Farkle
Dice Game

Noel Verduzco
CSC 5
Fall 2018
47993

Table of Contents

Introduction

How the Dice Game Works

Object of the Game:

To accumulate 10,000 points or more.

Rules of the Game:

Farkle is typically a two-to-four person game. The rules are as follows:

- 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 (see the scoring guide below).

- 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.

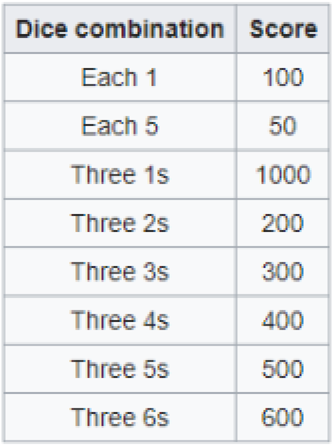
- If none of the dice score in any given throw, the player has “Farkled” and all points for that turn are lost.

- At the end of the player’s turn, the dice are handed to the next player in succession (usually in a 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, and claim the win for themselves.

Scoring of the Game:

The following scores for single dice or combinations of dice are widely established, in that they are common to all, or nearly all, of the above-cited descriptions of Farkle scoring.



Dice combination	Score
Each 1	100
Each 5	50
Three 1s	1000
Three 2s	200
Three 3s	300
Three 4s	400
Three 5s	500
Three 6s	600

For example, if a player throws 1-2-3-3-3-4, they could do any of the following:

- Score three 3s as a 300 and then throw the remaining three dice
- Score the single 1 as 100 and then throw the remaining five dice
- Score the single 5 as 50 and then throw the remaining five dice
- Score three 3s, the single 1, and the single 5 for a total of 450 and then throw the remaining die
- Score three 3s, the single 1, and the single 5 for a total of 450 and stop, banking 450 points in that turn

This is not an exhaustive list of plays based on that throw, but it covers the most likely ones. If the player continues throwing, as in any of the above cases except the last, they risk farkling and thus losing all accumulated points. On the other hand, if they score five dice and have only one die to throw, they have a 1 in 3 chance of scoring a single 1 or a single 5, and then having scored all six dice they will have "hot dice" and can throw all six dice again to further increase their score.

Each scoring combination must be achieved in a single throw. For example, if a player has already set aside two individual 1s and then throws a third with the four dice remaining, they do not have a triplet of 1s for a score of 1000 but merely three individual 1s for a score of 300.

Equipment Needed:

- Dice (6, or 5 in some variations)

- Paper and a pencil or pen for score-keeping

My Approach to the Game

Translating Gameplay Rules into Programming Language:

My first time thinking about how to implement the game's mechanics into code was a challenge. For the first-half of the semester, the tools I could utilize were limited and primitive. After building a complete, but highly inefficient game for Project 1, I asked myself, "How can I make the code more efficient?"

- Utilize functions for tasks that are called multiple times

- Utilize functions for a more modular and organized program

- Utilize arrays for a simpler and quicker dice rolling

I reviewed the old code I used to create Farkle and began rewriting that code into functions that could be reused for player 1's and player 2's various values. Sections of the program that took several hundred lines of code to execute the same function, were now condensed into less than fifty. The beauty of utilizing functions meant I only needed to call a function once with a single line of code wherever needed, with different function parameters for each player's variables, instead of pasting the same chunks of code several times over. The same can be said for the array I created that handles almost one-hundred dice rolls for both players.

Similarities to the Physical Game:

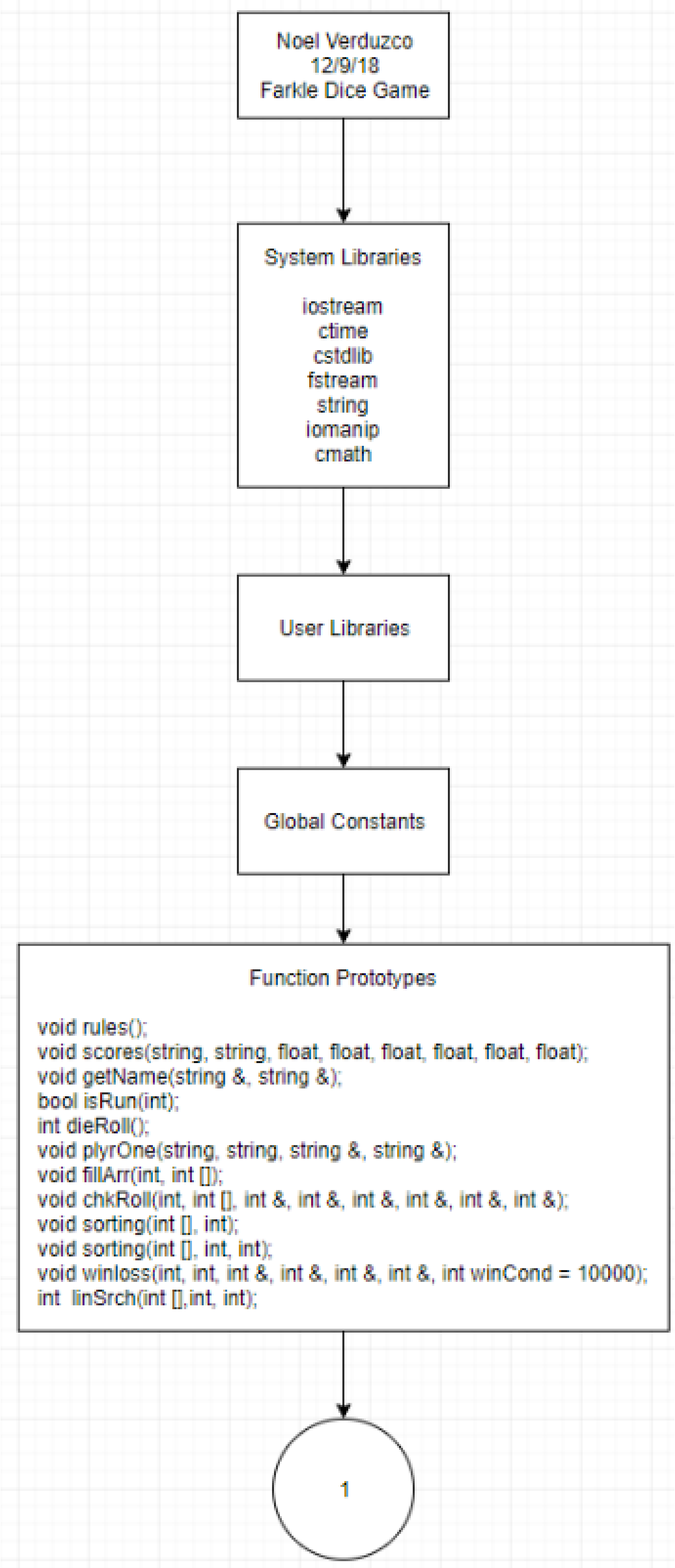
My program follows many of the same core mechanics such as rolling six dice, banking or tossing points, and reaching the 10,000 point limit for a win.

Differences from the Physical Game:

My program uses slightly altered scoring and rolling mechanics. To simplify and quicken the gameplay, I allow the players to bank every possible combination of dice possible during their turn, but only for that turn (instead of rerolling the remaining dice for another chance at points or a Farkle). Not only this, but instead of losing 1000 points for three Farkles in-a-row, I instead halve the player's total points. Although this is more detrimental to the player's score, I make up for this loss by giving back in multiple claims of dice combinations for their turn.

The Logic of it All

Flowchart with Psudocode:

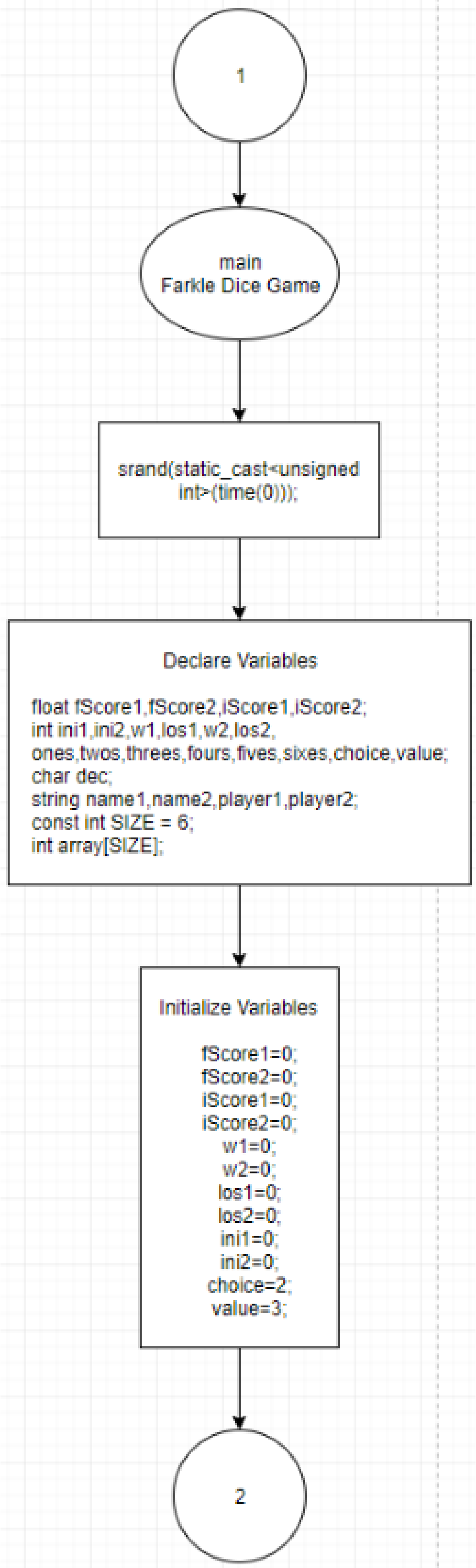


```
void rules();
void scores(string, string, float, float, float, float, float, float);
void getName(string &, string &);
bool isRun(int);
int dieRoll();
void plyrOne(string, string, string &, string &);
void fillArr(int, int []);
void chkRoll(int, int [], int &, int &, int &, int &, int &, int &);
void sorting(int [], int);
void sorting(int [], int, int);
void winloss(int, int, int &, int &, int &, int &, int winCond = 10000);
int  linSrch(int [],int, int);
```

The program reads through my function prototypes and makes sure to initialize all of the arguments and code that will follow

The random number seed is generated

All of main’s variables are declared and initialized once



```
srand(static_cast<unsigned
int>(time(0)));
```

```
float fScore1,fScore2,iScore1,iScore2;
int ini1,ini2,w1,los1,w2,los2,
ones,twos,threes,fours,fives,sixes,choice,value;
char dec;
string name1,name2,player1,player2;
const int SIZE = 6;
int array[SIZE];
```

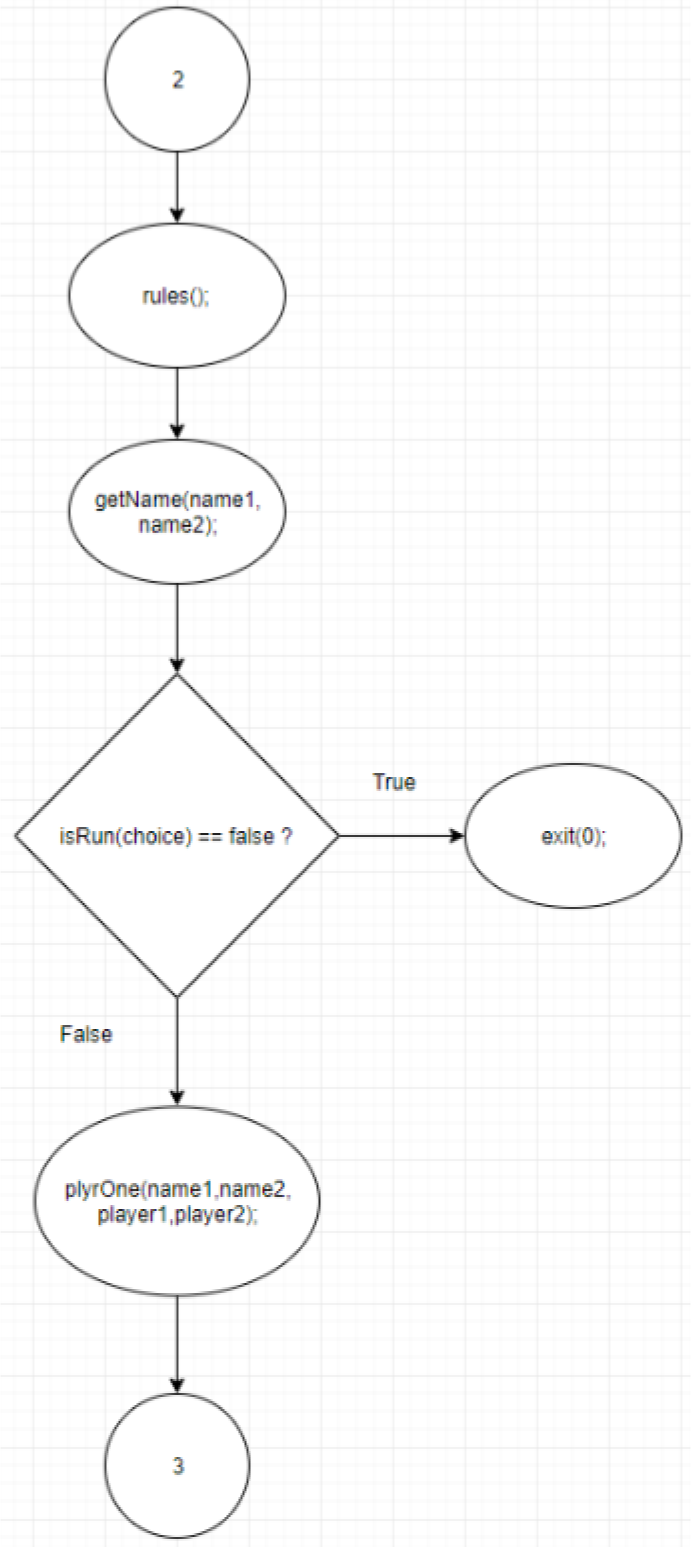
```
fScore1=0;
fScore2=0;
iScore1=0;
iScore2=0;
w1=0;
w2=0;
los1=0;
los2=0;
ini1=0;
ini2=0;
choice=2;
value=3;
```

The rules(); function is called, which displays the text from a text file in the program's folder

getName(); is called, making the program ask the user for the names of both players

The function isRun(); is called in order to ask the user whether they would like to proceed with starting the game, or exiting the program

If the user decides to play, plyrOne(); is called, which allows both players to roll one die each and determine who will go first based upon the highest die roll



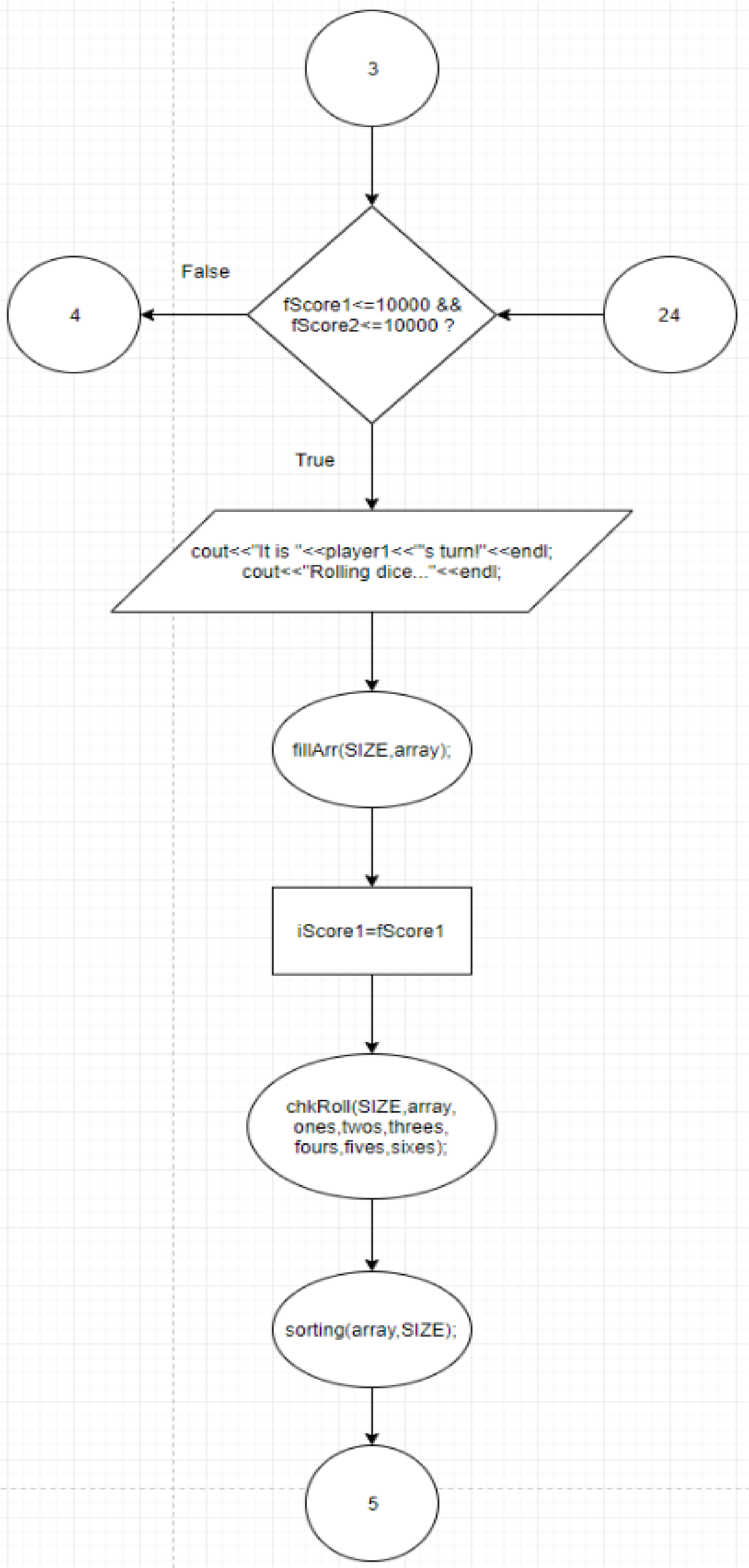
Both players' final score is checked in a while-loop to decide whether the game is won or not

The program displays player 1's name and rolls their dice in the function fillArr();

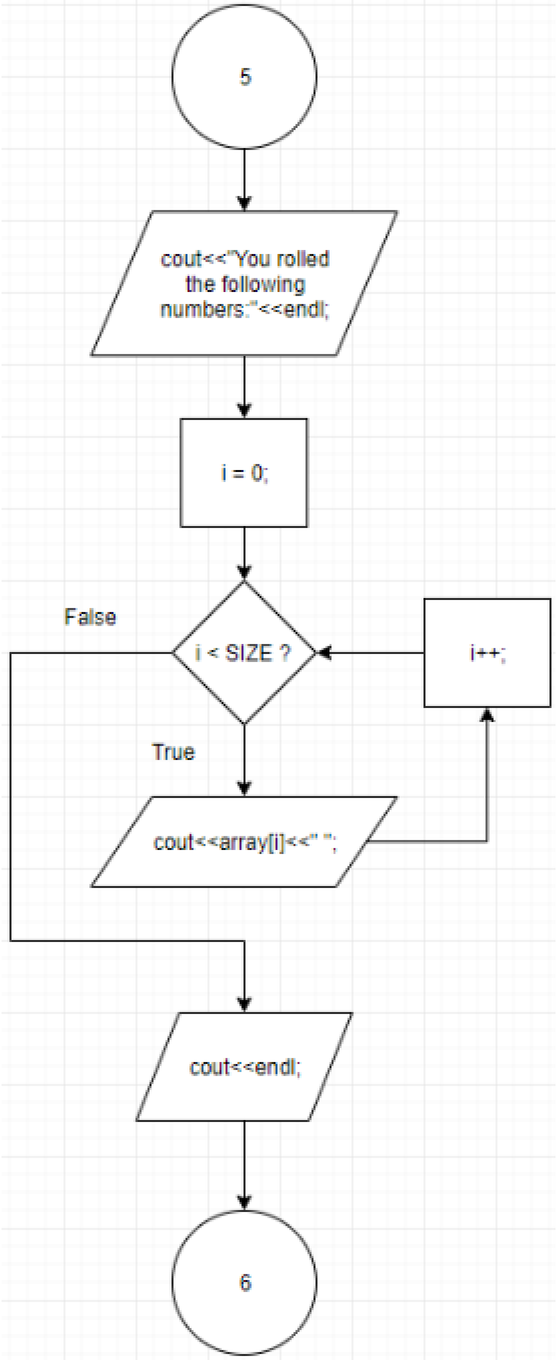
Player 1's initial score is set to their final score to determine whether they have farkled or not by the end of their turn

The function chkRoll(); is called to determine how many of the same type of dice were rolled

sorting(); is then called to organize the dice rolled in ascending order



The program displays the dice rolled in ascending order in a short for-loop

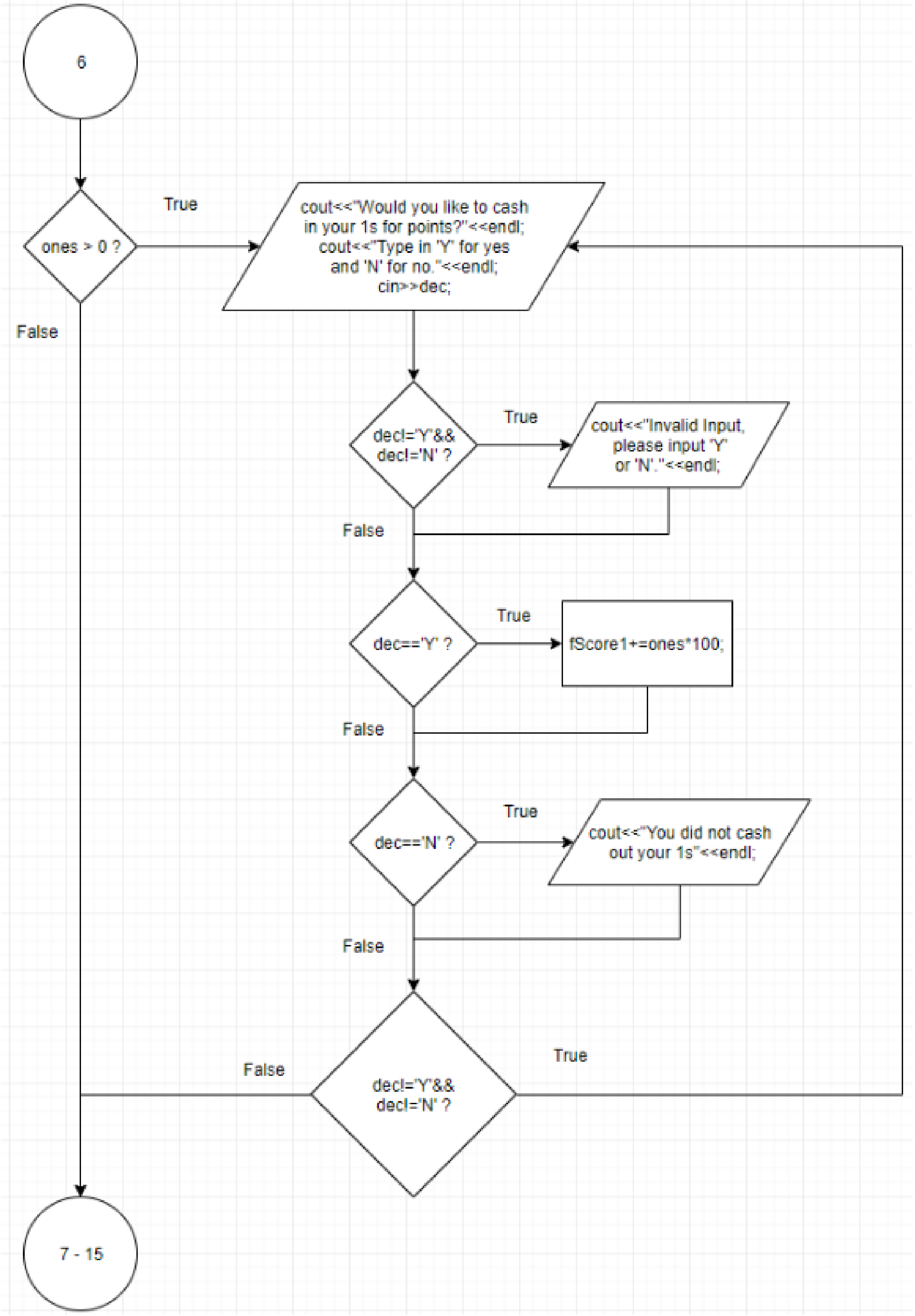


Once the number of same dice is counted, the program runs seven if-statements to ask the user whether they would like to add points to their score, or pass

The format is similar for each if-statement, minus the difference in the testing condition

I.e. "ones > 0 ?"

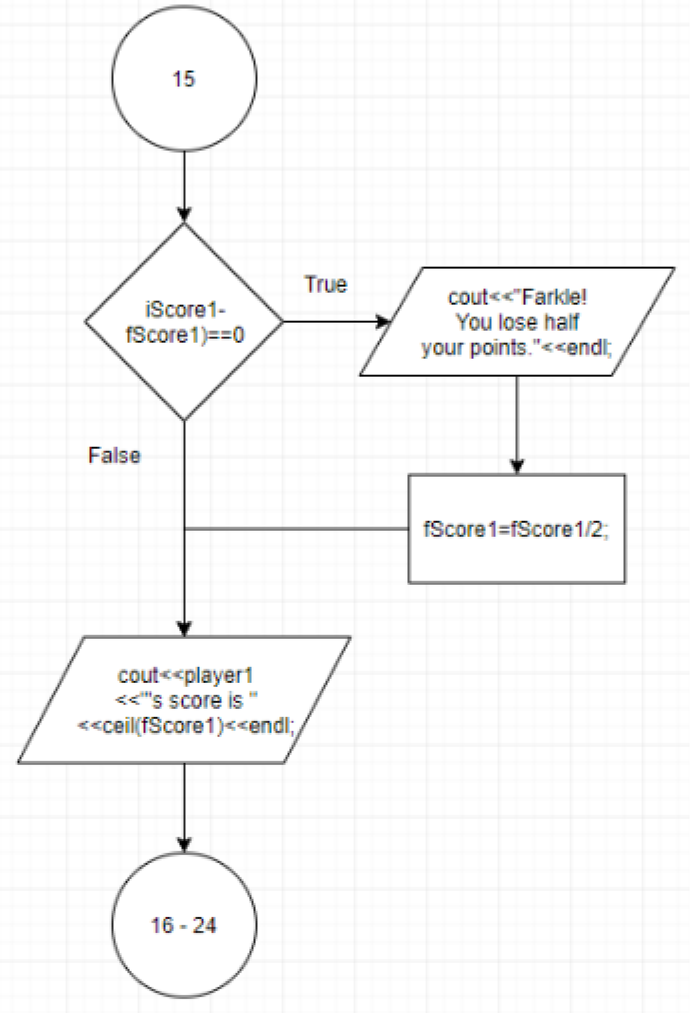
Each condition is checked based upon the scoring guide stated in the beginning of the program


6
ones > 0 ?
True
False
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;
dec!='Y'&& dec!='N' ?
True
cout<<"Invalid Input, please input 'Y' or 'N'."<<endl;
False
dec=='Y' ?
True
fScore1+=ones*100;
False
dec=='N' ?
True
cout<<"You did not cash out your 1s"<<endl;
False
False
dec!='Y'&& dec!='N' ?
True
7 - 15

The program checks to see if player 1 has earned any points during their turn. If they have not earned any points that turn, they farkle, and lose half of their total points

After this, all of the same exact code is ran for player 2. Once player 1 and player 2 have their scores for their respective turns, the while-loop mentioned earlier is checked again to see if either player has reached at least 10000 points

If it has not been reach, the players play another full turn. If it has been reached, the program jumps out of the loop and proceeds to the last section below

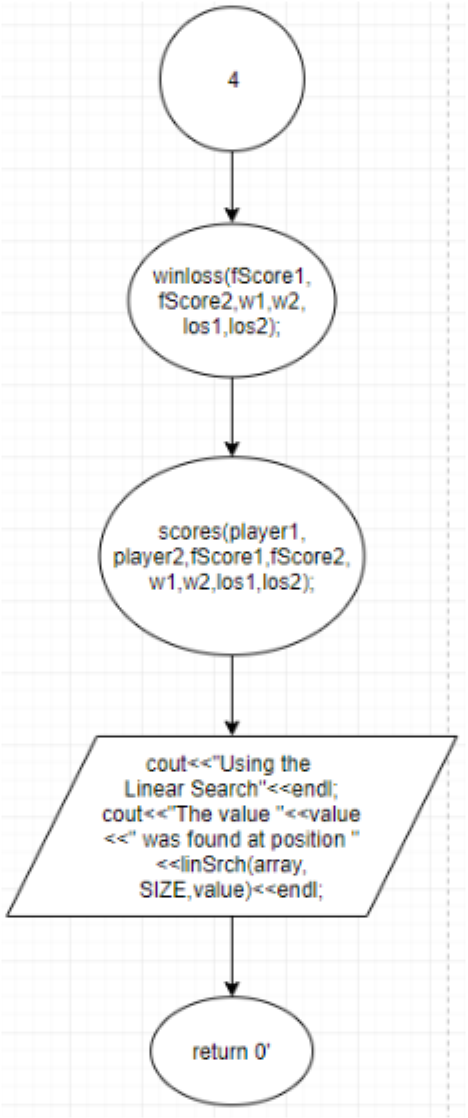

15
iScore1-fScore1)==0
True
cout<<"Farkle! You lose half your points."<<endl;
False
fScore1=fScore1/2;
cout<<player1 <<"'s score is " <<ceil(fScore1)<<endl;
16 - 24

The function winloss(); determines which player is given a win and which player is given a loss

scores(); is called immediately after to send the final scores, wins, and losses to a text file in the program's folder

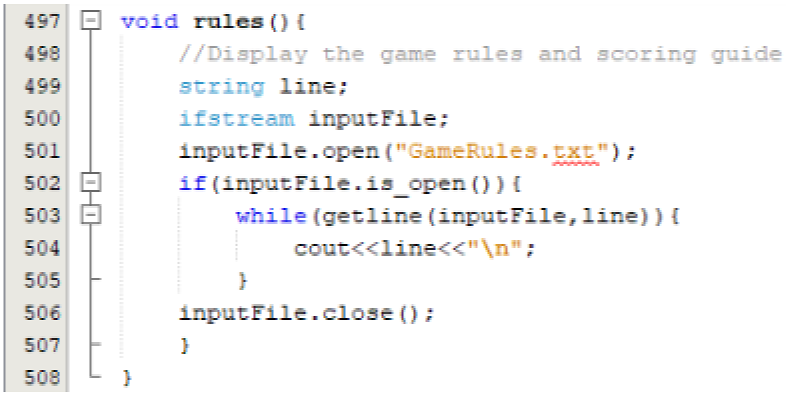
A random linear search is used to see if the array that holds the stored dice has the number "3" inside of it, and tells of its position in said array

The program exits after all is done

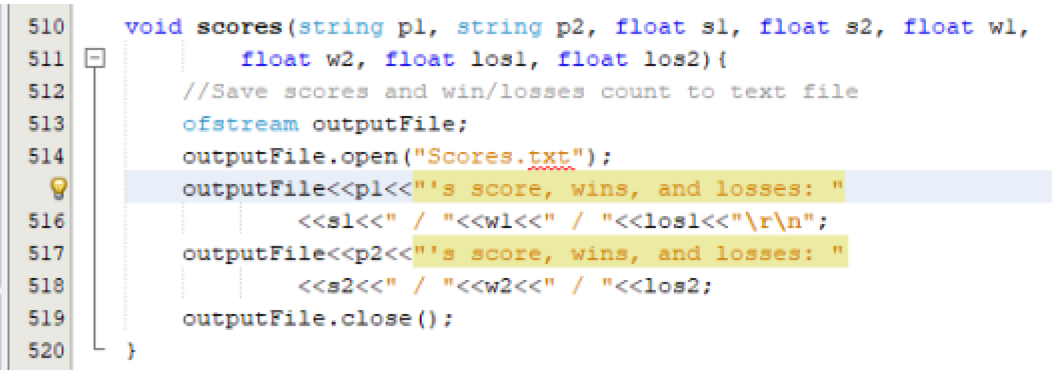

4
winloss(fScore1, fScore2,w1,w2, los1,los2);
scores(player1, player2,fScore1,fScore2, w1,w2,los1,los2);
cout<<"Using the Linear Search"<<endl; cout<<"The value "<<value <<" was found at position " <<linSrch(array, SIZE,value)<<endl;
return 0'

Program Functions

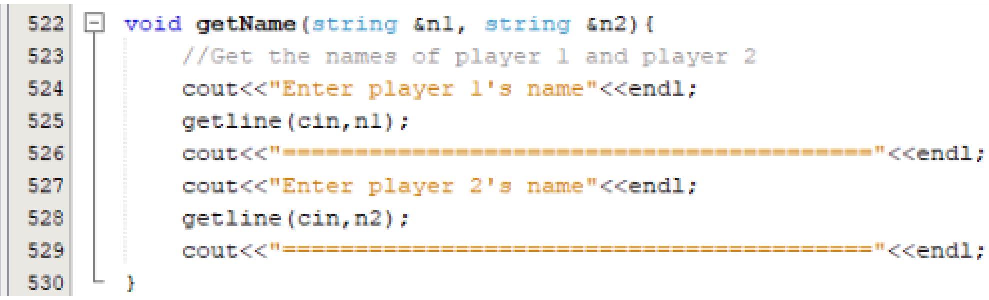
All of the above-mentioned functions-code will be displayed below



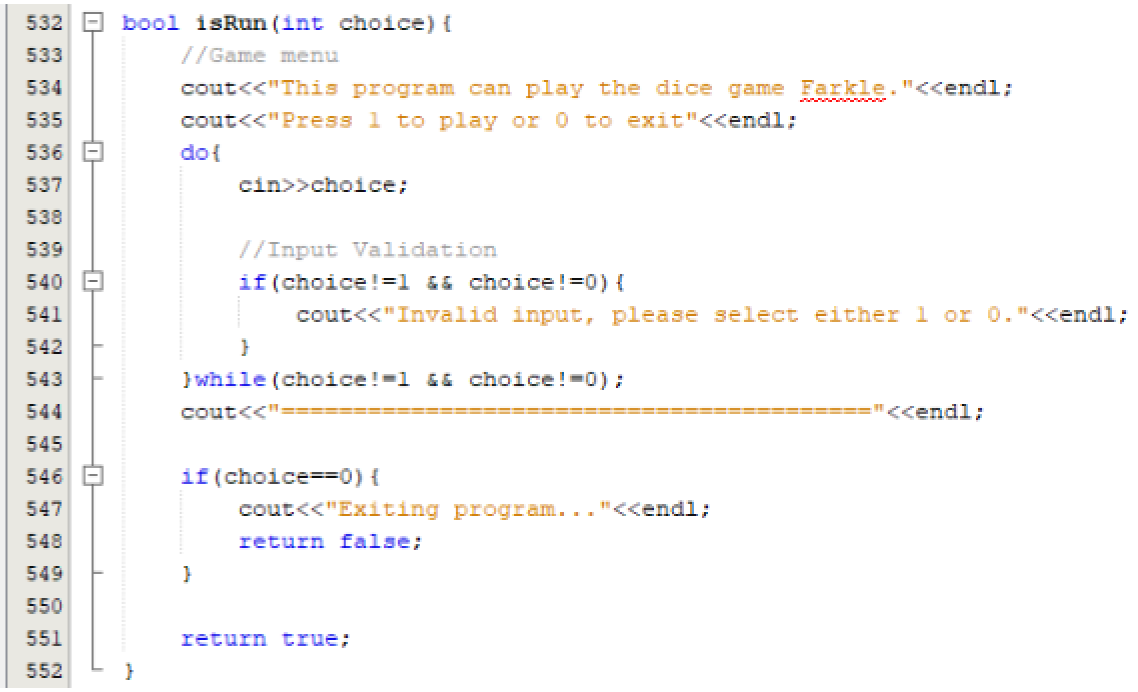
```
void rules(){
    //Display the game rules and scoring guide
    string line;
    ifstream inputFile;
    inputFile.open("GameRules.txt");
    if(inputFile.is_open()){
        while(getline(inputFile,line)){
            cout<<line<<"\n";
        }
    inputFile.close();
    }
}
```



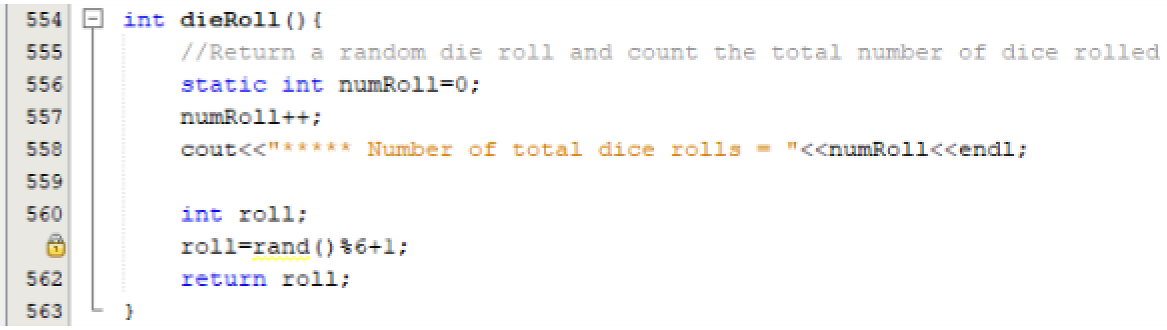
```
void scores(string p1, string p2, float s1, float s2, float w1,
        float w2, float los1, float los2){
    //Save scores and win/losses count to text file
    ofstream outputFile;
    outputFile.open("Scores.txt");
    outputFile<<p1<<"'s score, wins, and losses: "
            <<s1<<" / "<<w1<<" / "<<los1<<"\r\n";
    outputFile<<p2<<"'s score, wins, and losses: "
            <<s2<<" / "<<w2<<" / "<<los2;
    outputFile.close();
}
```



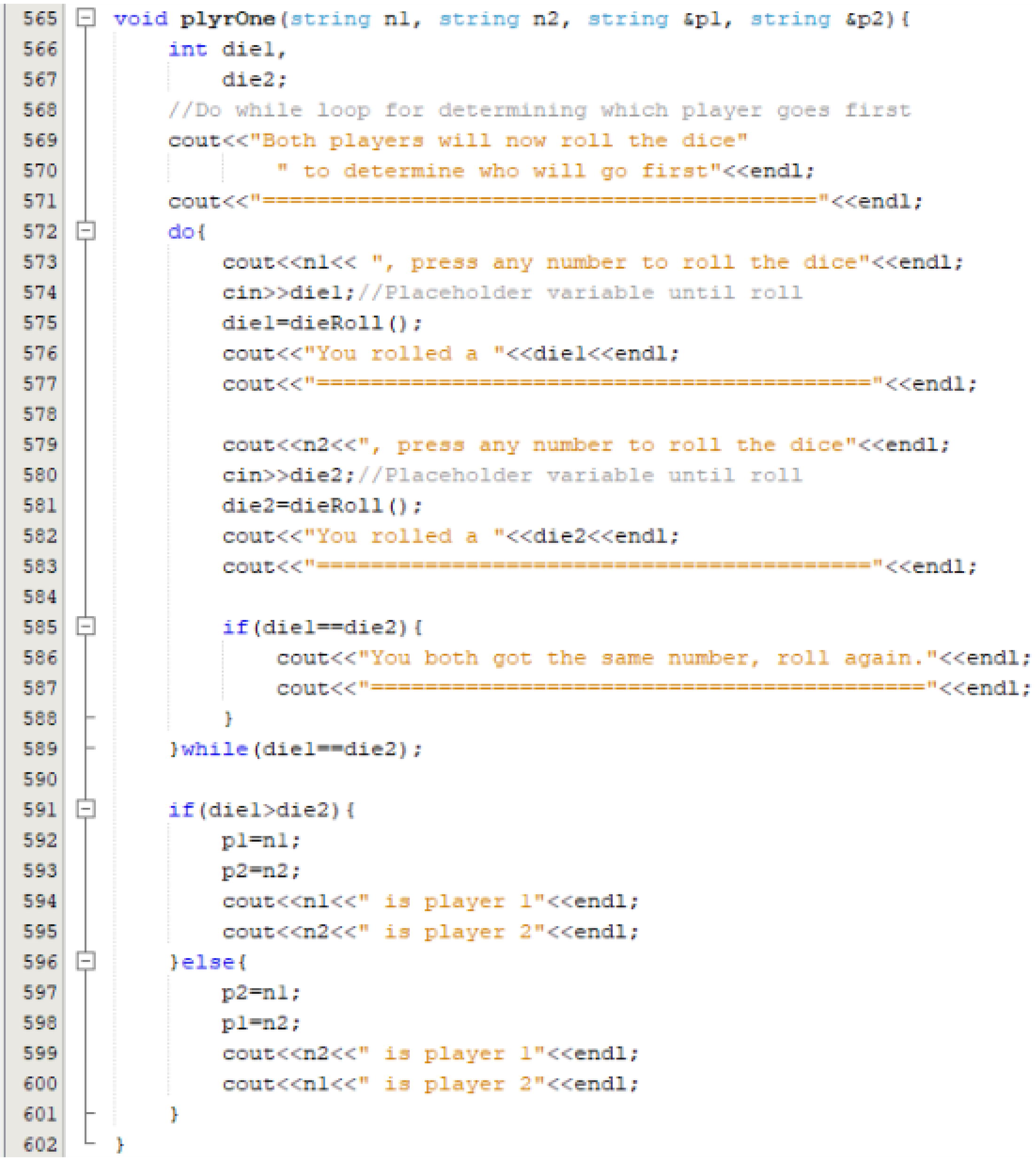
```
void getName(string &n1, string &n2){
    //Get the names of player 1 and player 2
    cout<<"Enter player 1's name"<<endl;
    getline(cin,n1);
    cout<<"========================================="<<endl;
    cout<<"Enter player 2's name"<<endl;
    getline(cin,n2);
    cout<<"========================================="<<endl;
}
```



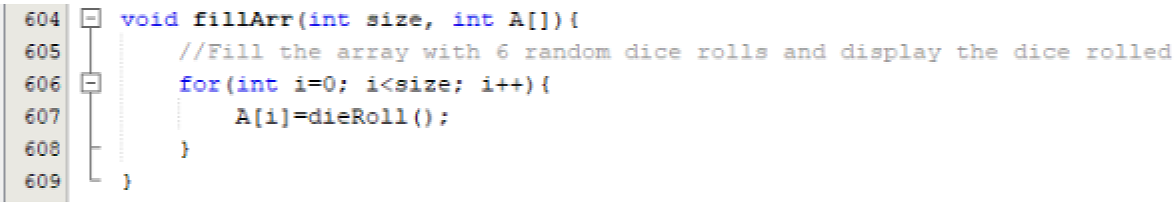
```cpp
bool isRun(int choice){
    //Game menu
    cout<<"This program can play the dice game Farkle."<<endl;
    cout<<"Press 1 to play or 0 to exit"<<endl;
    do{
        cin>>choice;

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

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

    return true;
}
```



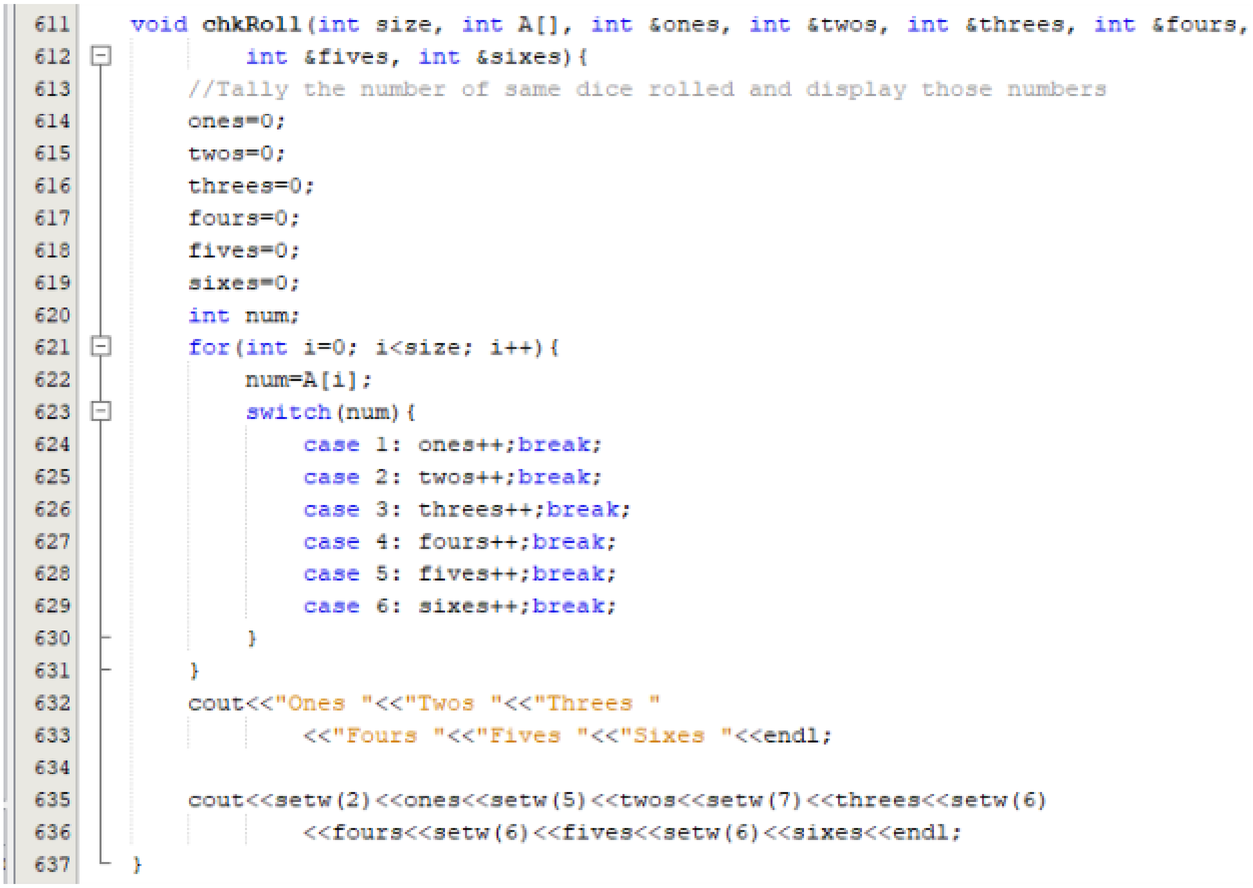
```cpp
int dieRoll(){
    //Return a random die roll and count the total number of dice rolled
    static int numRoll=0;
    numRoll++;
    cout<<"***** Number of total dice rolls = "<<numRoll<<endl;

    int roll;
    roll=rand()%6+1;
    return roll;
}
```



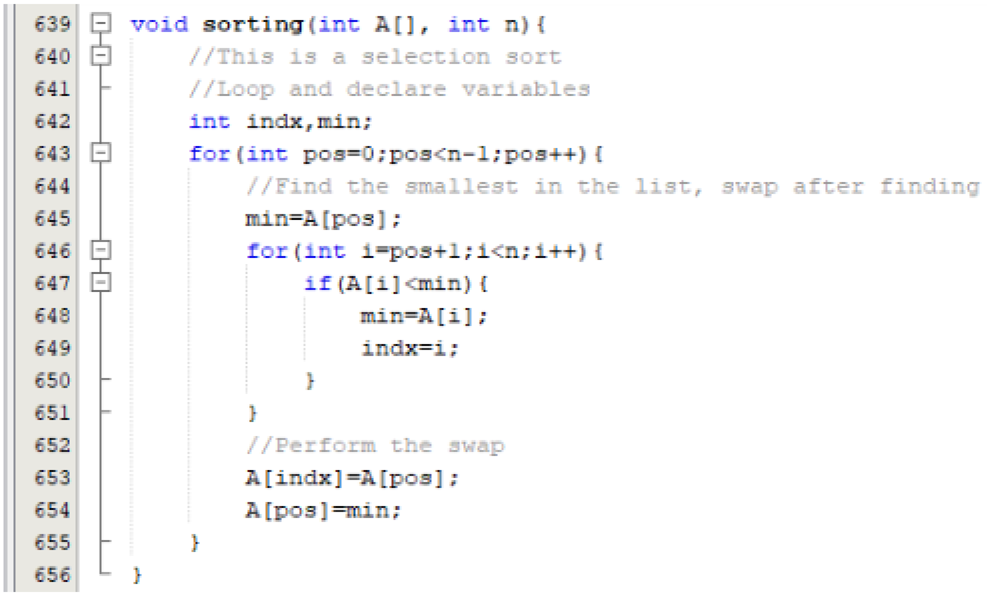
```cpp
void plyrOne(string n1, string n2, string &p1, string &p2){
    int die1,
        die2;
    //Do while loop for determining which player goes first
    cout<<"Both players will now roll the dice"
            " to determine who will go first"<<endl;
    cout<<"=========================================="<<endl;
    do{
        cout<<n1<< ", press any number to roll the dice"<<endl;
        cin>>die1;//Placeholder variable until roll
        die1=dieRoll();
        cout<<"You rolled a "<<die1<<endl;
        cout<<"=========================================="<<endl;

        cout<<n2<<", press any number to roll the dice"<<endl;
        cin>>die2;//Placeholder variable until roll
        die2=dieRoll();
        cout<<"You rolled a "<<die2<<endl;
        cout<<"=========================================="<<endl;

        if(die1==die2){
            cout<<"You both got the same number, roll again."<<endl;
            cout<<"=========================================="<<endl;
        }
    }while(die1==die2);

    if(die1>die2){
        p1=n1;
        p2=n2;
        cout<<n1<<" is player 1"<<endl;
        cout<<n2<<" is player 2"<<endl;
    }else{
        p2=n1;
        p1=n2;
        cout<<n2<<" is player 1"<<endl;
        cout<<n1<<" is player 2"<<endl;
    }
}
```



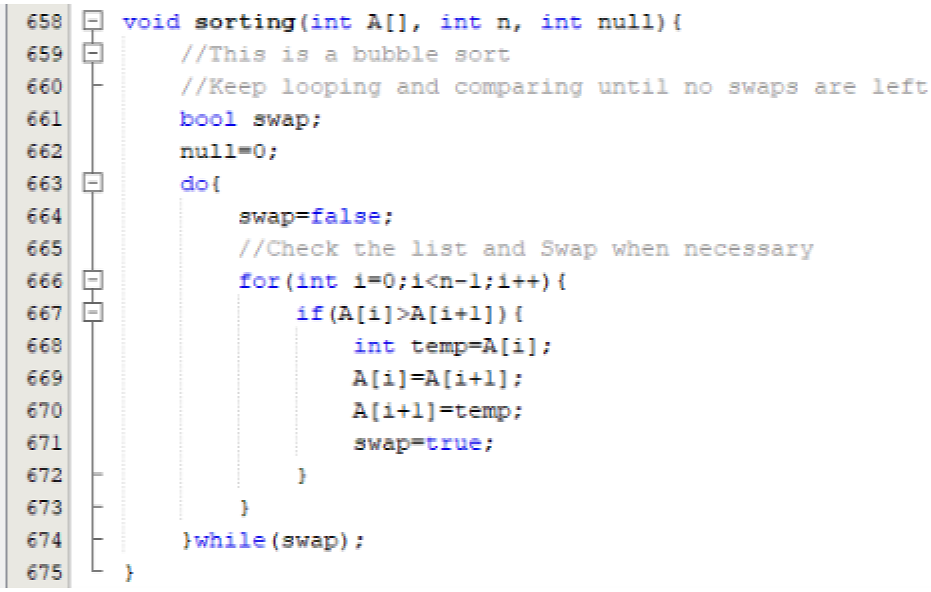
```cpp
void fillArr(int size, int A[]){
    //Fill the array with 6 random dice rolls and display the dice rolled
    for(int i=0; i<size; i++){
        A[i]=dieRoll();
    }
}
```



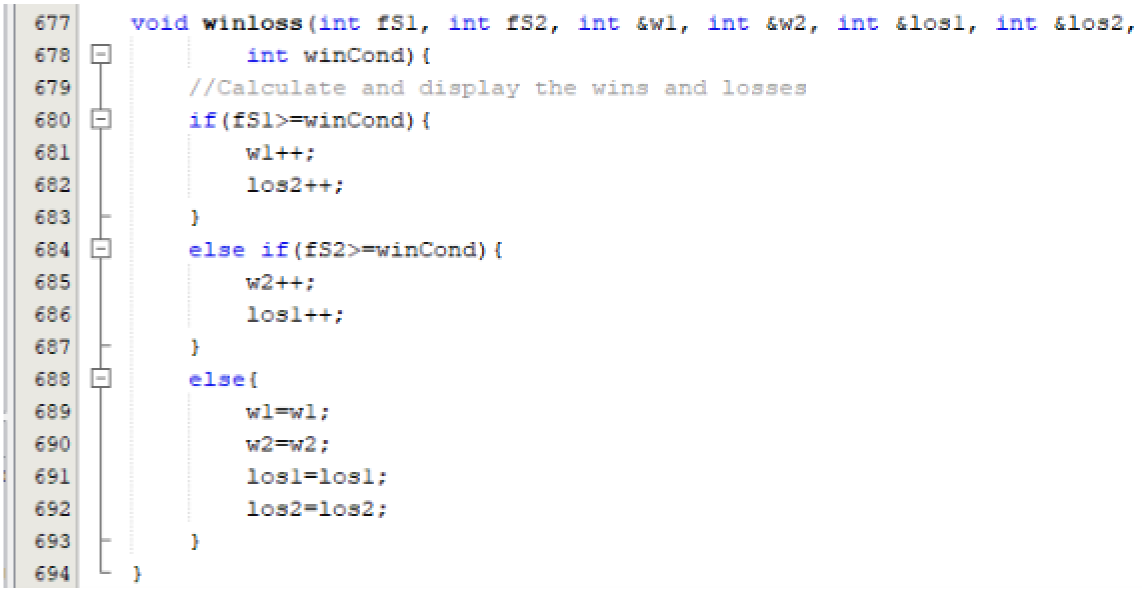
```
void chkRoll(int size, int A[], int &ones, int &twos, int &threes, int &fours,
        int &fives, int &sixes){
    //Tally the number of same dice rolled and display those numbers
    ones=0;
    twos=0;
    threes=0;
    fours=0;
    fives=0;
    sixes=0;
    int num;
    for(int i=0; i<size; i++){
        num=A[i];
        switch(num){
            case 1: ones++;break;
            case 2: twos++;break;
            case 3: threes++;break;
            case 4: fours++;break;
            case 5: fives++;break;
            case 6: sixes++;break;
        }
    }
    cout<<"Ones "<<"Twos "<<"Threes "
            <<"Fours "<<"Fives "<<"Sixes "<<endl;

    cout<<setw(2)<<ones<<setw(5)<<twos<<setw(7)<<threes<<setw(6)
            <<fours<<setw(6)<<fives<<setw(6)<<sixes<<endl;
}
```



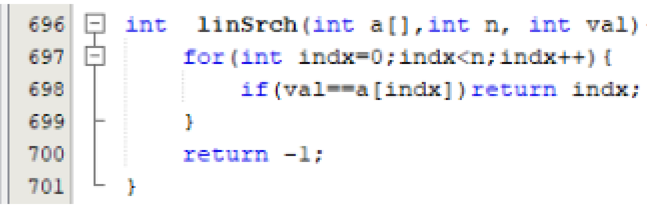
```
void sorting(int A[], int n){
    //This is a selection sort
    //Loop and declare variables
    int indx,min;
    for(int pos=0;pos<n-1;pos++){
        //Find the smallest in the list, swap after finding
        min=A[pos];
        for(int i=pos+1;i<n;i++){
            if(A[i]<min){
                min=A[i];
                indx=i;
            }
        }
        //Perform the swap
        A[indx]=A[pos];
        A[pos]=min;
    }
}
```



```
void sorting(int A[], int n, int null){
    //This is a bubble sort
    //Keep looping and comparing until no swaps are left
    bool swap;
    null=0;
    do{
        swap=false;
        //Check the list and Swap when necessary
        for(int i=0;i<n-1;i++){
            if(A[i]>A[i+1]){
                int temp=A[i];
                A[i]=A[i+1];
                A[i+1]=temp;
                swap=true;
            }
        }
    }while(swap);
}
```



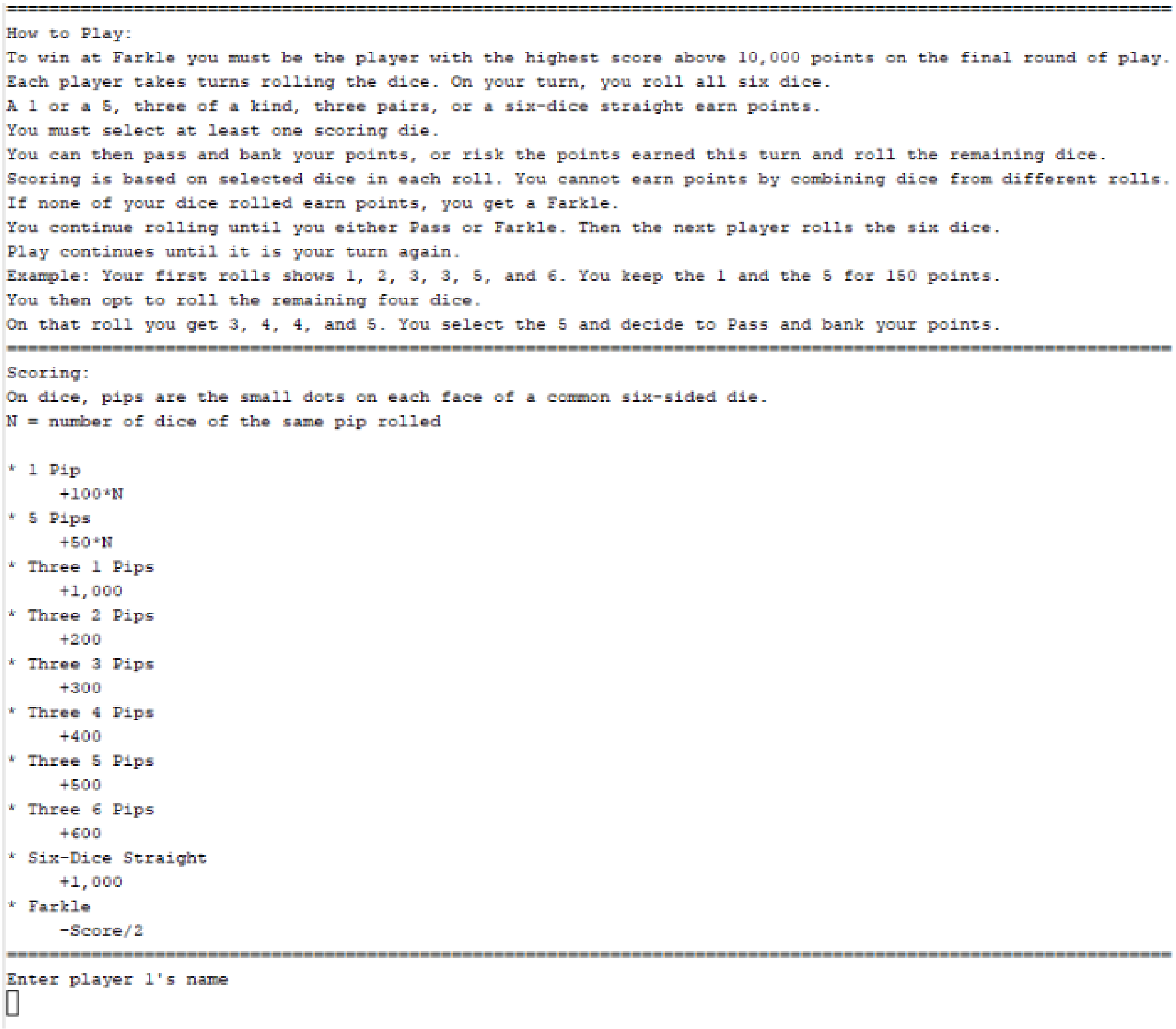
```
void winloss(int fS1, int fS2, int &w1, int &w2, int &los1, int &los2,
        int winCond){
    //Calculate and display the wins and losses
    if(fS1>=winCond){
        w1++;
        los2++;
    }
    else if(fS2>=winCond){
        w2++;
        los1++;
    }
    else{
        w1=w1;
        w2=w2;
        los1=los1;
        los2=los2;
    }
}
```



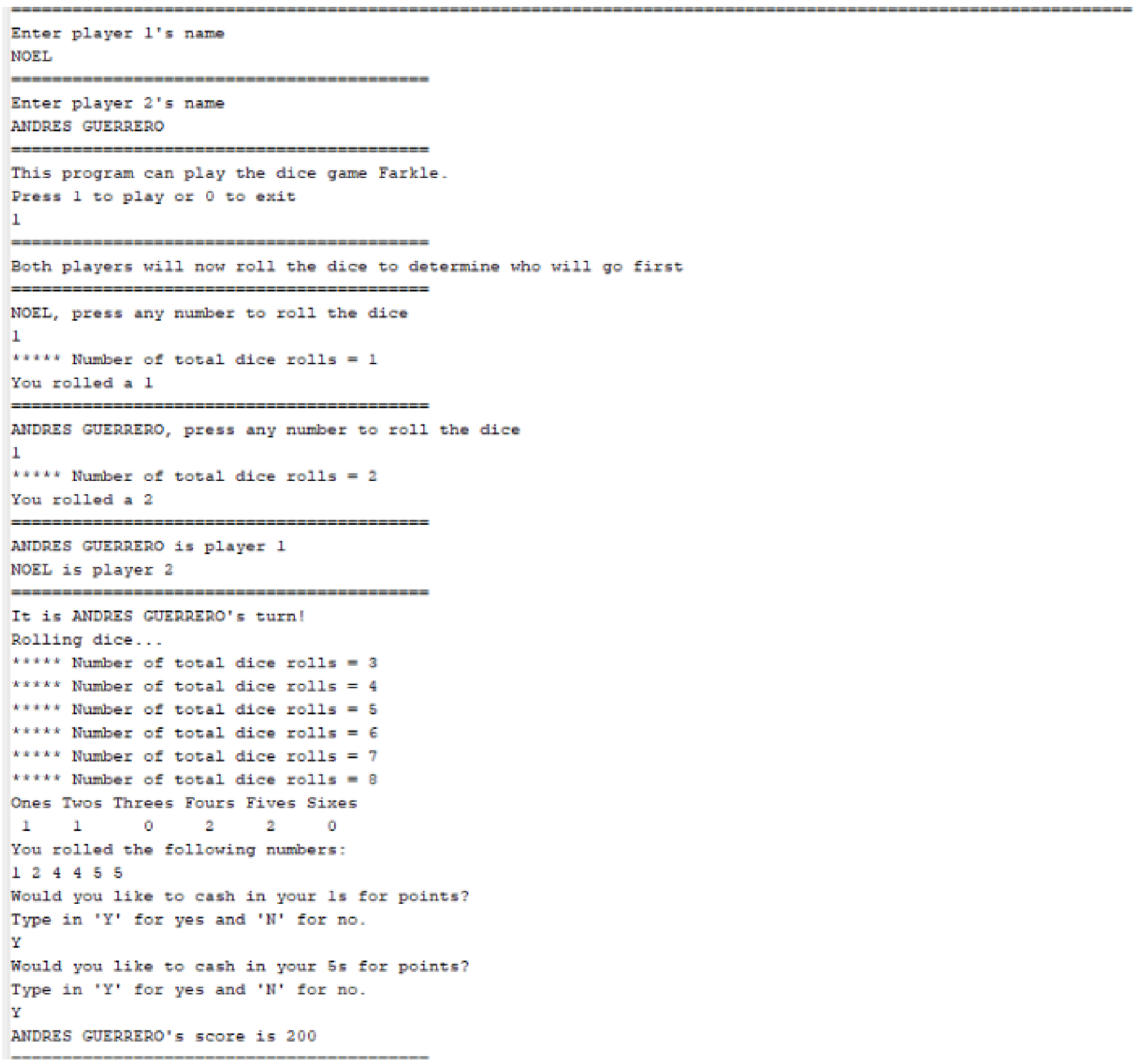
```
int  linSrch(int a[],int n, int val){
    for(int indx=0;indx<n;indx++){
        if(val==a[indx])return indx;
    }
    return -1;
}
```

Proof of a Working Product

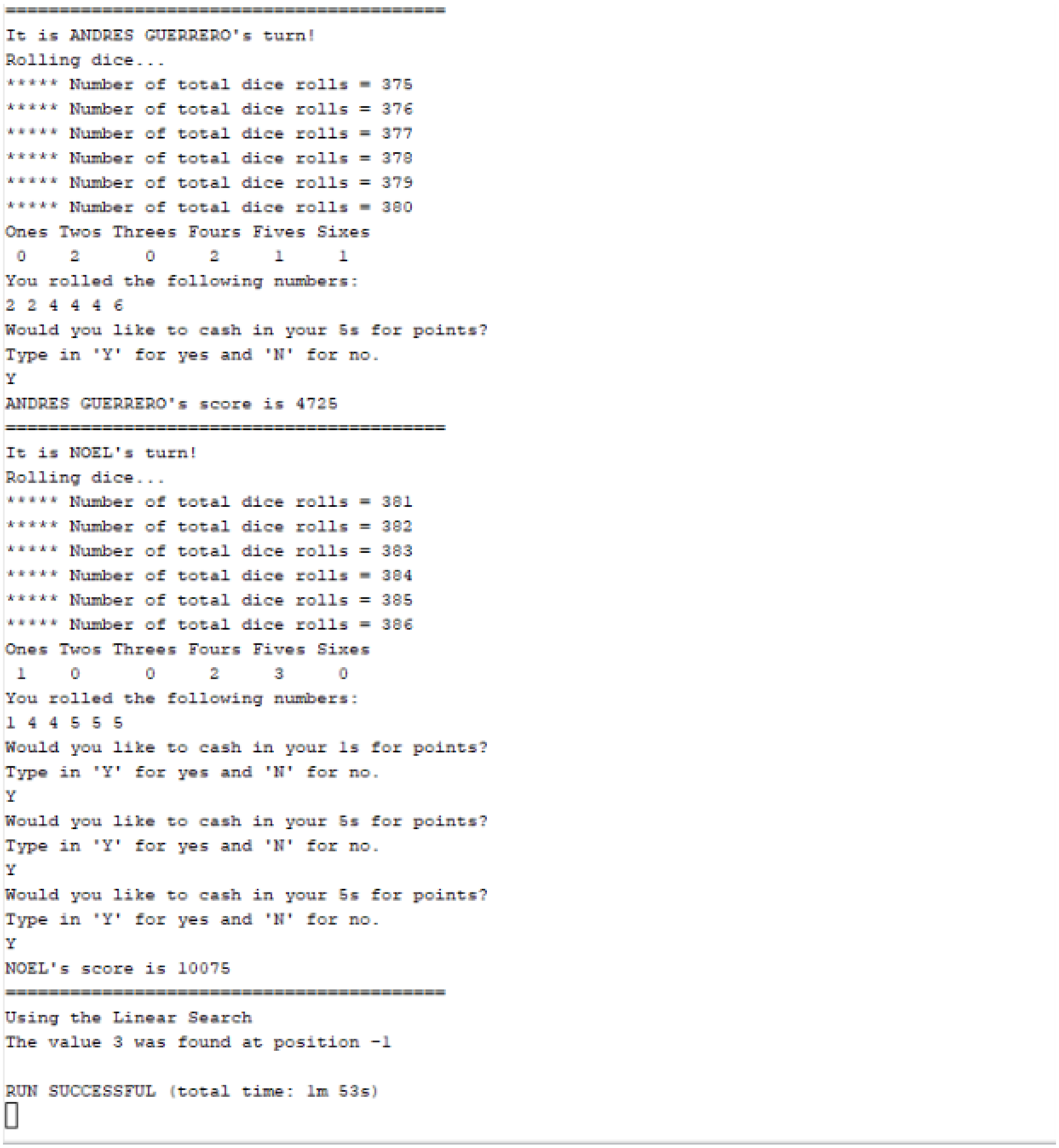
Below are examples of program output to illustrate the full-functionality of Farkle



```
========================================================================================================
How to Play:
To win at Farkle you must be the player with the highest score above 10,000 points on the final round of play.
Each player takes turns rolling the dice. On your turn, you roll all six dice.
A 1 or a 5, three of a kind, three pairs, or a six-dice straight earn points.
You must select at least one scoring die.
You can then pass and bank your points, or risk the points earned this turn and roll the remaining dice.
Scoring is based on selected dice in each roll. You cannot earn points by combining dice from different rolls.
If none of your dice rolled earn points, you get a Farkle.
You continue rolling until you either Pass or Farkle. Then the next player rolls the six dice.
Play continues until it is your turn again.
Example: Your first rolls shows 1, 2, 3, 3, 5, and 6. You keep the 1 and the 5 for 150 points.
You then opt to roll the remaining four dice.
On that roll you get 3, 4, 4, and 5. You select the 5 and decide to Pass and bank your points.
========================================================================================================
Scoring:
On dice, pips are the small dots on each face of a common six-sided die.
N = number of dice of the same pip rolled

* 1 Pip
     +100*N
* 5 Pips
     +50*N
* Three 1 Pips
     +1,000
* Three 2 Pips
     +200
* Three 3 Pips
     +300
* Three 4 Pips
     +400
* Three 5 Pips
     +500
* Three 6 Pips
     +600
* Six-Dice Straight
     +1,000
* Farkle
     -Score/2
========================================================================================================
Enter player 1's name
```



```
==============================================================================================================
Enter player 1's name
NOEL
=========================================
Enter player 2's name
ANDRES GUERRERO
=========================================
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
***** Number of total dice rolls = 1
You rolled a 1
=========================================
ANDRES GUERRERO, press any number to roll the dice
1
***** Number of total dice rolls = 2
You rolled a 2
=========================================
ANDRES GUERRERO is player 1
NOEL is player 2
=========================================
It is ANDRES GUERRERO's turn!
Rolling dice...
***** Number of total dice rolls = 3
***** Number of total dice rolls = 4
***** Number of total dice rolls = 5
***** Number of total dice rolls = 6
***** Number of total dice rolls = 7
***** Number of total dice rolls = 8
Ones Twos Threes Fours Fives Sixes
 1    1      0     2     2     0
You rolled the following numbers:
1 2 4 4 5 5
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
ANDRES GUERRERO's score is 200
=========================================
```



```
==========================================
It is ANDRES GUERRERO's turn!
Rolling dice...
***** Number of total dice rolls = 375
***** Number of total dice rolls = 376
***** Number of total dice rolls = 377
***** Number of total dice rolls = 378
***** Number of total dice rolls = 379
***** Number of total dice rolls = 380
Ones Twos Threes Fours Fives Sixes
 0    2      0     2     1     1
You rolled the following numbers:
2 2 4 4 4 6
Would you like to cash in your 5s for points?
Type in 'Y' for yes and 'N' for no.
Y
ANDRES GUERRERO's score is 4725
==========================================
It is NOEL's turn!
Rolling dice...
***** Number of total dice rolls = 381
***** Number of total dice rolls = 382
***** Number of total dice rolls = 383
***** Number of total dice rolls = 384
***** Number of total dice rolls = 385
***** Number of total dice rolls = 386
Ones Twos Threes Fours Fives Sixes
 1    0      0     2     3     0
You rolled the following numbers:
1 4 4 5 5 5
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
Would you like to cash in your 5s for points?
Type in 'Y' for yes and 'N' for no.
Y
NOEL's score is 10075
==========================================
Using the Linear Search
The value 3 was found at position -1

RUN SUCCESSFUL (total time: 1m 53s)
```