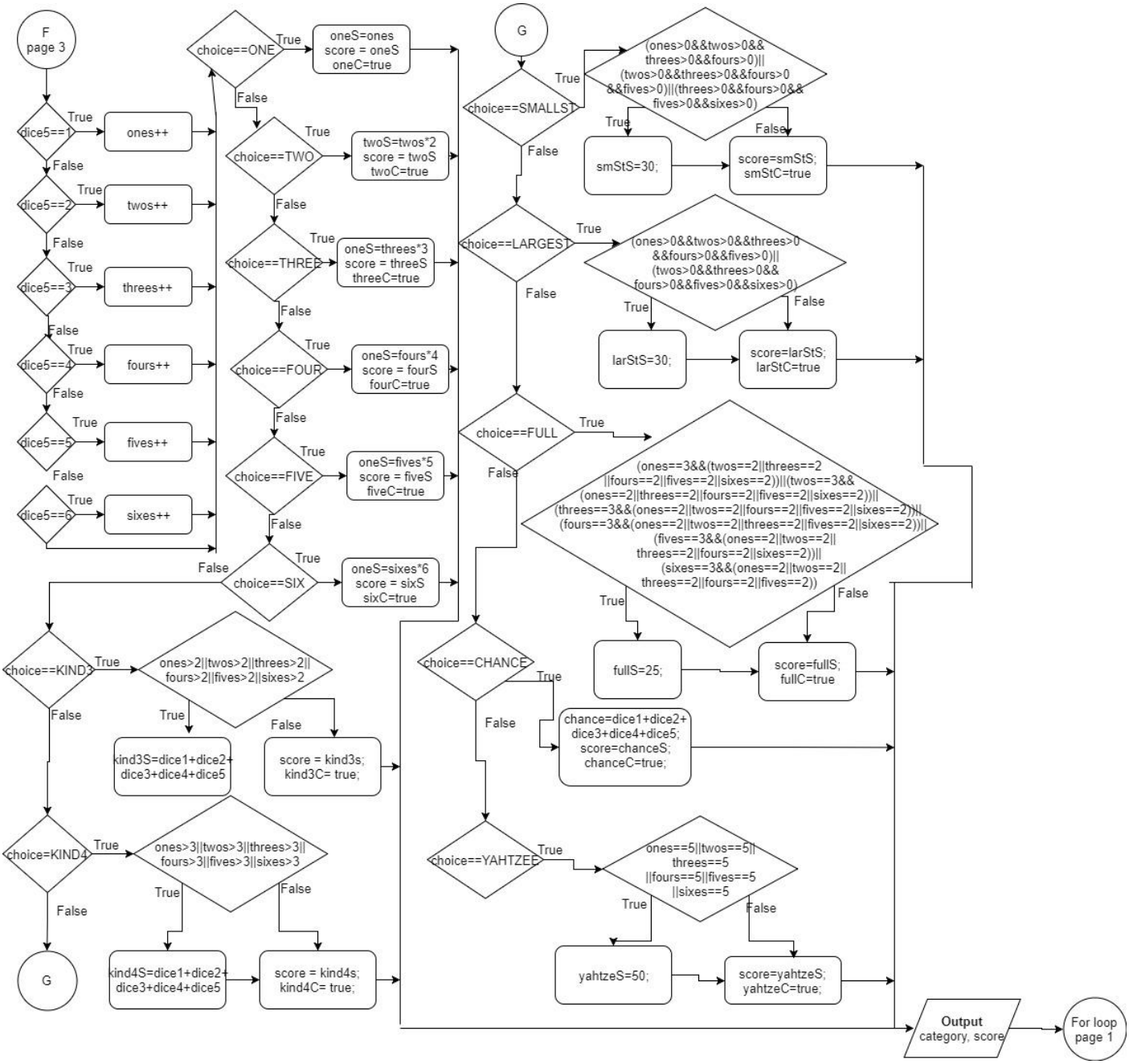
Output: Category: large straight Score: 40

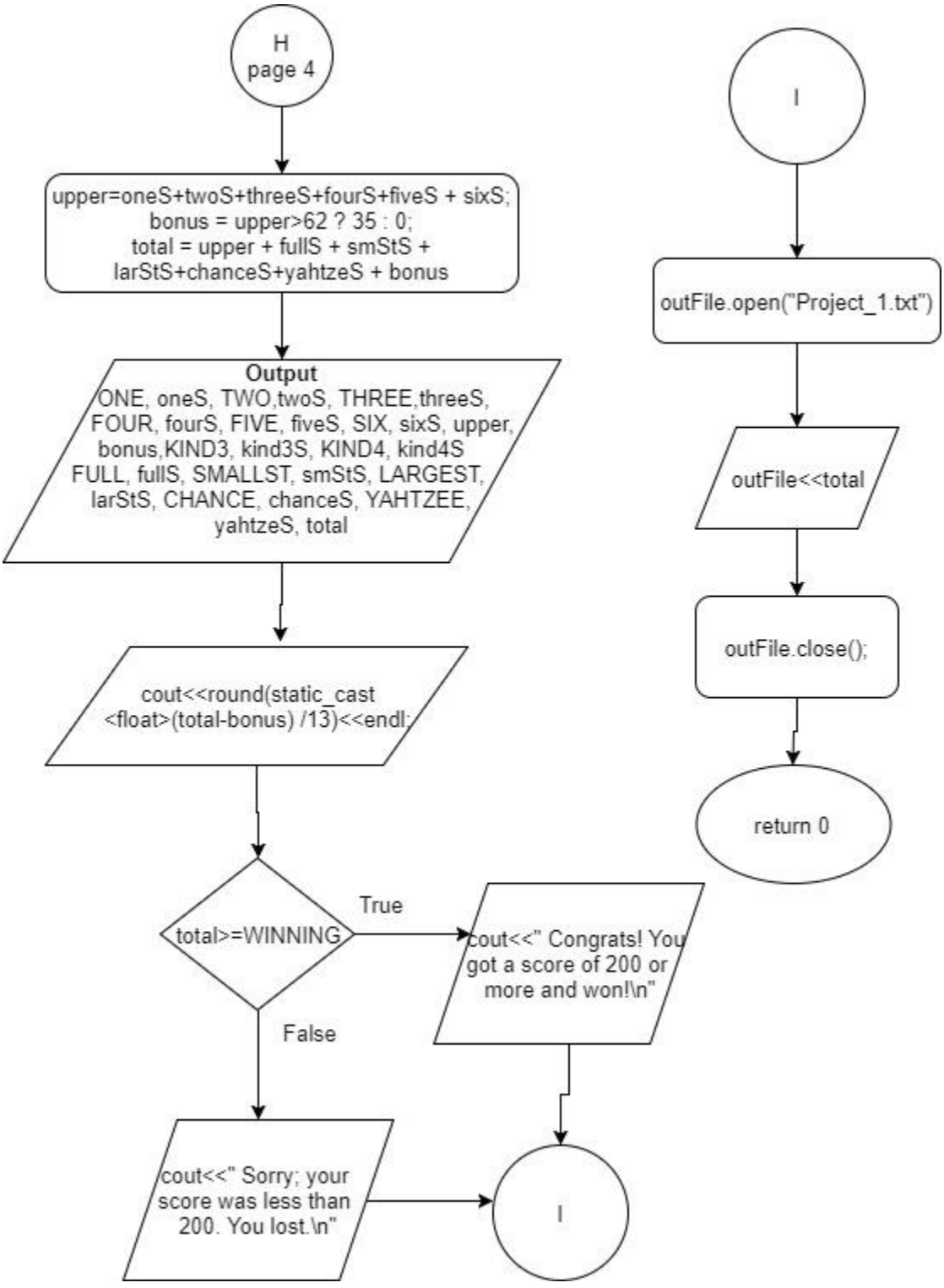
The program will continue for 12 more rounds in the same fashion.

Flowcharts


Name: Allison Ohara (40652)
Purpose: Project 1
System Libraries
cstdlib, iostream, ctime, iomanip, string, cmath, fstream
User Libraries
none
Global Constants
none
Function Prototypes
none
A
A
Declare/initialize constants:
int WINNING;
string ONE, TWO, THREE, FOUR, FIVE, SIX, KIND3, KIND4, FULL, SMALLST, LARGEST, YAHTZEE, CHANCE
set random seed
srand(static_cast<unsigned int>(time(0))
Declare Variables:
int dice1, dice2, dice3, dice4, dice5;
int oneS, twoS, threeS, fourS, fiveS, sixS
int ones, twos, threes, fours, fives, sixes;
bool oneC, twoC, threeC, fourC, fiveC, sixC, kind3C, kind4C, fullC, smStC, larStC, yahtzeC, chanceC;
bool roll, roll1, roll2, roll3, roll4, roll5;
int kind3S, kind4S, fullS, smStS, larStS, yahtzeS, chanceS;
int upper, total, bonus, score, rolls, formerScore;
char r;
string choice;
ifstream inFile;
ofstream outFile;
Initialize Variables:
oneC=twoC=threeC=fourC=fiveC=sixC=kind3C=kind4C=fullC=smStC=larStC=yahtzeC=chanceC=false;
oneS=twoS=threeS=fourS=fiveS=sixS=kind3S=kind4S=fullS=smStS=larStS=yahtzeS=chanceS=0;
Open File:
inFile.open("Project_1.txt
inFile>>formerScore
cout<<formerScore<<endl;
Close File
inFile.close();
B
B
int i =1;
For loop page 1
i<=13
False
H page 4
True
ones=twos=threes=fours=fives=sixes=0;
rolls=0;
rolls++;
User Input:
cin>>r
r!='r'||r!='R';
True
False
roll1
True
dice1=rand()%6+1
False
roll2
True
dice2=rand()%6+1
False
roll3
True
dice3=rand()%6+1
False
roll4
True
dice4=rand()%6+1
False
roll5
True
dice5=rand()%6+1
False
Output dice1, dice2, dice3, dic4, dice5
rolls<3
True
False
User Input
roll
True
roll
False
User Input:
roll1, roll2, roll3
roll4, roll5
rolls = 4
True
False
rolls<3
C
C
!oneC
True
cout<<ONE<<endl;
False
!twoC
True
cout<<TWO<<endl;
False
!threeC
True
cout<<THREE<<endl;
False
!fourC
True
cout<<FOUR<<endl;
False
!fiveC
True
cout<<FIVE<<endl;
False
!sixC
True
cout<<SIX<<endl;
False
!kind3C
True
cout<<KIND3<<endl;
False
!kind4C
True
cout<<KIND4<<endl;
False
D (page 2)


D
page 2
!fullC
True
cout<<FULL<<endl
False
!smStC
True
cout<<SMALLST
<<endl;
False
!larStC
True
cout<<LARGEST
<<endl;
False
!chanceC
True
cout<<CHANCE
<<endl;
False
!yahtzeC
True
cout<<YAHTZEE
<<endl;
False
User Input
cin.ignore();
getline(cin, choice);
choice!=ONE&&choice!=TWO&&
choice!=THREE&&choice!=FOUR&&
choice!=FIVE&&choice!=SIX&&
choice!=KIND3&&choice!=KIND4&&
choice!=FULL&choice!=SMALLST&&
choice!=LARGEST&&choice!=CHANCE
&&choice!=YAHTZEE
False
True
User Input
getline(cin, choice)
dice1==1
True
ones++
False
dice1==2
True
twos++
False
dice1==3
True
threes++
False
dice1==4
True
fours++
False
dice1==5
True
fives++
False
dice1==6
True
sixes++
dice2==1
True
ones++
False
dice2==2
True
twos++
False
dice2==3
True
threes++
False
dice=2=4
True
fours++
False
dice2==5
True
fives++
False
dice2==6
True
sixes++
E
E
dice3==1
True
ones++
False
dice3==2
True
twos++
False
dice3==3
True
threes++
False
dice3==4
True
fours++
False
dice3==5
True
fives++
False
dice3==6
True
sixes++
False
dice4==1
True
ones++
False
dice4==2
True
twos++
False
dice4==3
True
threes++
False
dice4==4
True
fours++
False
dice4==5
True
fives++
False
dice4==6
True
sixes++
False
F
page 3


F
page 3
dice5==1
True
ones++
False
dice5==2
twos++
dice5==3
threes++
dice5==4
fours++
dice5==5
fives++
dice5==6
sixes++
choice==ONE
oneS=ones
score = oneS
oneC=true
choice==TWO
twoS=twos*2
score = twoS
twoC=true
choice==THREE
oneS=threes*3
score = threeS
threeC=true
choice==FOUR
oneS=fours*4
score = fourS
fourC=true
choice==FIVE
oneS=fives*5
score = fiveS
fiveC=true
choice==SIX
oneS=sixes*6
score = sixS
sixC=true
G
choice==SMALLST
(ones>0&&twos>0&&
threes>0&&fours>0)||
(twos>0&&threes>0&&fours>0
&&fives>0)||(threes>0&&fours>0&&
fives>0&&sixes>0)
smStS=30;
score=smStS;
smStC=true
choice==LARGEST
(ones>0&&twos>0&&threes>0
&&fours>0&&fives>0)||
(twos>0&&threes>0&&
fours>0&&fives>0&&sixes>0)
larStS=30;
score=larStS;
larStC=true
choice==FULL
(ones==3&&(twos==2||threes==2
||fours==2||fives==2||sixes==2))||(twos==3&&
(ones==2||threes==2||fours==2||fives==2||sixes==2))||
(threes==3&&(ones==2||twos==2||fours==2||fives==2||sixes==2))||
(fours==3&&(ones==2||twos==2||threes==2||fives==2||sixes==2))||
(fives==3&&(ones==2||twos==2||
threes==2||fours==2||sixes==2))||
(sixes==3&&(ones==2||twos==2||
threes==2||fours==2||fives==2))
fullS=25;
score=fullS;
fullC=true
choice==CHANCE
chance=dice1+dice2+
dice3+dice4+dice5;
score=chanceS;
chanceC=true;
choice==YAHTZEE
ones==5||twos==5||
threes==5
||fours==5||fives==5
||sixes==5
yahtzeS=50;
score=yahtzeS;
yahtzeC=true;
choice==KIND3
ones>2||twos>2||threes>2||
fours>2||fives>2||sixes>2
kind3S=dice1+dice2+
dice3+dice4+dice5
score = kind3s;
kind3C= true;
choice=KIND4
ones>3||twos>3||threes>3||
fours>3||fives>3||sixes>3
G
kind4S=dice1+dice2+
dice3+dice4+dice5
score = kind4s;
kind4C= true;
Output
category, score
For loop
page 1


H
page 4
upper=oneS+twoS+threeS+fourS+fiveS + sixS;
bonus = upper>62 ? 35 : 0;
total = upper + fullS + smStS +
larStS+chanceS+yahtzeS + bonus
Output
ONE, oneS, TWO,twoS, THREE,threeS,
FOUR, fourS, FIVE, fiveS, SIX, sixS, upper,
bonus,KIND3, kind3S, KIND4, kind4S
FULL, fullS, SMALLST, smStS, LARGEST,
larStS, CHANCE, chanceS, YAHTZEE,
yahtzeS, total
cout<<round(static_cast
<float>(total-bonus) /13)<<endl;
total>=WINNING
True
cout<<" Congrats! You
got a score of 200 or
more and won!\n"
False
cout<<" Sorry; your
score was less than
200. You lost.\n"
I
I
outFile.open("Project_1.txt")
outFile<<total
outFile.close();
return 0

Pseudo-code

```
/*
 * File:   main.cpp
 * Author: Allison Ohara
 * Project 1: Yahtzee Psuedocode
 * Created on January 23, 2018, 11:33 AM
 * Updated on January 28, 2018,  9:32 PM
 */

// cstdlib, used for rand()
//iostream
//ctime, used to seed srand()
//iomanip, used to format output with setw
//string, used for strings
//cmath, used for round() function
//fstream, used for file I/O
//namespace

//main function
    //constant with score needed to win

    //make 13 string constants for yahtzee scoring categories

    //seed the random number generator

    //declare variables
    //numbers rolled on dice
    //upper section Yahtzee scores
```

```
//number of each number rolled in each round
//whether or not each category has already been chosen
//whether or not each die should be rolled again
//lower section Yahtzee scores
//scores, bonus score, and number of rolls used in round
//user-inputted character to roll the dice
//user inputted scoring category

//ifstream and ofstream for input/output file

//initialize bools at false so all categories are available
//initialize category scores at 0

//opening message

//open the input file
//read in the last score recorded from the file
//output the score
//close the file

//for loop to do 13 rounds, to fill the 13 categories
    //start quantity of each number rolled at 0 for each round
    //roll all the dice at beginning of each round, so set each bool for rolling each dice at true
    //set number of rolls used to 0 for beginning of each round

    //roll the dice up to three times

    //wait until user inputs r to roll dice
```

//roll the dice, all of them at first and then whichever one the user wants to the next 2 times

//get a random number 1-6 for die 1

//get a random number 1-6 for die 2

//get a random number 1-6 for die 3

//get a random number 1-6 for die 4

//get a random number 1-6 for die 5

//output rolled numbers

//allow user to roll again if he hasn't rolled 3 times yet

//allow user to choose which dice to reroll (if they choose to reroll)

//if they don't want to reroll, or have already rolled three times, move on to scoring

//list categories not already chosen

//allow user to input category choice

//validate user input; make sure their choice is an actual category

//see how many of each number was rolled on the dice, i.e. how many ones, twos, etc

//score the roll according to the user inputted choice, and cross the category off available category list

//if category choice is ones, score is how many ones were rolled

//cross category off list of available categories

//if category choice is twos, score is sum of all twos rolled

//cross category off list of available categories

//if category choice is threes, score is sum of all threes rolled

//cross category off list of available categories

//if category choice is fours, score is sum of all fours rolled

//cross category off list of available categories

//if category choice is fives, score is sum of all fives rolled

//cross category off list of available categories

//if category choice is sixes, score is sum of all sixes rolled

//cross category off list of available categories

//if category choice is three of a kind, see if there are three of a kind

//if so, score is sum of all die. Otherwise, score is zero.

//cross category off list of available categories

//if category choice is four of a kind, see if there are four of a kind

//if so score is sum of all die. Otherwise, score is zero.

//cross category off list of available categories

//if category choice is small straight, see if a small straight was rolled

//if so, score is 30. Otherwise, score is zero.

//cross category off list of available categories

```
        //if category choice is large straight, see if a large straight was rolled
        //If so, score is 40. Otherwise, score is zero.
        //cross category off list of available categories

        //if category choice is full house, see if a full house was rolled
        //if so, score is 25. Otherwise, score is zero
        //cross category off list of available categories

        //if category choice is chance, score is sum of all die
        //cross category off list of available categories

        //if category choice is yahtzee, see if five of the same number was rolled
        //if so, score is 50. Otherwise score is zero
        //cross category off list of available categories

    //output score
    //find upper score total
    //if the upper total is greater than 62, than user receives a 35 point bonus
    //find total score
    //output all categories and the scores they received

    //output total score
    //output average score in each category

    //if total score is 200 or better, user wins
    //otherwise, player loses
    //save score in file to be displayed at beginning of next game
//end program
```

Completed Check-Off Sheet

(also available as its own pdf in Project 1 folder)

Cross Reference for Project 1

You are to fill-in with where located in code

Note: For all topics, line #'s for examples (not all instances) are given

Chapter	Section	Topic	Where Line #'s	Pts	Notes
2	2	cout	92-97		
	3	libraries	9-15	8	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals	25-52		No variables in global area, failed project!
	5	identifiers	25-52		
	6	Integers	43-45, 50	3	
	7	Characters	51	3	
	8	Strings	27-39	3	
	9	Floats No Doubles	339	3	Using doubles will fail the project, floats OK!
	10	Bools	47-48	4	
	11	Sizeof *****			
	12	Variables 7 characters or less	42-52		All variables <= 7 characters
	13	Scope ***** No Global Variables			
	14	Arithmetic operators	231,314,		
	15	Comments 20%+	see "Pseudocode"	5	Model as pseudo code
	16	Named Constants	25-39		All Local, only Conversions/Physics/Math in Global area
	17	Programming Style ***** Emulate			Emulate style in book/in class repository
3	1	cin	78, 107		
	2	Math Expression	236, 339		
	3	Mixing data types *****			
	4	Overflow/Underflow *****			
	5	Type Casting	41, 339	4	
	6	Multiple assignment *****			
	7	Formatting output	318-333	4	
	8	Strings	27-39	3	
	9	Math Library	339	4	All libraries included have to be used
	10	Hand tracing ******			
4	1	Relational Operators			
	2	If	81, 117	4	Independent if
	4	If-else	102/109	4	
	5	Nesting	74/99/102	4	
	6	If-else-if	225,230,235	4	
	7	Flags *****			
	8	Logical operators	289-294	4	
	11	Validating user input	147	4	
	13	Conditional Operator	315	4	
	14	Switch	153-166	4	
5	1	Increment/Decrement	69, 75	4	
	2	While	147	4	
	5	Do-while	74/113	4	
	6	For loop	69	4	
	11	Files input/output both	65, 348	8	
	12	No breaks in loops ******			Failed Project if included
****** Not required to show			Total	100	