Project 1

Title:

Farkle

Course:

CSC-5

Section:

47993

Due Date:

November 7, 2018

Author:

Noel Verduzco

Summary of Game:

Farkle is played by two or more players, with each player in succession having a turn at throwing the dice. Each player's turn results in a score, and the scores for each player accumulate to some winning total (usually 10,000).

- At the beginning of each turn, the player throws all the dice at once.
- After each throw, one or more scoring dice must be set aside.
- The player may then either end their turn and bank the score accumulated so far, or continue to throw the remaining dice.
- If the player has scored all six dice, they have "hot dice" and may continue their turn with a new throw of all six dice, adding to the score they have already accumulated.
- There is no limit to the number of "hot dice" a player may roll in one turn.
- If none of the dice score in any given throw, the player has "farkled" and all points for that turn are lost.
- At the end of the player's turn, the dice are handed to the next player in succession (usually in clockwise rotation), and they have their turn.
- Once a player has achieved a winning point total, each other player has one last turn to score enough points to surpass that high-score.

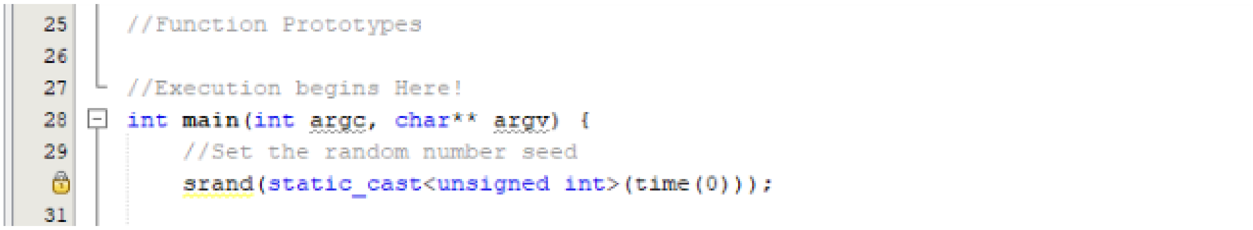
My version of the game is modified in that I have not implemented every single original aspect into the game. The scoring for my game is as follows:

- 100 points per X amount of 1’s
- 100 points per X amount of 5’s
- 200 points if the player has three 2’s
- 400 points if the player has three 3’s
- 600 points if the player has three 4’s
- 800 points if the player has three 6’s

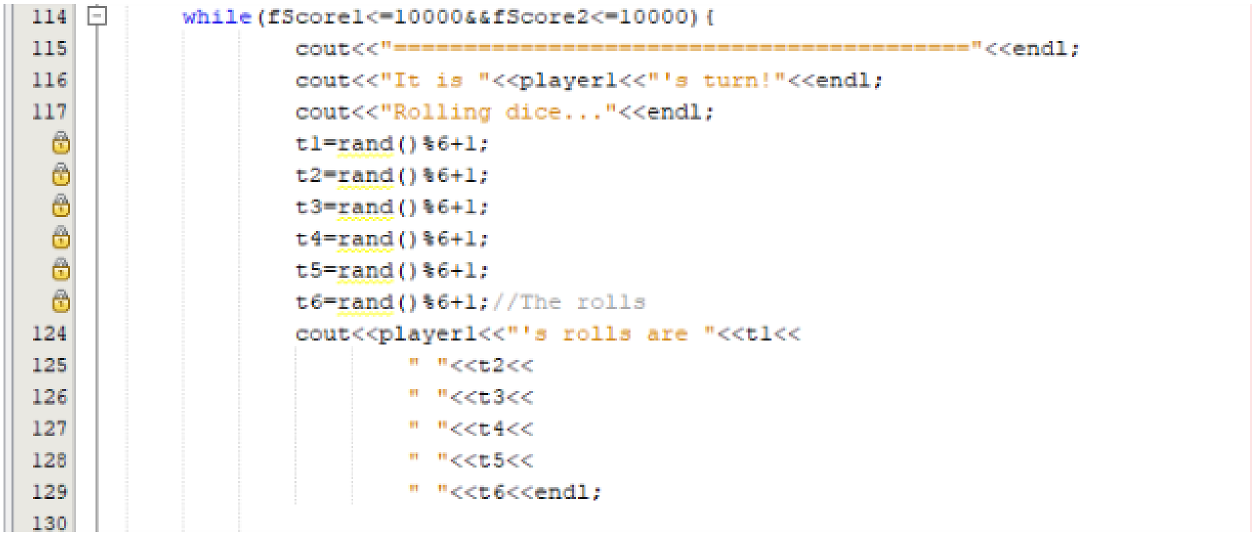
Alongside this, instead of losing all points for the player’s turn if they get no score for their dice roll, the player loses half their points.

Implementations:

This program primarily utilizes the C Standard Library and C Time Library which are responsible for the randomly generated numbers that give the die a value. The game would not be possible without this implementation considering Farkle’s gameplay revolves around rolling die randomly and deriving a player score from the values rolled.

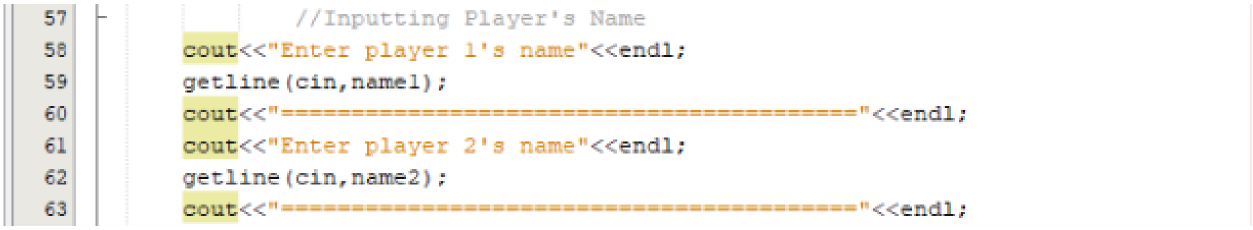


```
//Function Prototypes

//Execution begins Here!
int main(int argc, char** argv) {
    //Set the random number seed
    srand(static_cast<unsigned int>(time(0)));

```



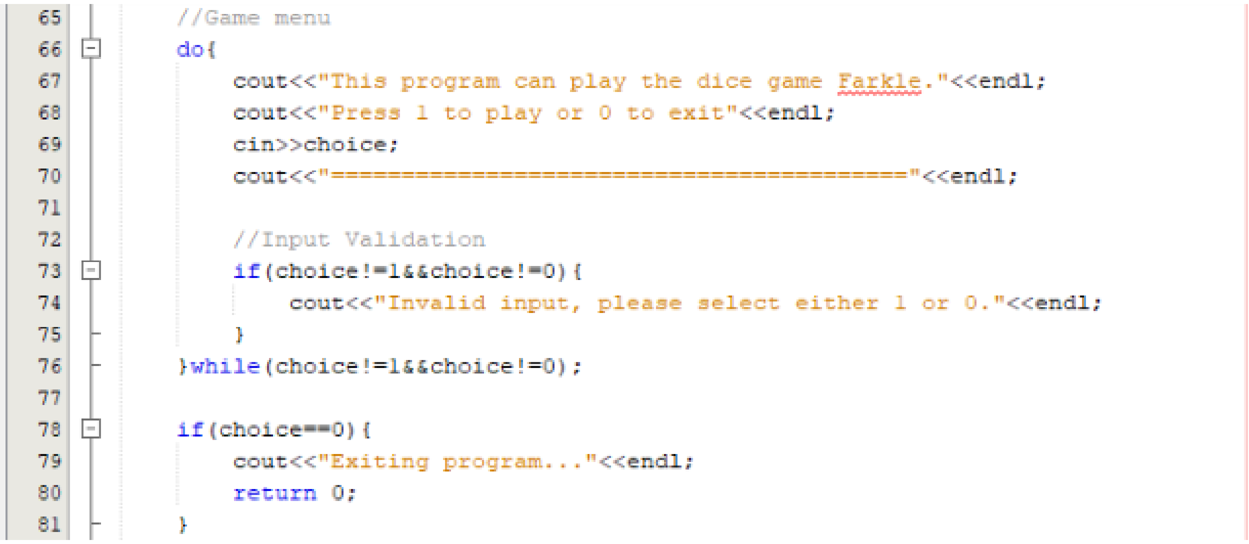
```
    while(fScore1<=10000&&fScore2<=10000){
            cout<<"=========================================="<<endl;
            cout<<"It is "<<player1<<"'s turn!"<<endl;
            cout<<"Rolling dice..."<<endl;
            t1=rand()%6+1;
            t2=rand()%6+1;
            t3=rand()%6+1;
            t4=rand()%6+1;
            t5=rand()%6+1;
            t6=rand()%6+1;//The rolls
            cout<<player1<<"'s rolls are "<<t1<<
                    "  "<<t2<<
                    "  "<<t3<<
                    "  "<<t4<<
                    "  "<<t5<<
                    "  "<<t6<<endl;

```

Before the random die are generated, the program asks for player 1’s and player 2’s names.



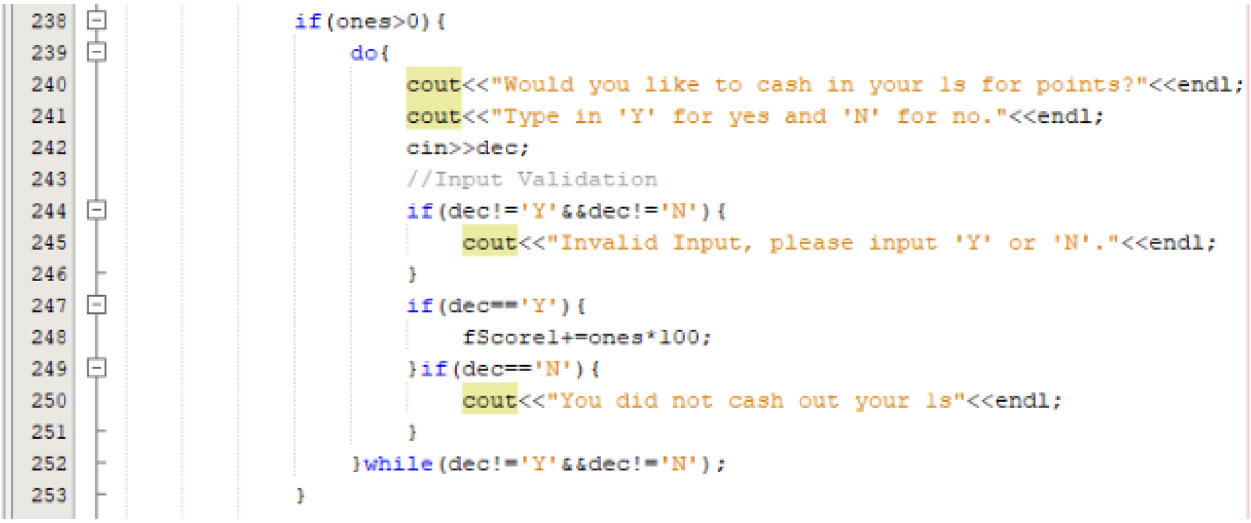
```
            //Inputting Player's Name
    cout<<"Enter player 1's name"<<endl;
    getline(cin,name1);
    cout<<"=========================================="<<endl;
    cout<<"Enter player 2's name"<<endl;
    getline(cin,name2);
    cout<<"=========================================="<<endl;
```

Another important implementation is the use of looping and nesting. The functionality of the program stems directly from the “while” and “do-while” loops that ensure input validation and proper scoring for the players. In the event that the user’s input is not desired, the user is notified of the error and told how to correct it. The user can also choose to continue through the program, or exit the game.



```
//Game menu
do{
    cout<<"This program can play the dice game Farkle."<<endl;
    cout<<"Press 1 to play or 0 to exit"<<endl;
    cin>>choice;
    cout<<"========================================="<<endl;

    //Input Validation
    if(choice!=1&&choice!=0){
        cout<<"Invalid input, please select either 1 or 0."<<endl;
    }
}while(choice!=1&&choice!=0);

if(choice==0){
    cout<<"Exiting program..."<<endl;
    return 0;
}
```

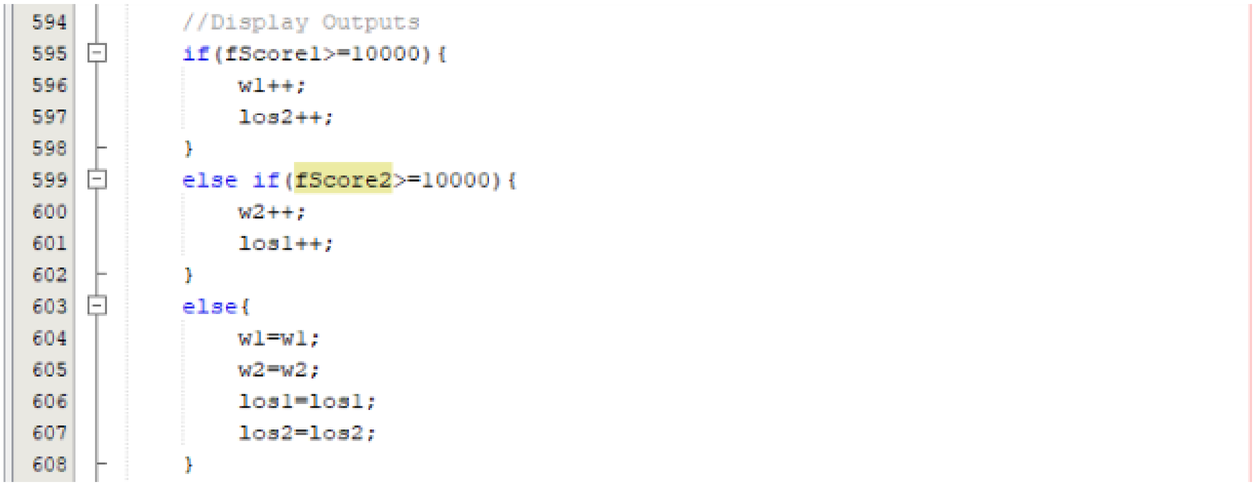
The several “do-while” loops organized throughout the main “while” loop serve to check the number of the same dice rolled for the player’s turn. After this, the loops prompt the user with the choice to cash in the points for an addition to their current score, or to forfeit the possible points for the rolled die in question.



```
if(ones>0){
    do{
        cout<<"Would you like to cash in your 1s for points?"<<endl;
        cout<<"Type in 'Y' for yes and 'N' for no."<<endl;
        cin>>dec;
        //Input Validation
        if(dec!='Y'&&dec!='N'){
            cout<<"Invalid Input, please input 'Y' or 'N'."<<endl;
        }
        if(dec=='Y'){
            fScore1+=ones*100;
        }if(dec=='N'){
            cout<<"You did not cash out your 1s"<<endl;
        }
    }while(dec!='Y'&&dec!='N');
}
```

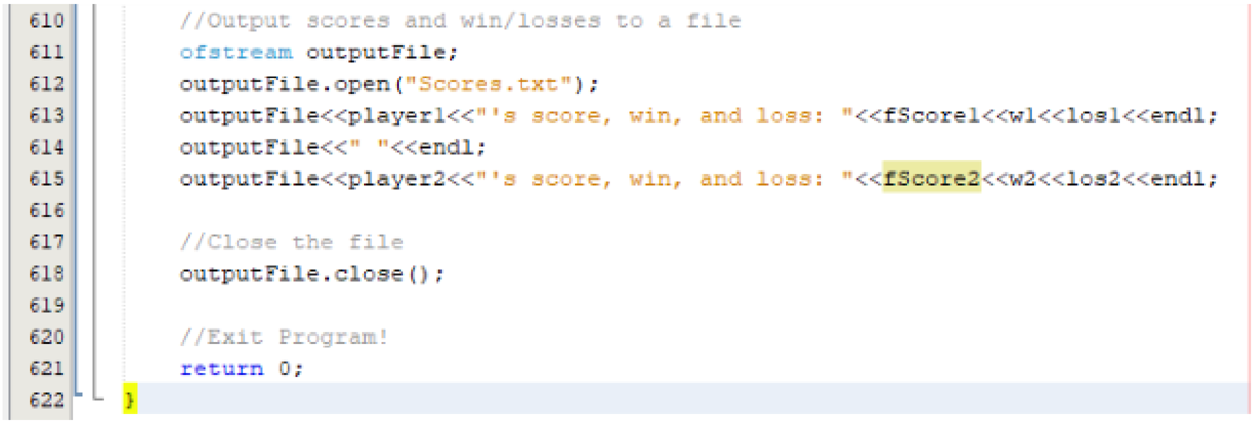
Similar iterations of the above loops are used to check the values for the twos, threes, fours, and sixes, and prompt the user with the previously mentioned choices.

After the scores have been added to, halved, or left unaffected, and the condition to jump out of the “while” loop is met (player 1 or player 2 must have accumulated 10000 or greater to exit the loop), a final score-check is made to assign player 1 and player 2 their respective win or loss for the game.



```
    //Display Outputs
    if(fScore1>=10000){
        w1++;
        los2++;
    }
    else if(fScore2>=10000){
        w2++;
        los1++;
    }
    else{
        w1=w1;
        w2=w2;
        los1=los1;
        los2=los2;
    }
```

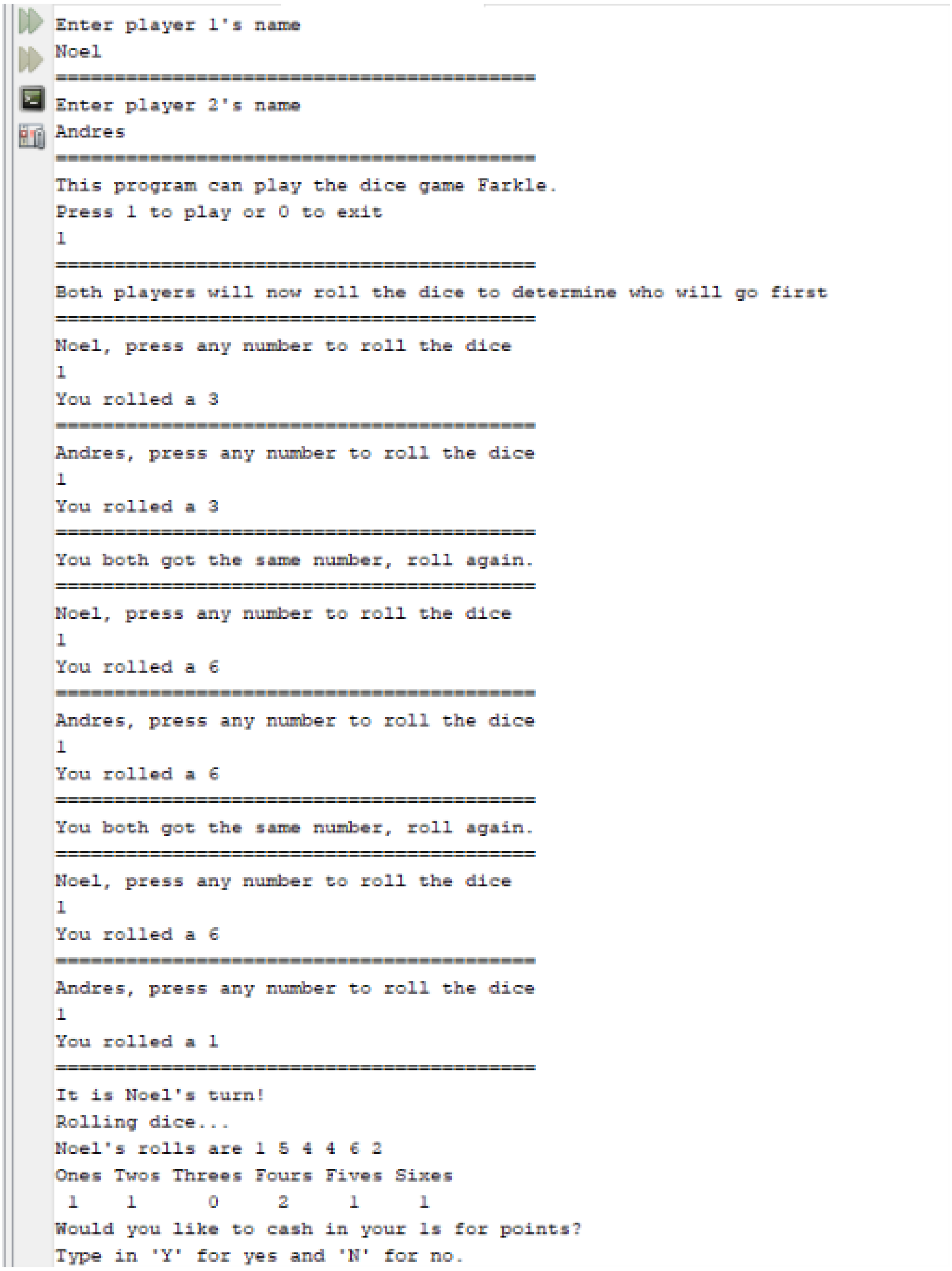
Finally, once the game has concluded, the program searches for a text file named “Scores.txt.,” and both player’s scores, wins, and losses are sent to that file in that order. After the information is sent, the file is closed and the program is terminated.



```
    //Output scores and win/losses to a file
    ofstream outputFile;
    outputFile.open("Scores.txt");
    outputFile<<player1<<"'s score, win, and loss: "<<fScore1<<w1<<los1<<endl;
    outputFile<<" "<<endl;
    outputFile<<player2<<"'s score, win, and loss: "<<fScore2<<w2<<los2<<endl;

    //Close the file
    outputFile.close();

    //Exit Program!
    return 0;
}
```

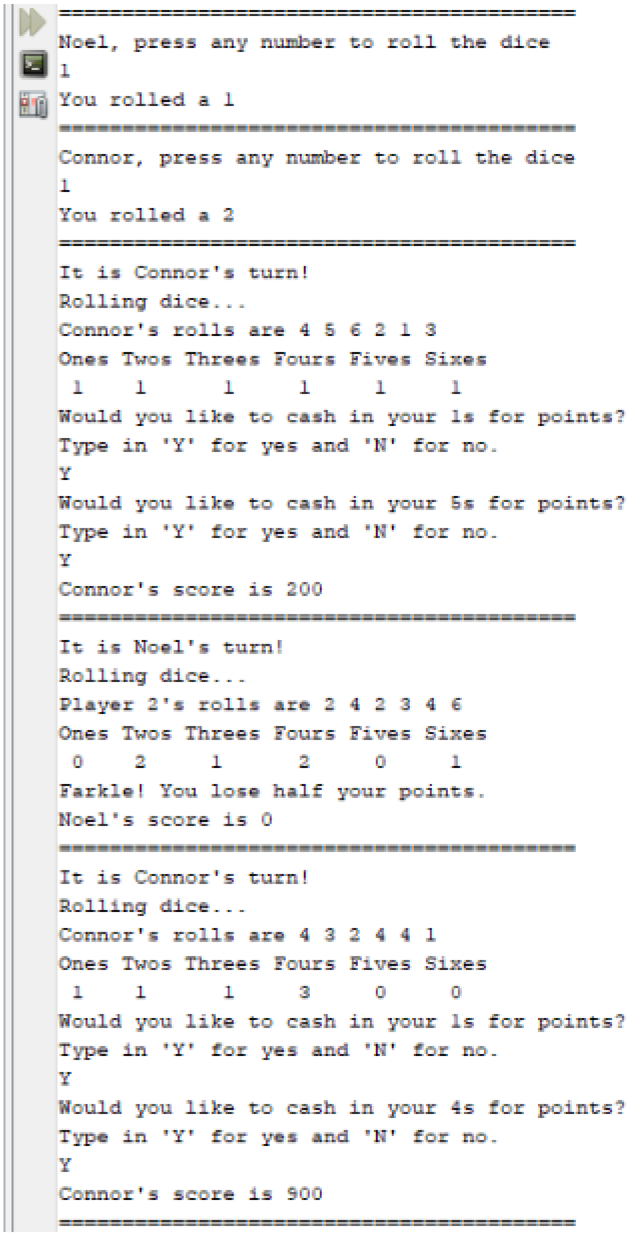
Examples of Input/Output:

Rolling the same die twice for both players:



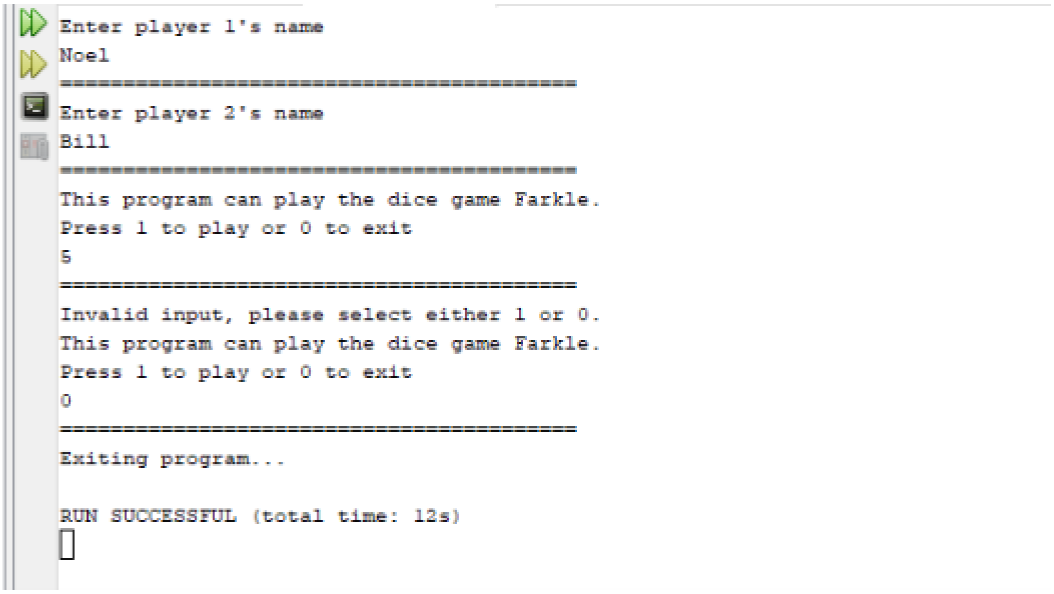
```
Enter player 1's name
Noel
=========================================
Enter player 2's name
Andres
=========================================
This program can play the dice game Farkle.
Press 1 to play or 0 to exit
1
=========================================
Both players will now roll the dice to determine who will go first
=========================================
Noel, press any number to roll the dice
1
You rolled a 3
=========================================
Andres, press any number to roll the dice
1
You rolled a 3
=========================================
You both got the same number, roll again.
=========================================
Noel, press any number to roll the dice
1
You rolled a 6
=========================================
Andres, press any number to roll the dice
1
You rolled a 6
=========================================
You both got the same number, roll again.
=========================================
Noel, press any number to roll the dice
1
You rolled a 6
=========================================
Andres, press any number to roll the dice
1
You rolled a 1
=========================================
It is Noel's turn!
Rolling dice...
Noel's rolls are 1 5 4 4 6 2
Ones Twos Threes Fours Fives Sixes
 1    1      0     2     1     1
Would you like to cash in your 1s for points?
Type in 'Y' for yes and 'N' for no.
```

Starting the first couple of turns and adding, halving, or leaving score alone:



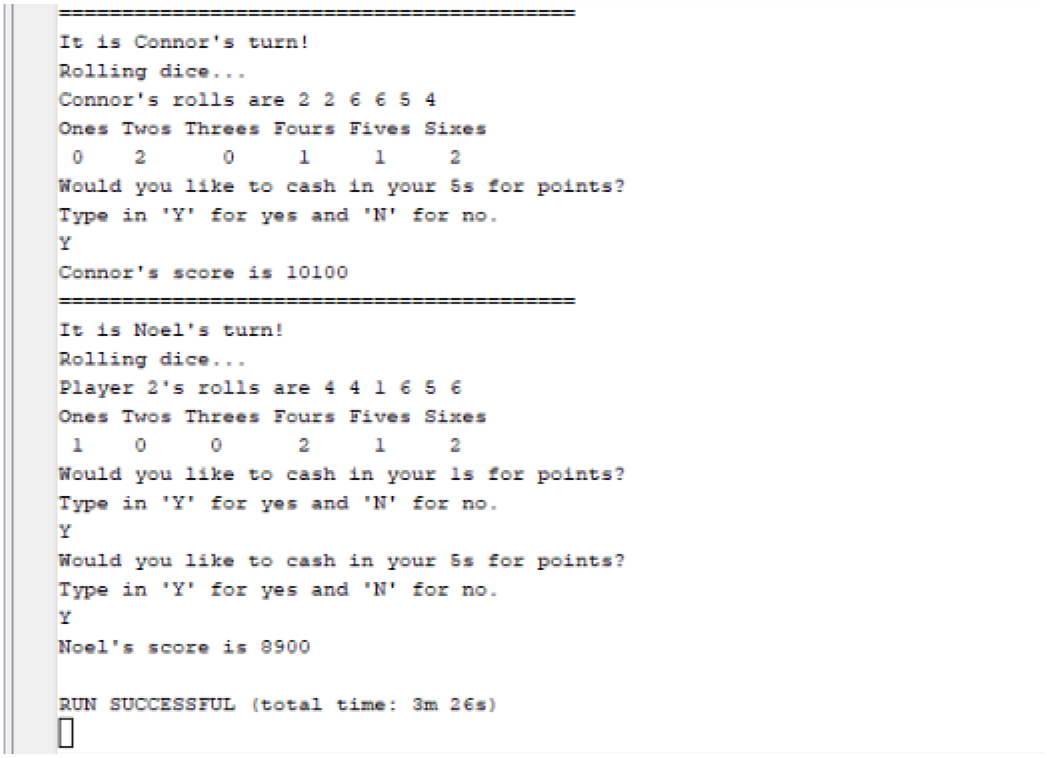
```
=========================================
Noel, press any number to roll the dice
1
You rolled a 1
=========================================
Connor, press any number to roll the dice
1
You rolled a 2
=========================================
It is Connor's turn!
Rolling dice...
Connor's rolls are 4 5 6 2 1 3
Ones Twos Threes Fours Fives Sixes
 1    1      1     1     1     1
Would you like to cash in your 1s for points?
Type in 'Y' for yes and 'N' for no.
Y
Would you like to cash in your 5s for points?
Type in 'Y' for yes and 'N' for no.
Y
Connor's score is 200
=========================================
It is Noel's turn!
Rolling dice...
Player 2's rolls are 2 4 2 3 4 6
Ones Twos Threes Fours Fives Sixes
 0    2     1      2     0     1
Farkle! You lose half your points.
Noel's score is 0
=========================================
It is Connor's turn!
Rolling dice...
Connor's rolls are 4 3 2 4 4 1
Ones Twos Threes Fours Fives Sixes
 1    1      1     3     0     0
Would you like to cash in your 1s for points?
Type in 'Y' for yes and 'N' for no.
Y
Would you like to cash in your 4s for points?
Type in 'Y' for yes and 'N' for no.
Y
Connor's score is 900
=========================================
```

Entering invalid input and exiting the game:



```
Enter player 1's name
Noel
==========================================
Enter player 2's name
Bill
==========================================
This program can play the dice game Farkle.
Press 1 to play or 0 to exit
5
==========================================
Invalid input, please select either 1 or 0.
This program can play the dice game Farkle.
Press 1 to play or 0 to exit
0
==========================================
Exiting program...

RUN SUCCESSFUL (total time: 12s)
```

Winning after many turns later:



```
==========================================
It is Connor's turn!
Rolling dice...
Connor's rolls are 2 2 6 6 5 4
Ones Twos Threes Fours Fives Sixes
 0    2      0     1     1     2
Would you like to cash in your 5s for points?
Type in 'Y' for yes and 'N' for no.
Y
Connor's score is 10100
==========================================
It is Noel's turn!
Rolling dice...
Player 2's rolls are 4 4 1 6 5 6
Ones Twos Threes Fours Fives Sixes
 1    0     0      2     1     2
Would you like to cash in your 1s for points?
Type in 'Y' for yes and 'N' for no.
Y
Would you like to cash in your 5s for points?
Type in 'Y' for yes and 'N' for no.
Y
Noel's score is 8900

RUN SUCCESSFUL (total time: 3m 26s)
```