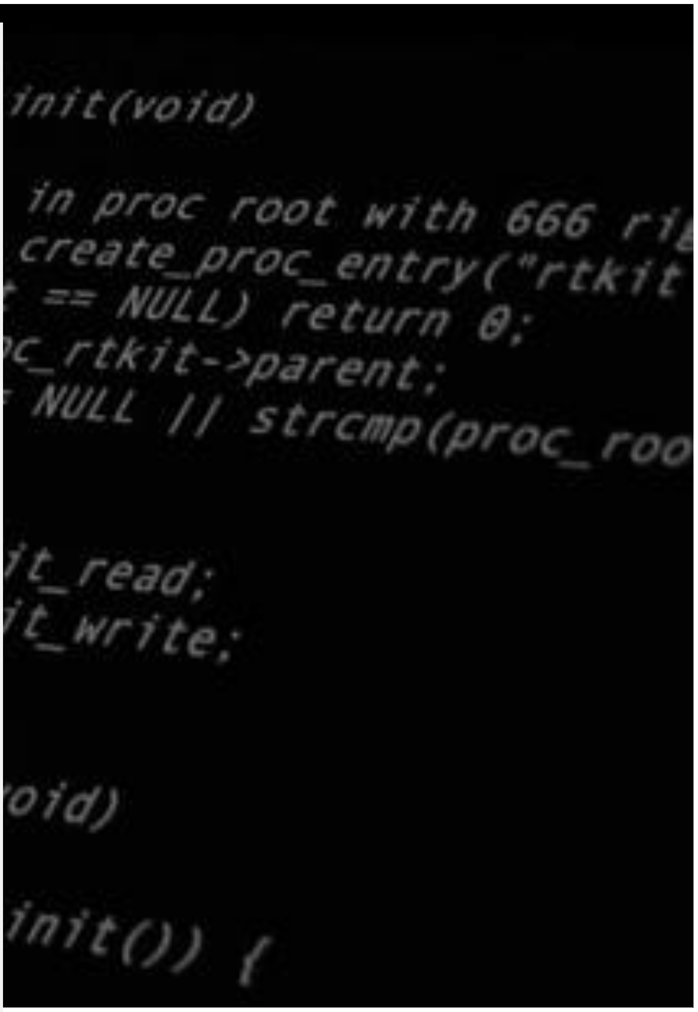
Rock, Paper, Scissor



```
init(void)

 in proc root with 666 ri
 create_proc_entry("rtkit
t == NULL) return 0;
c_rtkit->parent;
 NULL || strcmp(proc_roo

it_read;
it_write;

oid)

init()) {
```

GROUP NO 3

Mini Project
INTE 11223 PROGRAMMING CONCEPTS

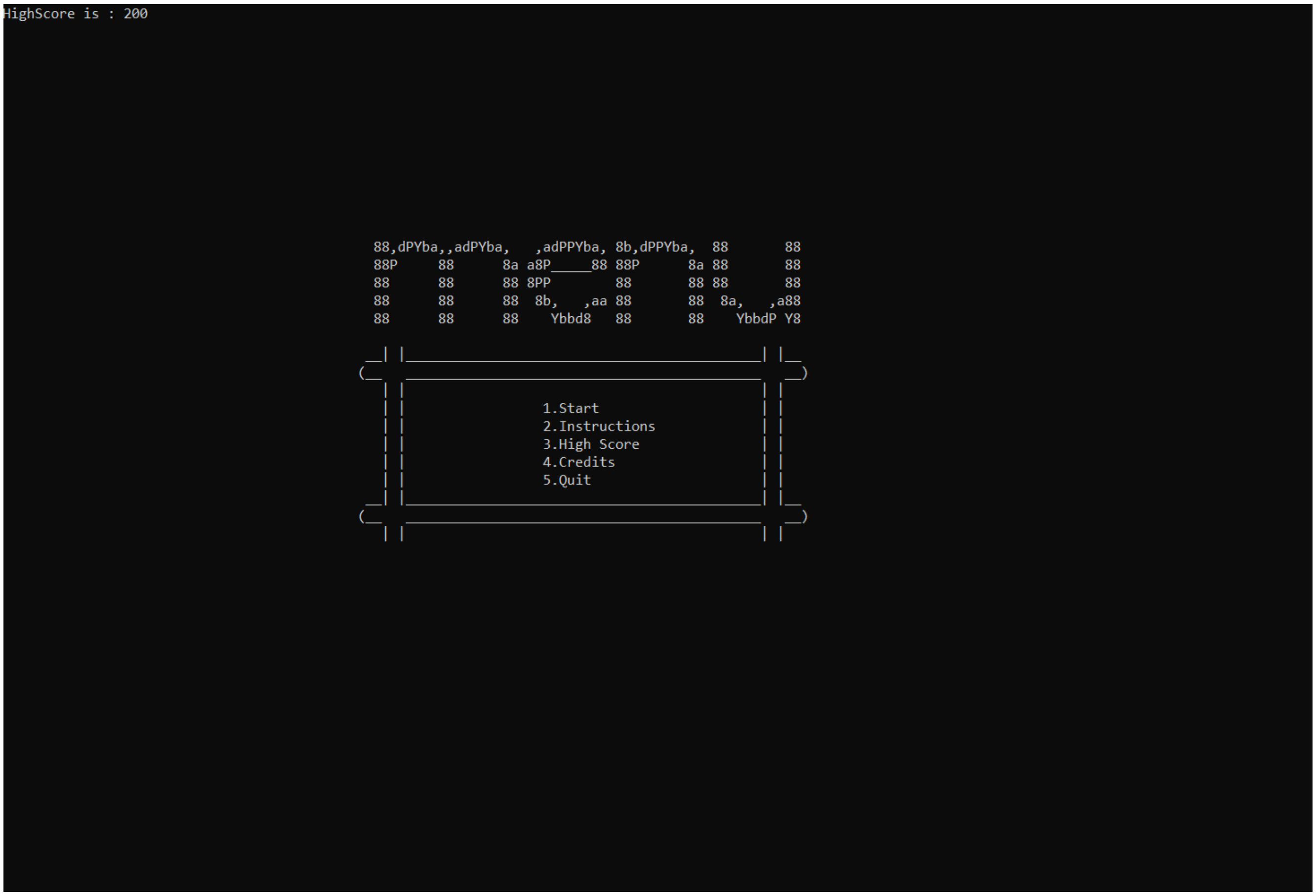
Team Console Tigers

Contents

Basic Documentation

First, you have to open the “firstSemFinal.exe” that comes with this documentation.

It will pop up a window like below after a welcome message,



```
HighScore is : 200



  88,dPYba,,adPYba,   ,adPPYba, 8b,dPPYba,  88       88
  88P     88      8a a8P_____88 88P      8a 88       88
  88      88      88 8PP        88       88 88       88
  88      88      88  8b,   ,aa 88       88  8a,   ,a88
  88      88      88    Ybbd8   88       88    YbbdP Y8

 __| |____________________________________________| |__
(__   ____________________________________________   __)
   | |                                            | |
   | |                 1.Start                    | |
   | |                 2.Instructions             | |
   | |                 3.High Score               | |
   | |                 4.Credits                  | |
   | |                 5.Quit                     | |
 __| |____________________________________________| |__
(__   ____________________________________________   __)
   | |                                            | |
```

You can press one to start the game.

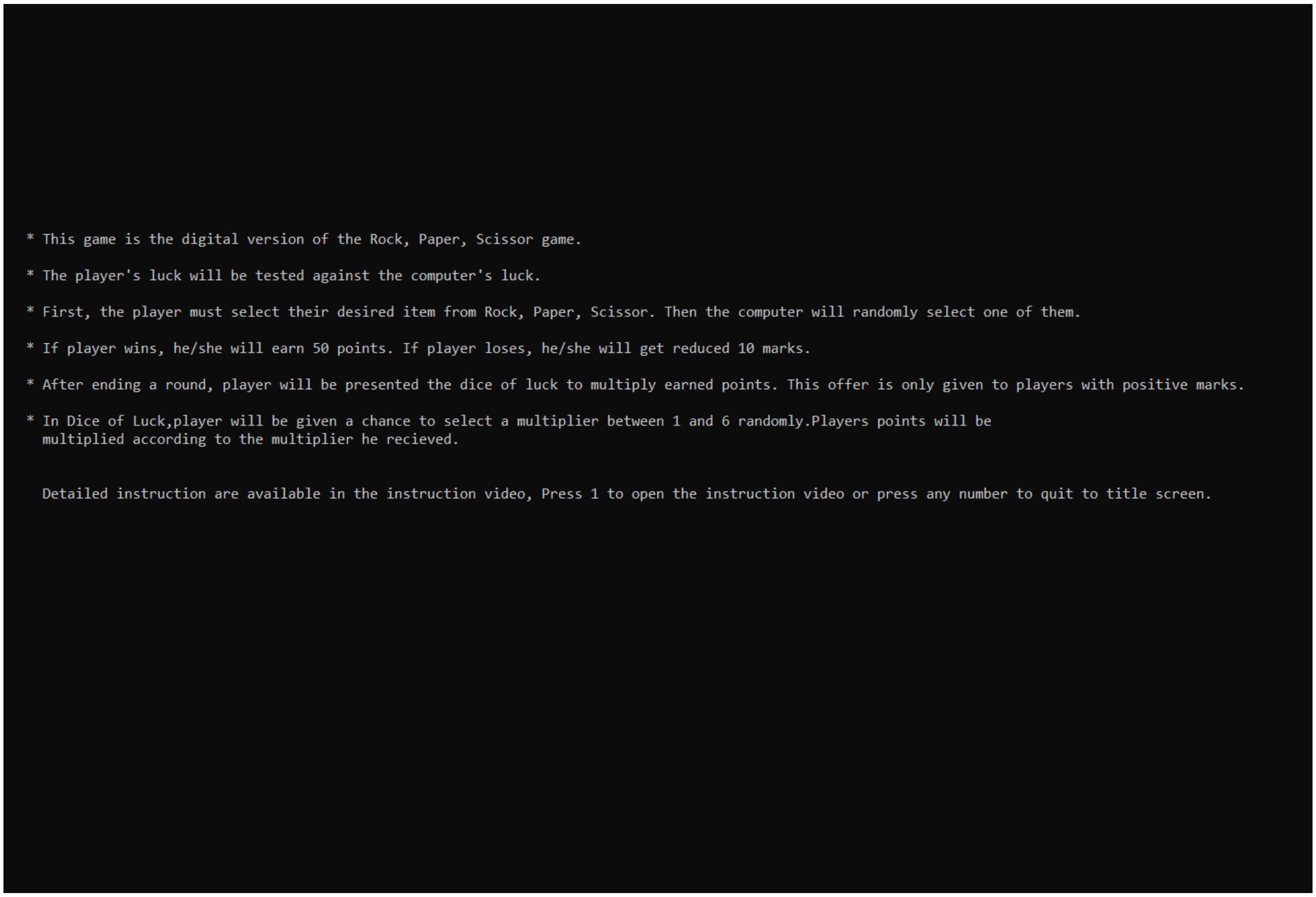
If you want to see the detailed instructions for the game, press 2. After pressing 2, a written, detailed instruction window will be opened. If you want further instructions for this game, you can again press 1. It will pop up a YouTube video in your default browser.

If you want to know your high score, press 3. By pressing any key in the high score menu will return you to the main menu.

If you want to know the names of the creators of this game press 4.

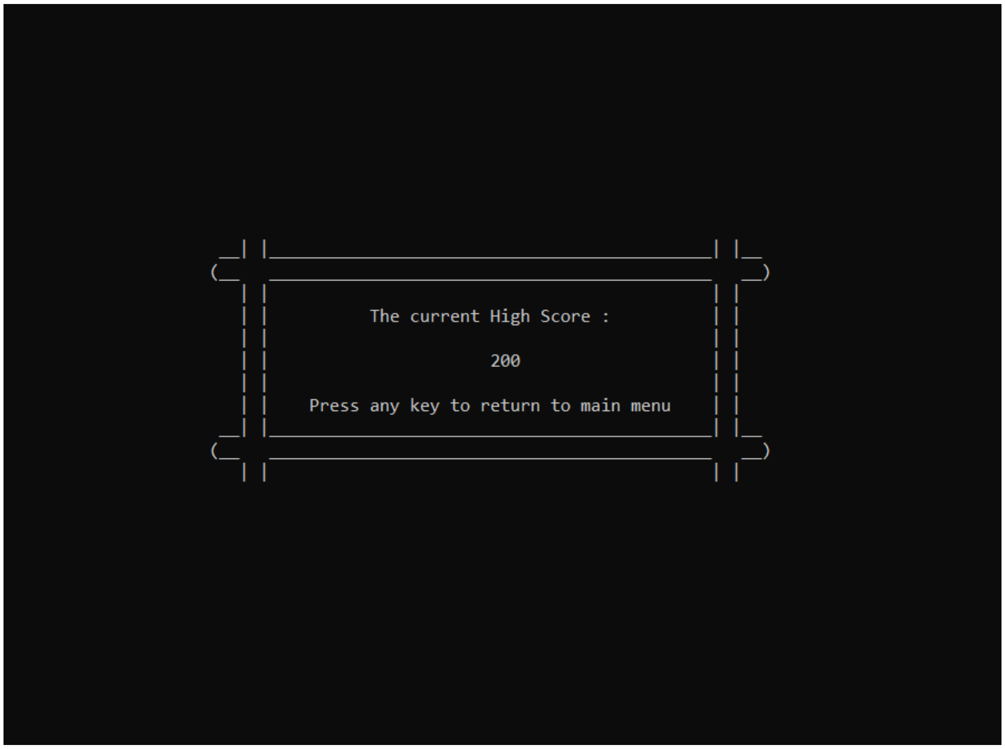
You can quit the game using 5.

Instruction window



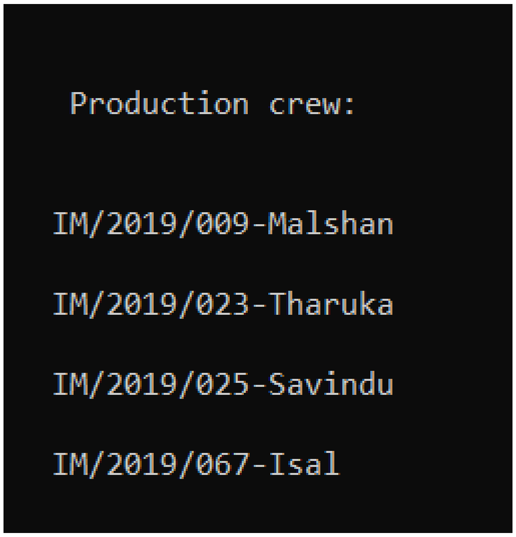
```
* This game is the digital version of the Rock, Paper, Scissor game.

* The player's luck will be tested against the computer's luck.

* First, the player must select their desired item from Rock, Paper, Scissor. Then the computer will randomly select one of them.

* If player wins, he/she will earn 50 points. If player loses, he/she will get reduced 10 marks.

* After ending a round, player will be presented the dice of luck to multiply earned points. This offer is only given to players with positive marks.

* In Dice of Luck,player will be given a chance to select a multiplier between 1 and 6 randomly.Players points will be
  multiplied according to the multiplier he recieved.


  Detailed instruction are available in the instruction video, Press 1 to open the instruction video or press any number to quit to title screen.
```

High Score window



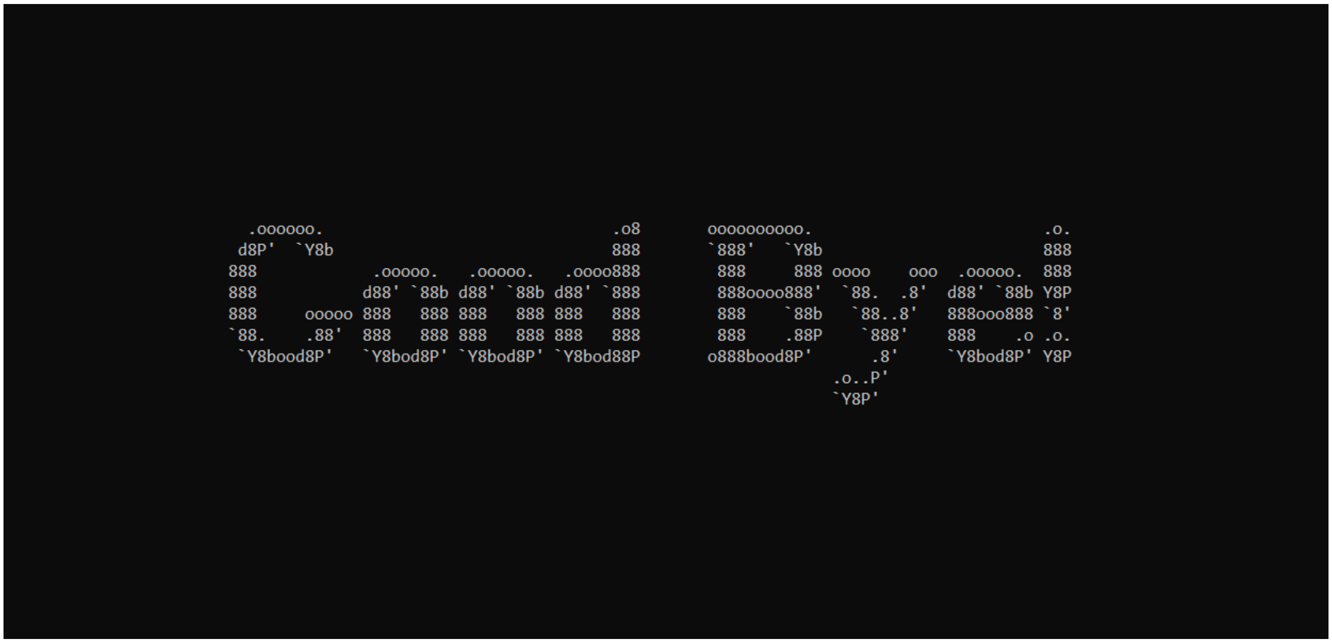
```
 __| |____________________________________________| |__
(__   ____________________________________________   __)
   | |                                            | |
   | |          The current High Score :          | |
   | |                                            | |
   | |                      200                   | |
   | |                                            | |
   | |    Press any key to return to main menu    | |
 __| |____________________________________________| |__
(__   ____________________________________________   __)
   | |                                            | |
```

Animated Credits window



```
 Production crew:


IM/2019/009-Malshan

IM/2019/023-Tharuka

IM/2019/025-Savindu

IM/2019/067-Isal
```

Quit Game window



```
  .oooooo.                              .o8       oooooooooo.                        .o.
 d8P'  `Y8b                             888       `888'   `Y8b                       888
888            .ooooo.   .ooooo.   .oooo888        888     888 oooo    ooo  .ooooo.  888
888           d88' `88b d88' `88b d88' `888        888oooo888'  `88.  .8'  d88' `88b Y8P
888     ooooo 888   888 888   888 888   888        888    `88b   `88..8'   888ooo888 `8'
`88.    .88'  888   888 888   888 888   888        888    .88P    `888'    888    .o .o.
 `Y8bood8P'   `Y8bod8P' `Y8bod8P' `Y8bod88P       o888bood8P'      .8'     `Y8bod8P' Y8P
                                                               .o..P'
                                                               `Y8P'
```

Instructions

- This game is the digital version of the Rock, Paper, Scissor game.
- The player's luck will be tested against the computer's luck.
- First, the player must select their desired item from Rock, Paper, Scissor. Then the computer will randomly select one of them.

- If the player wins, he/she will earn 50 points. If the player loses, he/she will get reduced 10 marks
- After ending a round, the player will be presented the dice of luck to multiply earned points. This offer is only given to players with positive marks.
- In Dice of Luck, the player will be given a chance to select a multiplier between 1 and 6 randomly. The Player's points will be multiplied according to the multiplier he received.

Problems/ Challenges we had with our project.

We created the basic gameplay and the game logic at the initial stages of creating this game and then we started adding graphics to our console game. Our main target was to increase the user experience, although this is a simple console game. We found several websites that generate those ASCII arts, but their outputs weren’t very accurate. Therefore, we had several challenges when creating some ASCII art game graphics for this game.

We planned to add multiplayer gameplay (in two PCs) other than our single-player mode to this game in the initial stages. But when we further discussed it again with our team members, we decided to replace that multiplayer section with another section that is achievable within this time frame.

What would we have done differently if we could do it again?

We can further develop this game by adding some features like,

- A multiplayer section for this game.

- By adding some difficulty levels for the user.
- Some background soundtrack.

CODE

We use 23 different functions other than main method to create this game. Their uses are as below,

void welcomeScreen();
This function will print the main menu and get the required inputs to redirect to the different menus.

void GameCore();
This function contains the key elements of the game.

void QuitGame();
You can quit the game through this function. “exit(0)" is used inside this.

void Instructions();
This function contains the instructions for this game, and you can open the instruction video through this.

void InvalidInput();
This function will be called whenever a user presses an invalid input.

void HighScoreViewer();
This function will be called when you press the highscore in the main menu.

void CompareSelection(char n);
This function is used to compare the user input with a random choice generated by the computer.

bool gameLogic(int pcGenrated, int userSelection);
This function contains the basic game logic of the game.

void printScissor();
This will print the scissor Ascii Art

void print_rock();
This prints the Rock ascii art

void credits();
This function will print "who created this game"

void print_welcome();
This will print the ascii art off the welcome text.

void Select_elements_inline();
This will print all 3 elements in the game in one row.

void print_paper();
This prints the paper's ascii art.

void print_main_menu();
This is used to print the main menu

void loading();
Loading animation.

void youWin();
This prints the “win” ascii art.

void youLoss();
This prints the “Loss” ascii art.

void print_HighScoreMenu();
This will prints the High score menu.

void fileHandler();

This function is used for handling the save status.

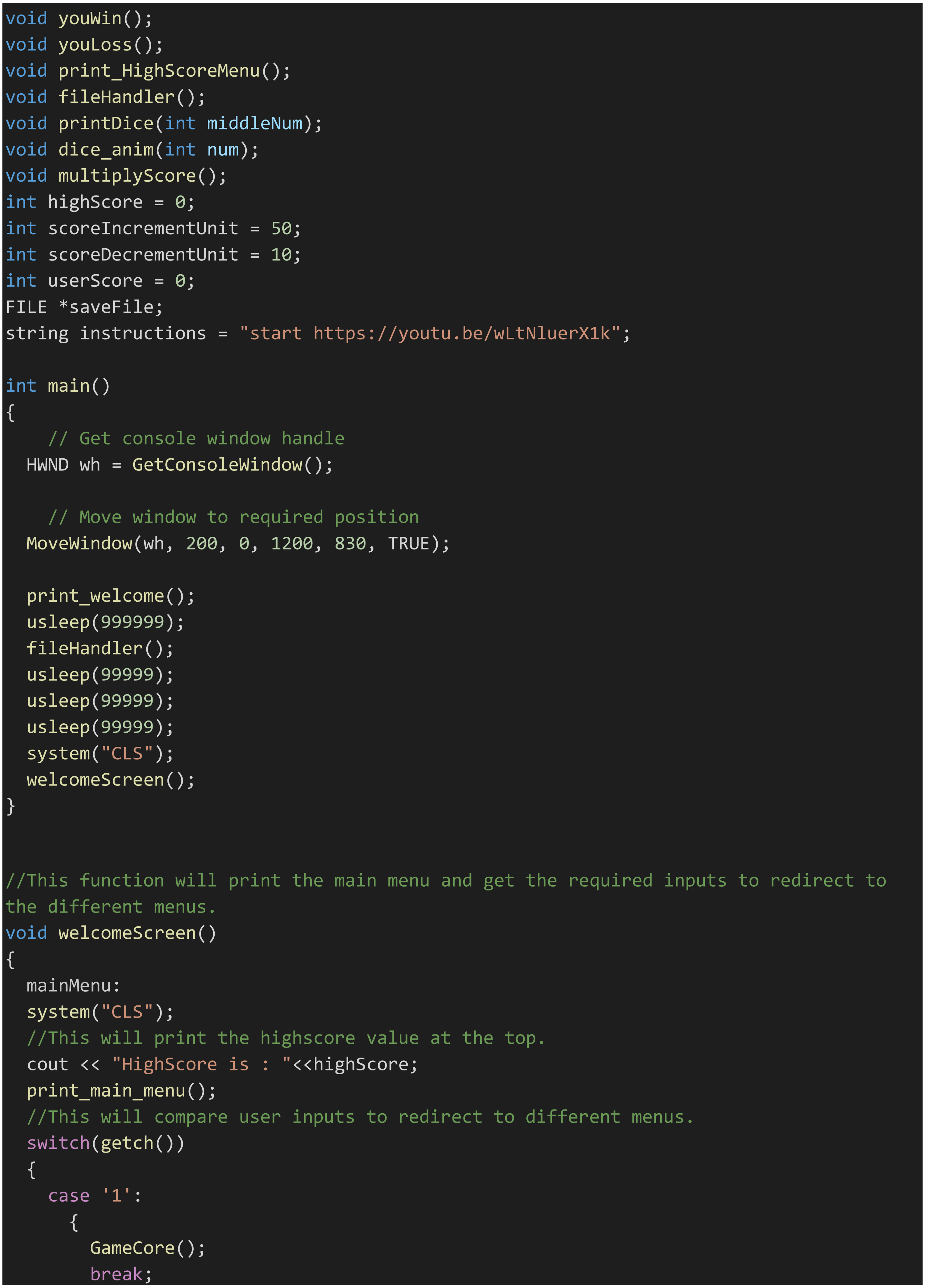
void printDice(int middleNum);
This prints the dice of the luck in this game

void dice_anim(int num);
This contains the animation of the dice.

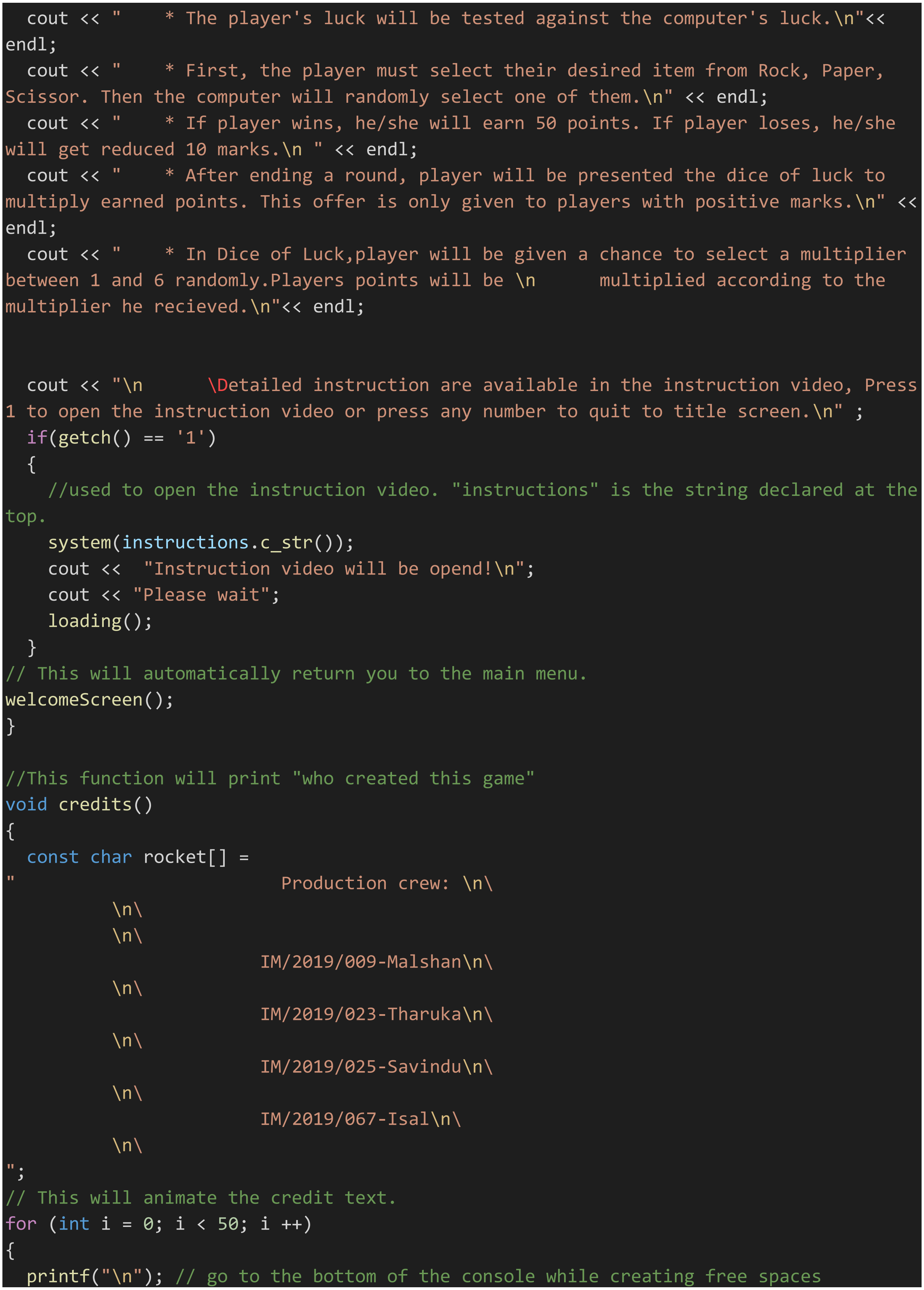
void multiplyScore();
This contains the logic to multiply the user score.

Complete code

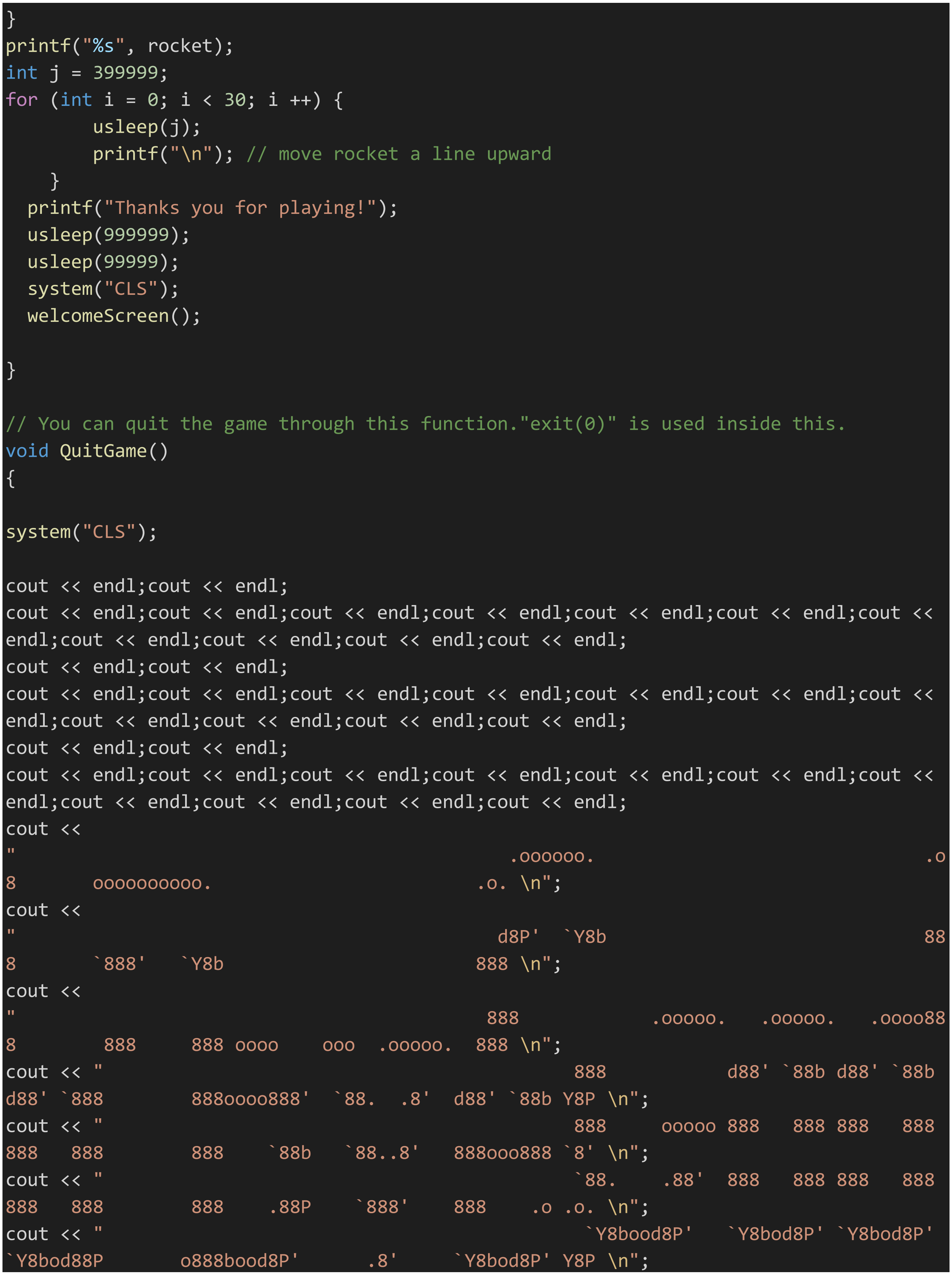
```
#include<iostream>
#include<cstdlib>
#include<stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include<conio.h> // cls
#include <windows.h>
#include<iomanip>
#include<string>
#include <cmath>
using namespace std;

void welcomeScreen();
void GameCore();
void QuitGame();
void Instructions();
void InvalidInput();
void HighScoreViewer();
void CompareSelection(char n);
bool gameLogic(int pcGenrated, int userSelection);
void printScissor();
void print_rock();
void credits();
void print_welcome();
void Select_elements_inline();
void print_paper();
void print_main_menu();
void loading();
```



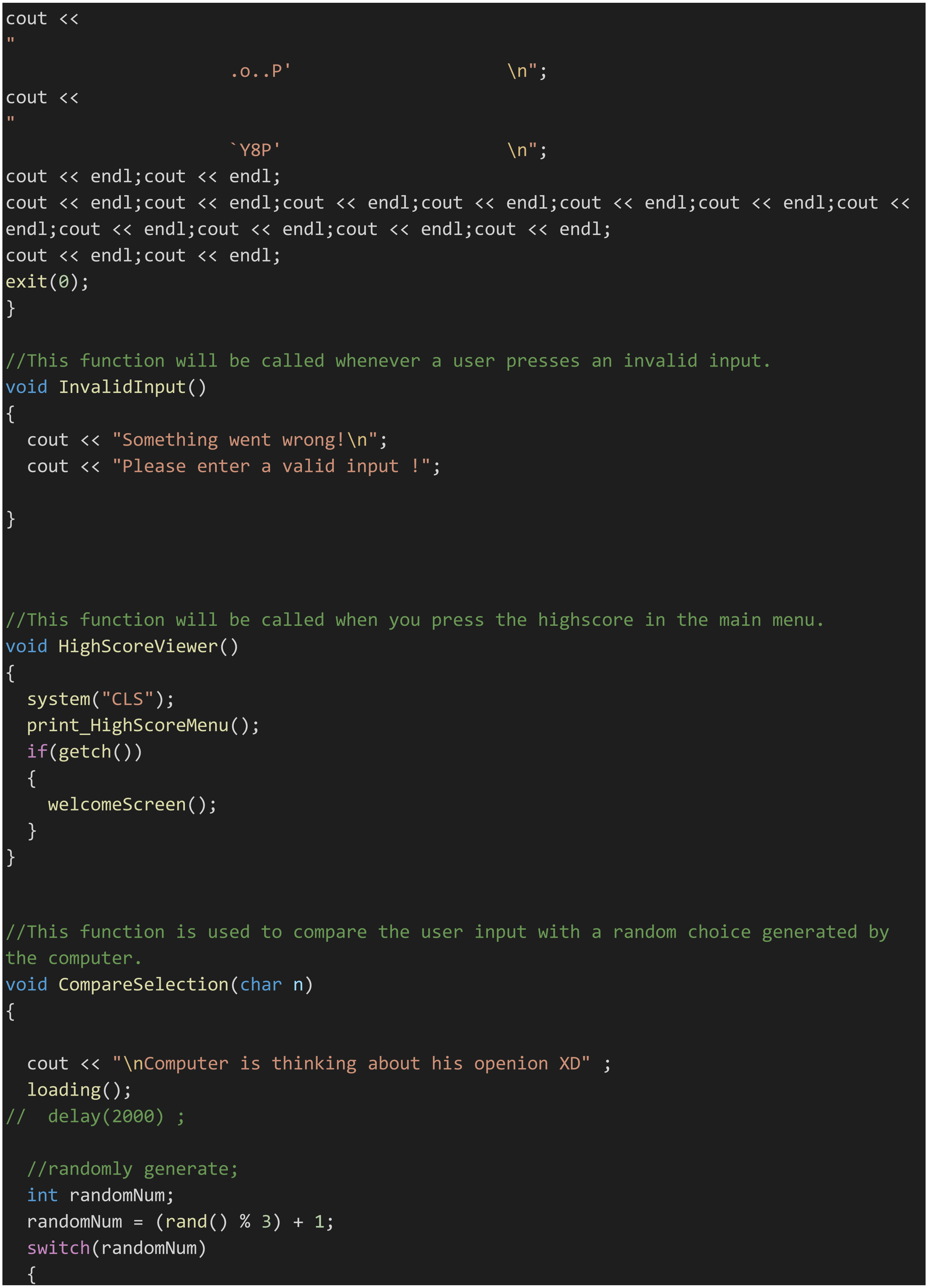
```
void youWin();
void youLoss();
void print_HighScoreMenu();
void fileHandler();
void printDice(int middleNum);
void dice_anim(int num);
void multiplyScore();
int highScore = 0;
int scoreIncrementUnit = 50;
int scoreDecrementUnit = 10;
int userScore = 0;
FILE *saveFile;
string instructions = "start https://youtu.be/wLtNluerX1k";

int main()
{
    // Get console window handle
  HWND wh = GetConsoleWindow();

    // Move window to required position
  MoveWindow(wh, 200, 0, 1200, 830, TRUE);

  print_welcome();
  usleep(999999);
  fileHandler();
  usleep(99999);
  usleep(99999);
  usleep(99999);
  system("CLS");
  welcomeScreen();
}


//This function will print the main menu and get the required inputs to redirect to
the different menus.
void welcomeScreen()
{
  mainMenu:
  system("CLS");
  //This will print the highscore value at the top.
  cout << "HighScore is : "<<highScore;
  print_main_menu();
  //This will compare user inputs to redirect to different menus.
  switch(getch())
  {
    case '1':
      {
        GameCore();
        break;
      }
    case '2':
      {
        Instructions();
        break;
      }
    case '3':
      {
        HighScoreViewer();
        break;
      }
    case '4':
      {
        credits();
        break;
      }
    case '5':
      {
        QuitGame();
        break;
      }
    default :
      {
        goto mainMenu;
        break;
      }
  }
}


//This function contains the key elements of the game.
void GameCore()
{


//This loop will run the game again and again.
while(true)
{


  Select_elements_inline();
  char selection = getch(); // selection input from user
  system("CLS"); // clear screen
  switch(selection)
  {
    case '1' :
      {
        cout << "You have selected Rock.\n" ;
        print_rock();
        break;
      }

    case '2' :
      {
        cout << "You have selected paper.\n" ;
        print_paper();
        break;
      }
    case '3' :
      {
        cout << "You have selected Scissor.\n" ;
        printScissor();
        break;
      }

    case '4' :
      {
        welcomeScreen() ;
        break;

      }
    default :
      {
        cout << "Invalid input.please select again\n" ;
        GameCore();
        break;
      }
  }

    CompareSelection(selection);

    cout <<
"\n..............................................................................\n";
}

}

//This function contains the instructions for this game, and you can open the
instruction video through this.
void Instructions()
{
  system("CLS");
  cout << endl;cout << endl;
  cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout
<< endl;cout << endl;cout << endl;cout << endl;cout << endl;
  cout << "     * This game is the digital version of the Rock, Paper, Scissor
game.\n"<< endl;
```



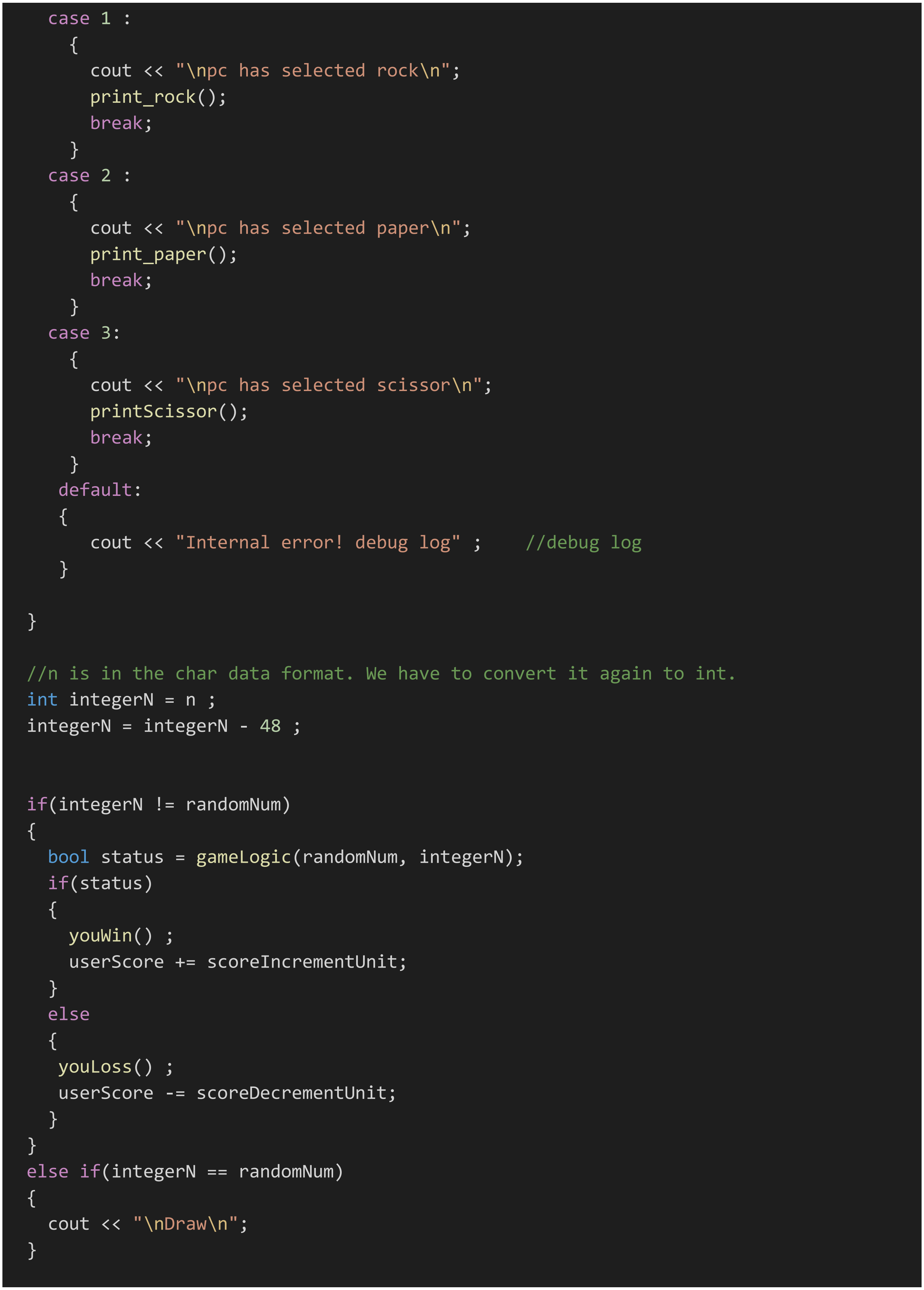
```
  cout << "    * The player's luck will be tested against the computer's luck.\n"<<
endl;
  cout << "    * First, the player must select their desired item from Rock, Paper,
Scissor. Then the computer will randomly select one of them.\n" << endl;
  cout << "    * If player wins, he/she will earn 50 points. If player loses, he/she
will get reduced 10 marks.\n " << endl;
  cout << "    * After ending a round, player will be presented the dice of luck to
multiply earned points. This offer is only given to players with positive marks.\n" <<
endl;
  cout << "    * In Dice of Luck,player will be given a chance to select a multiplier
between 1 and 6 randomly.Players points will be \n      multiplied according to the
multiplier he recieved.\n"<< endl;


  cout << "\n      \Detailed instruction are available in the instruction video, Press
1 to open the instruction video or press any number to quit to title screen.\n" ;
  if(getch() == '1')
  {
    //used to open the instruction video. "instructions" is the string declared at the
top.
    system(instructions.c_str());
    cout <<  "Instruction video will be opend!\n";
    cout << "Please wait";
    loading();
  }
// This will automatically return you to the main menu.
welcomeScreen();
}

//This function will print "who created this game"
void credits()
{
  const char rocket[] =
"                         Production crew: \n\
          \n\
          \n\
                        IM/2019/009-Malshan\n\
          \n\
                        IM/2019/023-Tharuka\n\
          \n\
                        IM/2019/025-Savindu\n\
          \n\
                        IM/2019/067-Isal\n\
          \n\
";
// This will animate the credit text.
for (int i = 0; i < 50; i ++)
{
  printf("\n"); // go to the bottom of the console while creating free spaces
```



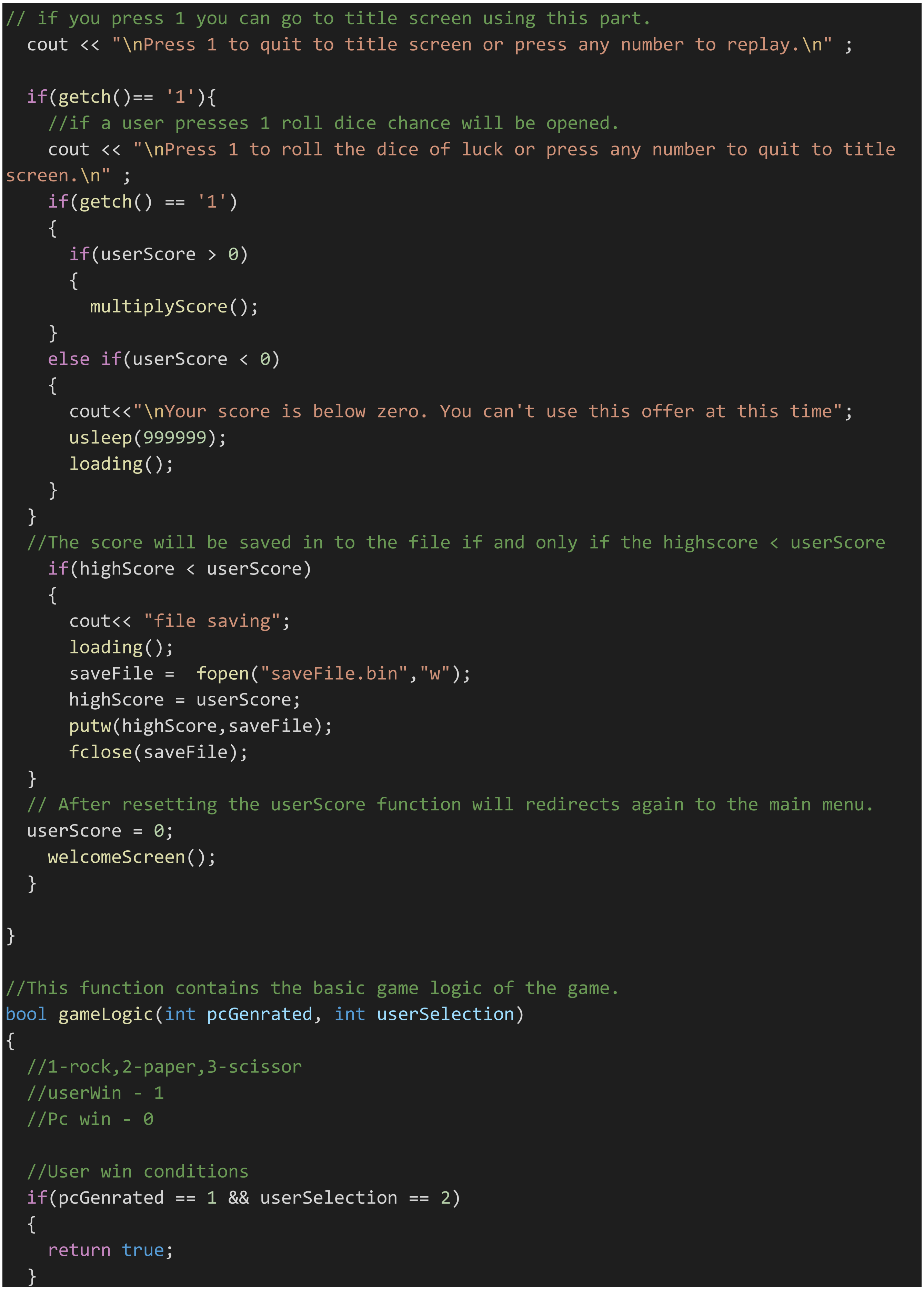
```
}
printf("%s", rocket);
int j = 399999;
for (int i = 0; i < 30; i ++) {
        usleep(j);
        printf("\n"); // move rocket a line upward
    }
  printf("Thanks you for playing!");
  usleep(999999);
  usleep(99999);
  system("CLS");
  welcomeScreen();

}

// You can quit the game through this function."exit(0)" is used inside this.
void QuitGame()
{

system("CLS");

cout << endl;cout << endl;
cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout <<
endl;cout << endl;cout << endl;cout << endl;cout << endl;
cout << endl;cout << endl;
cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout <<
endl;cout << endl;cout << endl;cout << endl;cout << endl;
cout << endl;cout << endl;
cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout <<
endl;cout << endl;cout << endl;cout << endl;cout << endl;
cout <<
"                                             .oooooo.                              .o
8       oooooooooo.                        .o. \n";
cout <<
"                                            d8P'  `Y8b                             88
8       `888'   `Y8b                       888 \n";
cout <<
"                                           888            .ooooo.   .ooooo.   .oooo88
8        888     888 oooo    ooo  .ooooo.  888 \n";
cout << "                                           888           d88' `88b d88' `88b
d88' `888        888oooo888'  `88.  .8'  d88' `88b Y8P \n";
cout << "                                           888     ooooo 888   888 888   888
888   888        888    `88b   `88..8'   888ooo888 `8' \n";
cout << "                                           `88.    .88'  888   888 888   888
888   888        888    .88P    `888'    888    .o .o. \n";
cout << "                                            `Y8bood8P'   `Y8bod8P' `Y8bod8P'
`Y8bod88P       o888bood8P'      .8'     `Y8bod8P' Y8P \n";
```



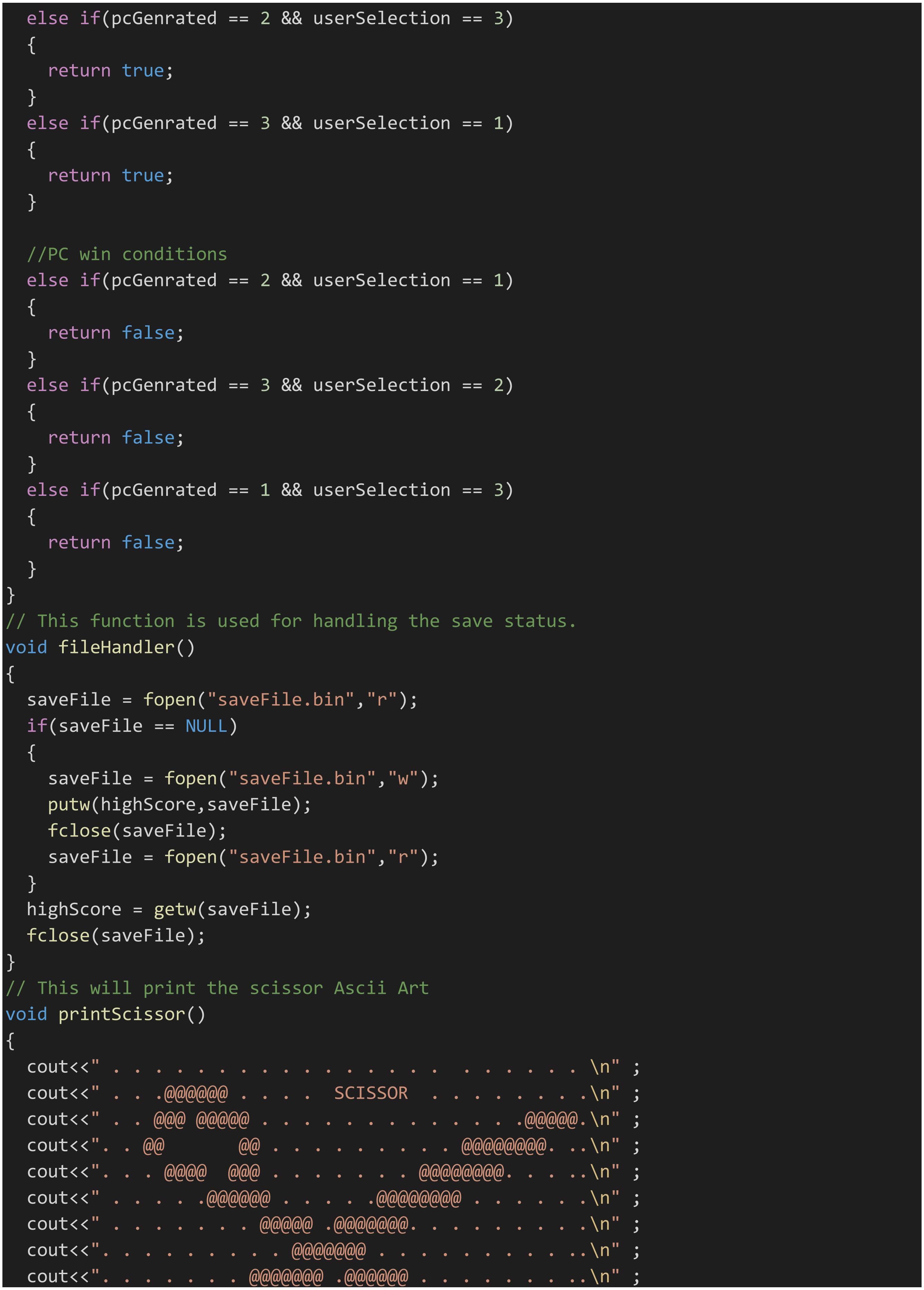
```
cout <<
"
                     .o..P'                    \n";
cout <<
"
                     `Y8P'                     \n";
cout << endl;cout << endl;
cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout <<
endl;cout << endl;cout << endl;cout << endl;cout << endl;
cout << endl;cout << endl;
exit(0);
}

//This function will be called whenever a user presses an invalid input.
void InvalidInput()
{
  cout << "Something went wrong!\n";
  cout << "Please enter a valid input !";

}



//This function will be called when you press the highscore in the main menu.
void HighScoreViewer()
{
  system("CLS");
  print_HighScoreMenu();
  if(getch())
  {
    welcomeScreen();
  }
}


//This function is used to compare the user input with a random choice generated by
the computer.
void CompareSelection(char n)
{

  cout << "\nComputer is thinking about his openion XD" ;
  loading();
//  delay(2000) ;

  //randomly generate;
  int randomNum;
  randomNum = (rand() % 3) + 1;
  switch(randomNum)
  {
```



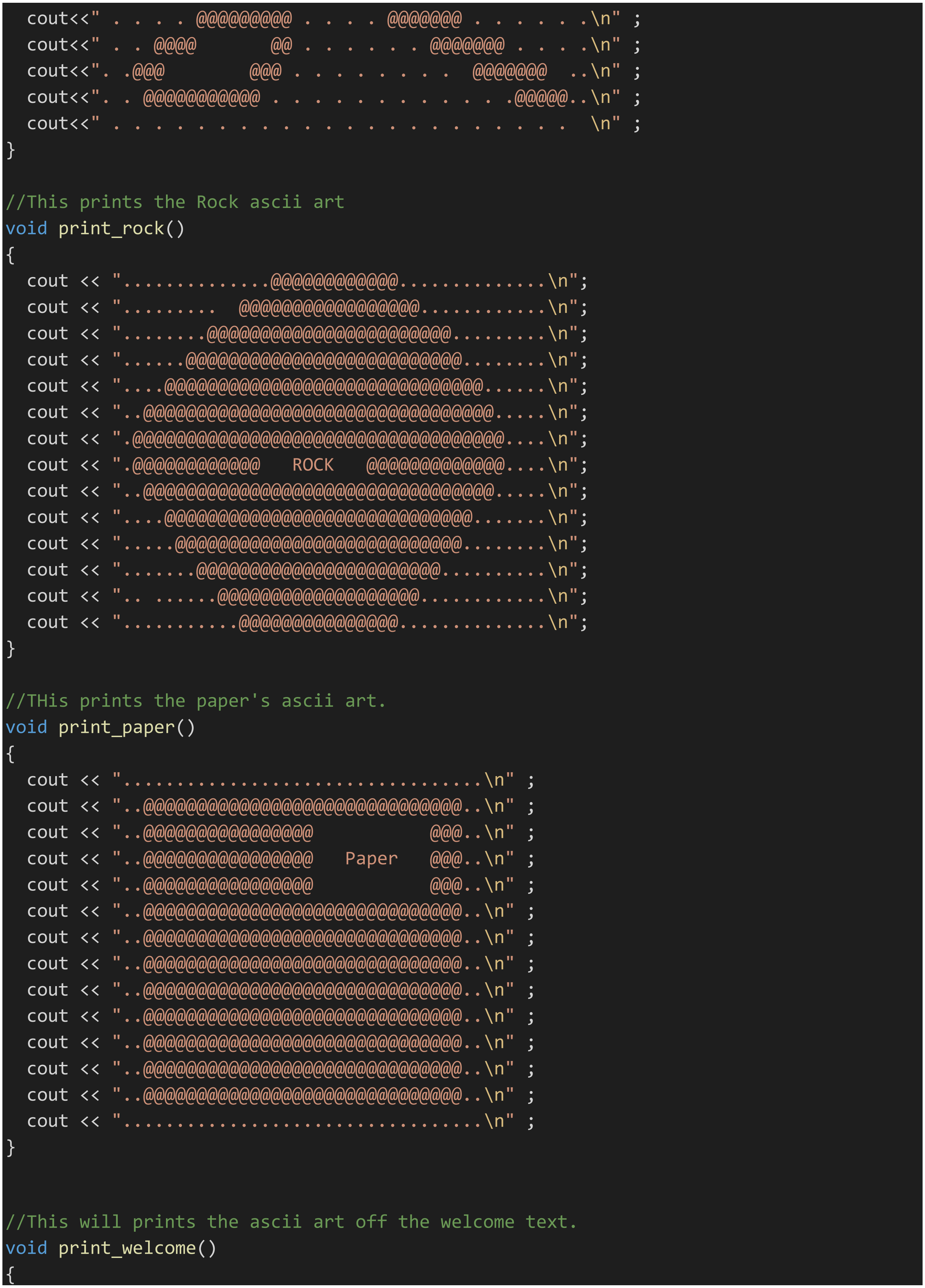
```
  case 1 :
    {
      cout << "\npc has selected rock\n";
      print_rock();
      break;
    }
  case 2 :
    {
      cout << "\npc has selected paper\n";
      print_paper();
      break;
    }
  case 3:
    {
      cout << "\npc has selected scissor\n";
      printScissor();
      break;
    }
   default:
   {
      cout << "Internal error! debug log" ;    //debug log
   }

}

//n is in the char data format. We have to convert it again to int.
int integerN = n ;
integerN = integerN - 48 ;


if(integerN != randomNum)
{
  bool status = gameLogic(randomNum, integerN);
  if(status)
  {
    youWin() ;
    userScore += scoreIncrementUnit;
  }
  else
  {
   youLoss() ;
   userScore -= scoreDecrementUnit;
  }
}
else if(integerN == randomNum)
{
  cout << "\nDraw\n";
}
```



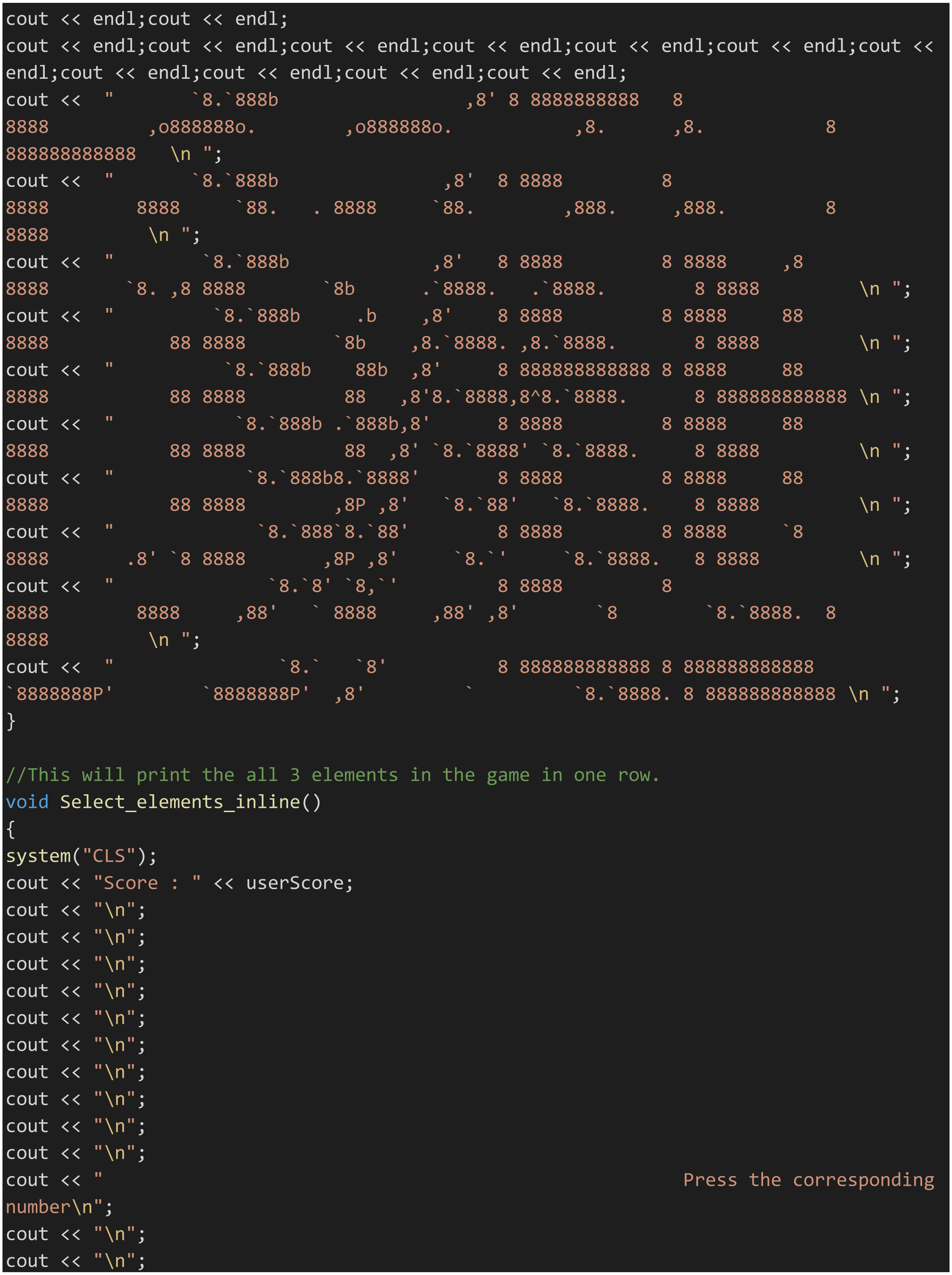
```
// if you press 1 you can go to title screen using this part.
 cout << "\nPress 1 to quit to title screen or press any number to replay.\n" ;

 if(getch()== '1'){
  //if a user presses 1 roll dice chance will be opened.
  cout << "\nPress 1 to roll the dice of luck or press any number to quit to title
screen.\n" ;
  if(getch() == '1')
  {
   if(userScore > 0)
   {
    multiplyScore();
  }
  else if(userScore < 0)
  {
   cout<<"\nYour score is below zero. You can't use this offer at this time";
   usleep(999999);
   loading();
  }
 }
 //The score will be saved in to the file if and only if the highscore < userScore
  if(highScore < userScore)
  {
   cout<< "file saving";
   loading();
   saveFile =  fopen("saveFile.bin","w");
   highScore = userScore;
   putw(highScore,saveFile);
   fclose(saveFile);
 }
 // After resetting the userScore function will redirects again to the main menu.
 userScore = 0;
  welcomeScreen();
 }

}

//This function contains the basic game logic of the game.
bool gameLogic(int pcGenrated, int userSelection)
{
 //1-rock,2-paper,3-scissor
 //userWin - 1
 //Pc win - 0

 //User win conditions
 if(pcGenrated == 1 && userSelection == 2)
 {
  return true;
 }
```



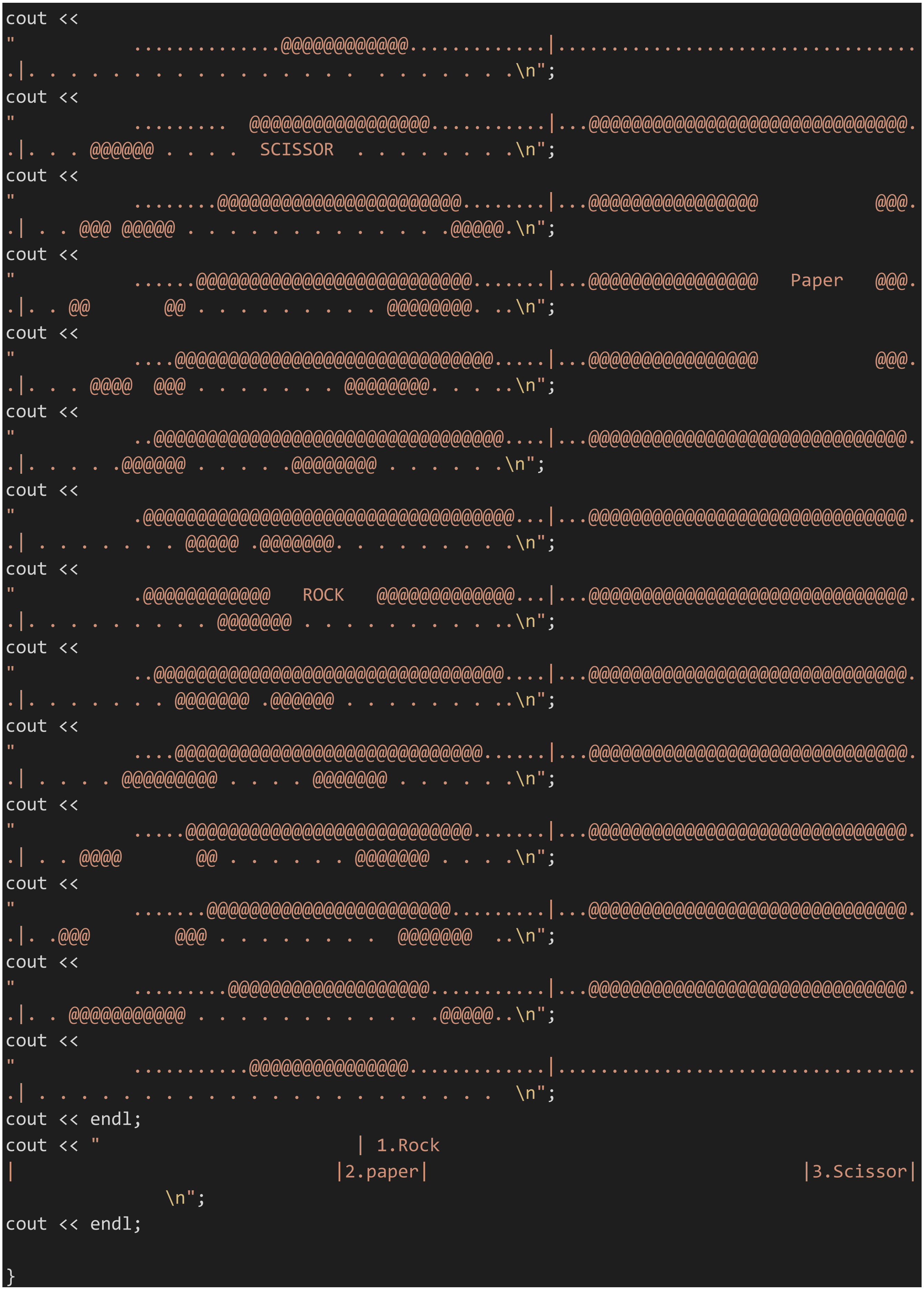
```
  else if(pcGenrated == 2 && userSelection == 3)
  {
    return true;
  }
  else if(pcGenrated == 3 && userSelection == 1)
  {
    return true;
  }

  //PC win conditions
  else if(pcGenrated == 2 && userSelection == 1)
  {
    return false;
  }
  else if(pcGenrated == 3 && userSelection == 2)
  {
    return false;
  }
  else if(pcGenrated == 1 && userSelection == 3)
  {
    return false;
  }
}
// This function is used for handling the save status.
void fileHandler()
{
  saveFile = fopen("saveFile.bin","r");
  if(saveFile == NULL)
  {
    saveFile = fopen("saveFile.bin","w");
    putw(highScore,saveFile);
    fclose(saveFile);
    saveFile = fopen("saveFile.bin","r");
  }
  highScore = getw(saveFile);
  fclose(saveFile);
}
// This will print the scissor Ascii Art
void printScissor()
{
  cout<<" . . . . . . . . . . . . . . . .  . . . . . . \n" ;
  cout<<" . . .@@@@@@ . . . .  SCISSOR  . . . . . . . .\n" ;
  cout<<" . . @@@ @@@@@ . . . . . . . . . . . . .@@@@@.\n" ;
  cout<<". . @@       @@ . . . . . . . . . @@@@@@@@. ..\n" ;
  cout<<". . . @@@@  @@@ . . . . . . . @@@@@@@@. . . ..\n" ;
  cout<<" . . . . .@@@@@@ . . . . .@@@@@@@@ . . . . . .\n" ;
  cout<<" . . . . . . . @@@@@ .@@@@@@@. . . . . . . . .\n" ;
  cout<<". . . . . . . . . @@@@@@@ . . . . . . . . . . ..\n" ;
  cout<<". . . . . . . @@@@@@@ .@@@@@@ . . . . . . . . ..\n" ;
```



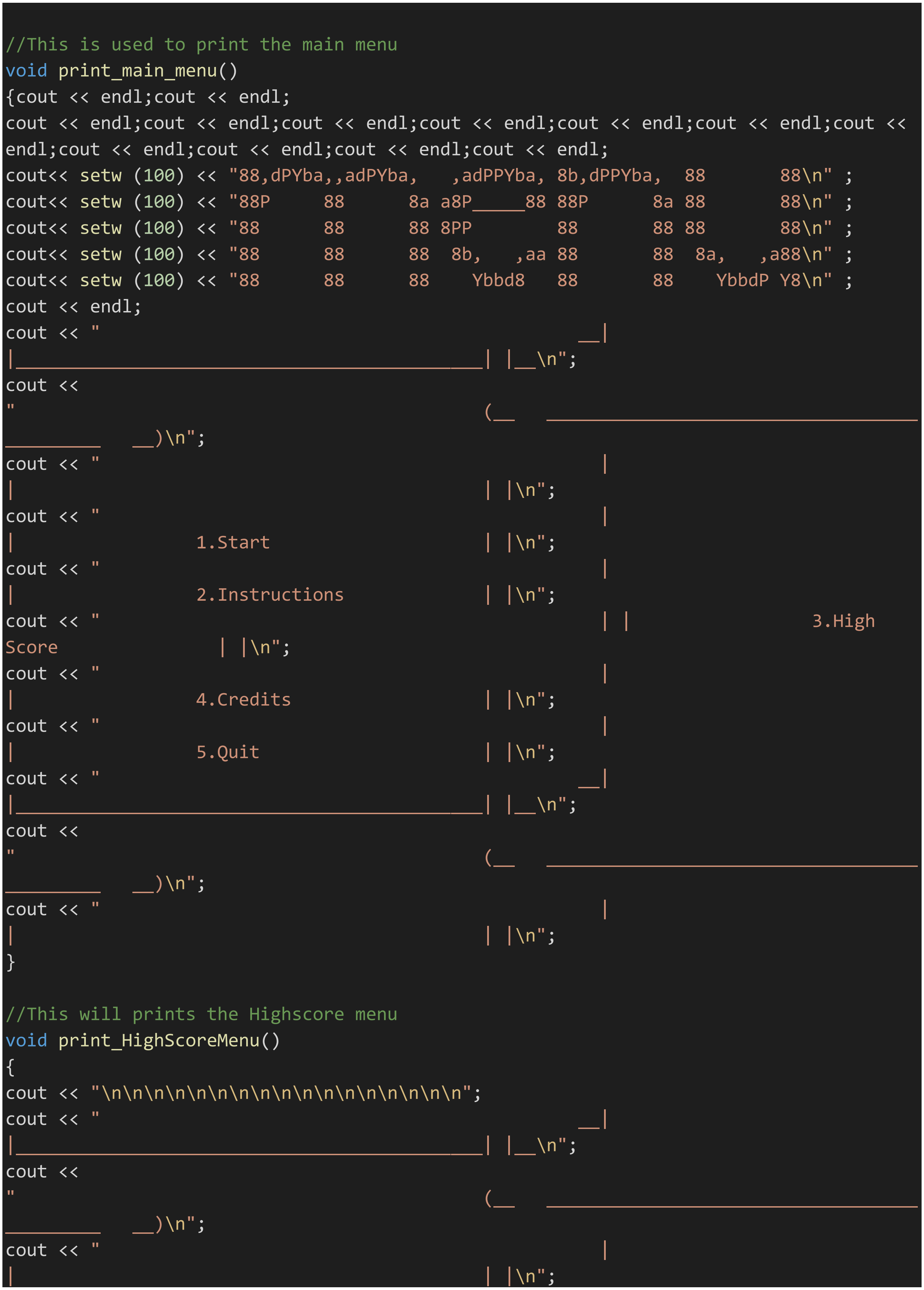
```
  cout<<" . . . . @@@@@@@@@ . . . . @@@@@@@ . . . . . .\n" ;
  cout<<" . . @@@@       @@ . . . . . . @@@@@@@ . . . .\n" ;
  cout<<". .@@@        @@@ . . . . . . . .  @@@@@@@  ..\n" ;
  cout<<". . @@@@@@@@@@@ . . . . . . . . . . . .@@@@@..\n" ;
  cout<<" . . . . . . . . . . . . . . . . . . . . . .  \n" ;
}

//This prints the Rock ascii art
void print_rock()
{
  cout << "..............@@@@@@@@@@@@..............\n";
  cout << ".........  @@@@@@@@@@@@@@@@@............\n";
  cout << "........@@@@@@@@@@@@@@@@@@@@@@@.........\n";
  cout << "......@@@@@@@@@@@@@@@@@@@@@@@@@@........\n";
  cout << "....@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@......\n";
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.....\n";
  cout << ".@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@....\n";
  cout << ".@@@@@@@@@@@@   ROCK   @@@@@@@@@@@@@....\n";
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.....\n";
  cout << "....@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.......\n";
  cout << ".....@@@@@@@@@@@@@@@@@@@@@@@@@@@........\n";
  cout << ".......@@@@@@@@@@@@@@@@@@@@@@@..........\n";
  cout << ".. ......@@@@@@@@@@@@@@@@@@@............\n";
  cout << "...........@@@@@@@@@@@@@@@..............\n";
}

//THis prints the paper's ascii art.
void print_paper()
{
  cout << "..................................\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@           @@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@   Paper   @@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@           @@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..................................\n" ;
}


//This will prints the ascii art off the welcome text.
void print_welcome()
{
```



```
cout << endl;cout << endl;
cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout <<
endl;cout << endl;cout << endl;cout << endl;cout << endl;
cout <<  "       `8.`888b                 ,8' 8 8888888888   8
8888         ,o888888o.        ,o888888o.           ,8.      ,8.           8
888888888888   \n ";
cout <<  "       `8.`888b               ,8'  8 8888         8
8888        8888     `88.   . 8888     `88.        ,888.     ,888.         8
8888         \n ";
cout <<  "        `8.`888b             ,8'   8 8888         8 8888     ,8
8888       `8. ,8 8888       `8b      .`8888.   .`8888.        8 8888         \n ";
cout <<  "         `8.`888b     .b    ,8'    8 8888         8 8888     88
8888           88 8888        `8b    ,8.`8888. ,8.`8888.       8 8888         \n ";
cout <<  "          `8.`888b    88b  ,8'     8 888888888888 8 8888     88
8888           88 8888         88   ,8'8.`8888,8^8.`8888.      8 888888888888 \n ";
cout <<  "           `8.`888b .`888b,8'      8 8888         8 8888     88
8888           88 8888         88  ,8' `8.`8888' `8.`8888.     8 8888         \n ";
cout <<  "            `8.`888b8.`8888'       8 8888         8 8888     88
8888           88 8888        ,8P ,8'   `8.`88'   `8.`8888.    8 8888         \n ";
cout <<  "             `8.`888`8.`88'        8 8888         8 8888     `8
8888       .8' `8 8888       ,8P ,8'     `8.`'     `8.`8888.   8 8888         \n ";
cout <<  "              `8.`8' `8,`'         8 8888         8
8888        8888     ,88'   ` 8888     ,88' ,8'       `8        `8.`8888.  8
8888         \n ";
cout <<  "               `8.`   `8'          8 888888888888 8 888888888888
`8888888P'        `8888888P'  ,8'         `         `8.`8888. 8 888888888888 \n ";
}

//This will print the all 3 elements in the game in one row.
void Select_elements_inline()
{
system("CLS");
cout << "Score : " << userScore;
cout << "\n";
cout << "\n";
cout << "\n";
cout << "\n";
cout << "\n";
cout << "\n";
cout << "\n";
cout << "\n";
cout << "\n";
cout << "\n";
cout << "                                                     Press the corresponding
number\n";
cout << "\n";
cout << "\n";
```

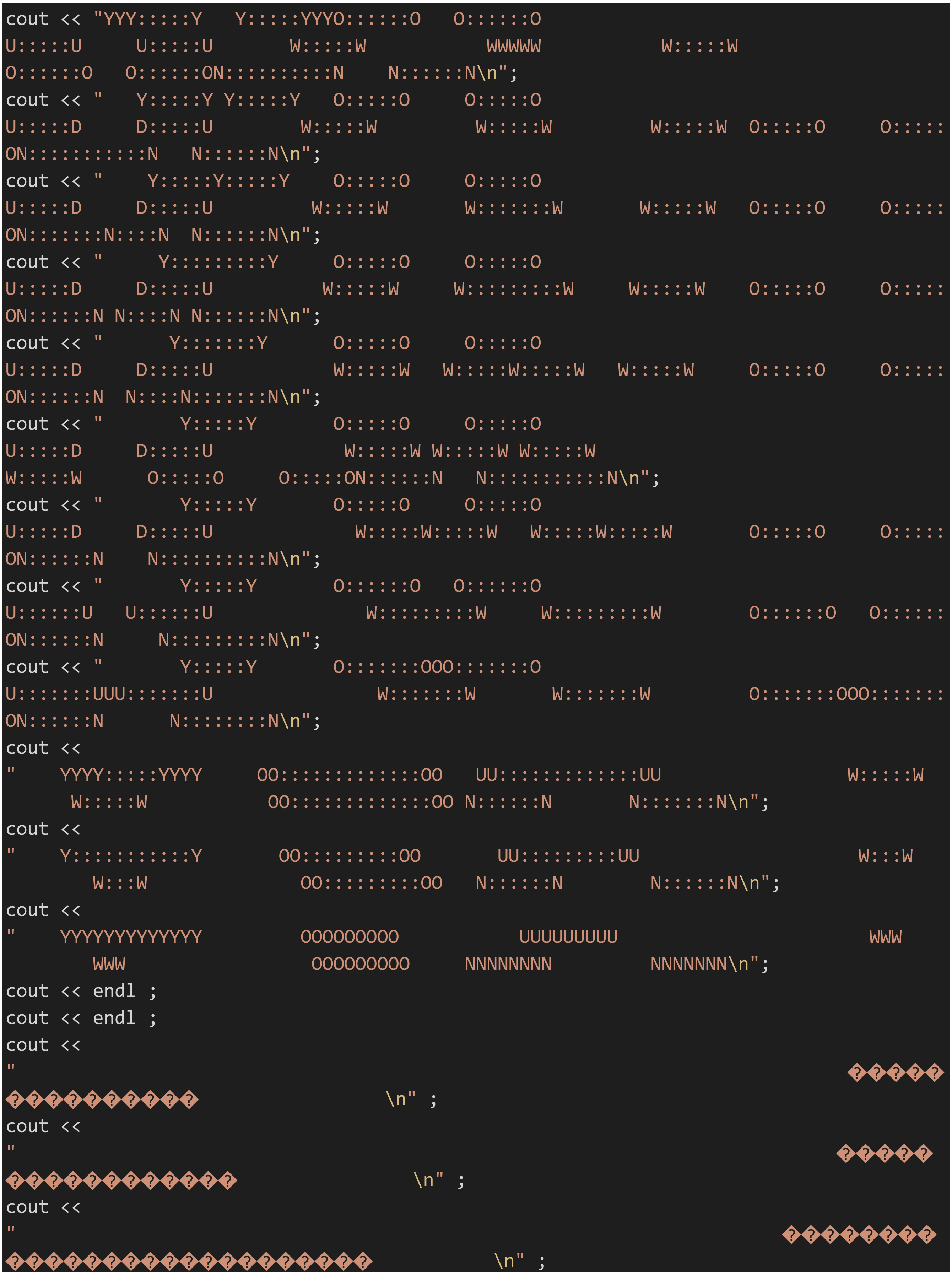


```
cout <<
"           ..............@@@@@@@@@@@@@.............|..................................
.|. . . . . . . . . . . . . . . .   . . . . . . .\n";
cout <<
"           .........  @@@@@@@@@@@@@@@@@@...........|...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
.|. . . @@@@@@ . . . .   SCISSOR   . . . . . . . .\n";
cout <<
"           ........@@@@@@@@@@@@@@@@@@@@@@@@........|...@@@@@@@@@@@@@@@@           @@@.
.| . . @@@ @@@@@ . . . . . . . . . . . . .@@@@@.\n";
cout <<
"           ......@@@@@@@@@@@@@@@@@@@@@@@@@@@.......|...@@@@@@@@@@@@@@@@   Paper   @@@.
.|. . @@       @@ . . . . . . . . . @@@@@@@@. ..\n";
cout <<
"           ....@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.....|...@@@@@@@@@@@@@@@@           @@@.
.|. . . @@@@  @@@ . . . . . . . @@@@@@@@. . . ..\n";
cout <<
"           ..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@....|...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
.|. . . . .@@@@@@ . . . . .@@@@@@@@ . . . . . .\n";
cout <<
"           .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@...|...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
.| . . . . . . . @@@@@ .@@@@@@@. . . . . . . . .\n";
cout <<
"           .@@@@@@@@@@@@   ROCK   @@@@@@@@@@@@@@...|...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
.|. . . . . . . . . @@@@@@@ . . . . . . . . . ..\n";
cout <<
"           ..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@....|...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
.|. . . . . . . @@@@@@@ .@@@@@@ . . . . . . . ..\n";
cout <<
"           ....@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@......|...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
.| . . . . @@@@@@@@@ . . . . @@@@@@@ . . . . . .\n";
cout <<
"           .....@@@@@@@@@@@@@@@@@@@@@@@@@@@@.......|...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
.| . . @@@@       @@ . . . . . . @@@@@@@ . . . .\n";
cout <<
"           .......@@@@@@@@@@@@@@@@@@@@@@@@.........|...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
.|. .@@@        @@@ . . . . . . . .  @@@@@@@  ..\n";
cout <<
"           .........@@@@@@@@@@@@@@@@@@@@...........|...@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
.|. . @@@@@@@@@@@ . . . . . . . . . . . .@@@@@..\n";
cout <<
"           ...........@@@@@@@@@@@@@@@@.............|..................................
.| . . . . . . . . . . . . . . . . . . . . . .   \n";
cout << endl;
cout << "                         | 1.Rock
|                               |2.paper|                                    |3.Scissor|
               \n";
cout << endl;

}
```



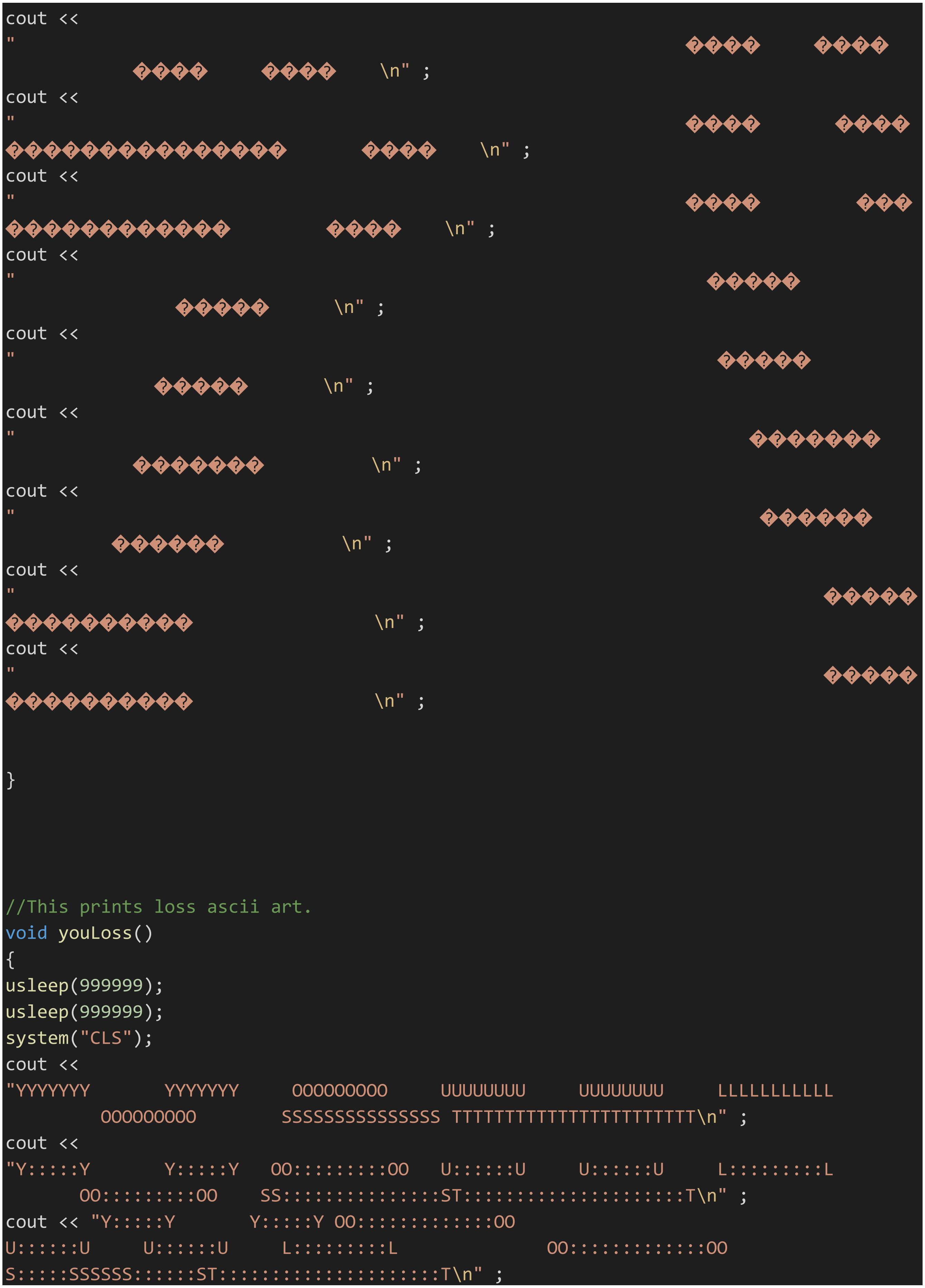
```
//This is used to print the main menu
void print_main_menu()
{cout << endl;cout << endl;
cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout << endl;cout <<
endl;cout << endl;cout << endl;cout << endl;cout << endl;
cout<< setw (100) << "88,dPYba,,adPYba,   ,adPPYba, 8b,dPPYba,  88       88\n" ;
cout<< setw (100) << "88P     88      8a a8P_____88 88P      8a 88       88\n" ;
cout<< setw (100) << "88      88      88 8PP        88       88 88       88\n" ;
cout<< setw (100) << "88      88      88  8b,   ,aa 88       88  8a,   ,a88\n" ;
cout<< setw (100) << "88      88      88    Ybbd8   88       88    YbbdP Y8\n" ;
cout << endl;
cout << "                                             __|
|____________________________________________| |__\n";
cout <<
"                                            (__   ___________________________________
_________   __)\n";
cout << "                                               |
|                                            | |\n";
cout << "                                               |
|                 1.Start                    | |\n";
cout << "                                               |
|                 2.Instructions             | |\n";
cout << "                                               | |                 3.High
Score               | |\n";
cout << "                                               |
|                 4.Credits                  | |\n";
cout << "                                               |
|                 5.Quit                     | |\n";
cout << "                                             __|
|____________________________________________| |__\n";
cout <<
"                                            (__   ___________________________________
_________   __)\n";
cout << "                                               |
|                                            | |\n";
}

//This will prints the Highscore menu
void print_HighScoreMenu()
{
cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n";
cout << "                                             __|
|____________________________________________| |__\n";
cout <<
"                                            (__   ___________________________________
_________   __)\n";
cout << "                                               |
|                                            | |\n";
```

```
cout << "                                                | |          The current High
Score :          | |\n";
cout << "                                                |
|                                            | |\n";
cout << "                                                | |                    "<<setw
(5)<<highScore<<"                   | |\n";
cout << "                                                |
|                                            | |\n";
cout << "                                                | |    Press any key to return
to main menu    | |\n";
cout << "                                              __|
|____________________________________________| |__\n";
cout <<
"                                            (__   ____________________________________
_________   __)\n";
cout << "                                                |
|                                            | |\n";
}

//Loading animation
void loading()
{
  for(int i=0; i< 3;i++)
  {
    cout <<" .";
    usleep(999999);
  }
  cout <<"\n";
}


// This prints win ascii art.
void youWin()
{
usleep(999999);
usleep(999999);
system("CLS");
cout <<
"YYYYYYY       YYYYYYY     OOOOOOOOO     UUUUUUUU      UUUUUUUU     WWWWWWWW
                WWWWWWWW     OOOOOOOOO     NNNNNNNN        NNNNNNNN\n";
cout <<
"Y:::::Y       Y:::::Y   OO:::::::::OO   U::::::U      U::::::U     W::::::W
                W::::::W   OO:::::::::OO   N:::::::N       N::::::N\n";
cout << "Y:::::Y       Y:::::Y OO:::::::::::::OO
U::::::U     U::::::U     W::::::W                            W::::::W
OO:::::::::::::OO N::::::::N      N::::::N\n";
cout <<
"Y::::::Y     Y::::::YO:::::::OOO:::::::OUU:::::U      U:::::UU     W::::::W
                W::::::WO:::::::OOO:::::::ON:::::::::N     N::::::N\n";
```

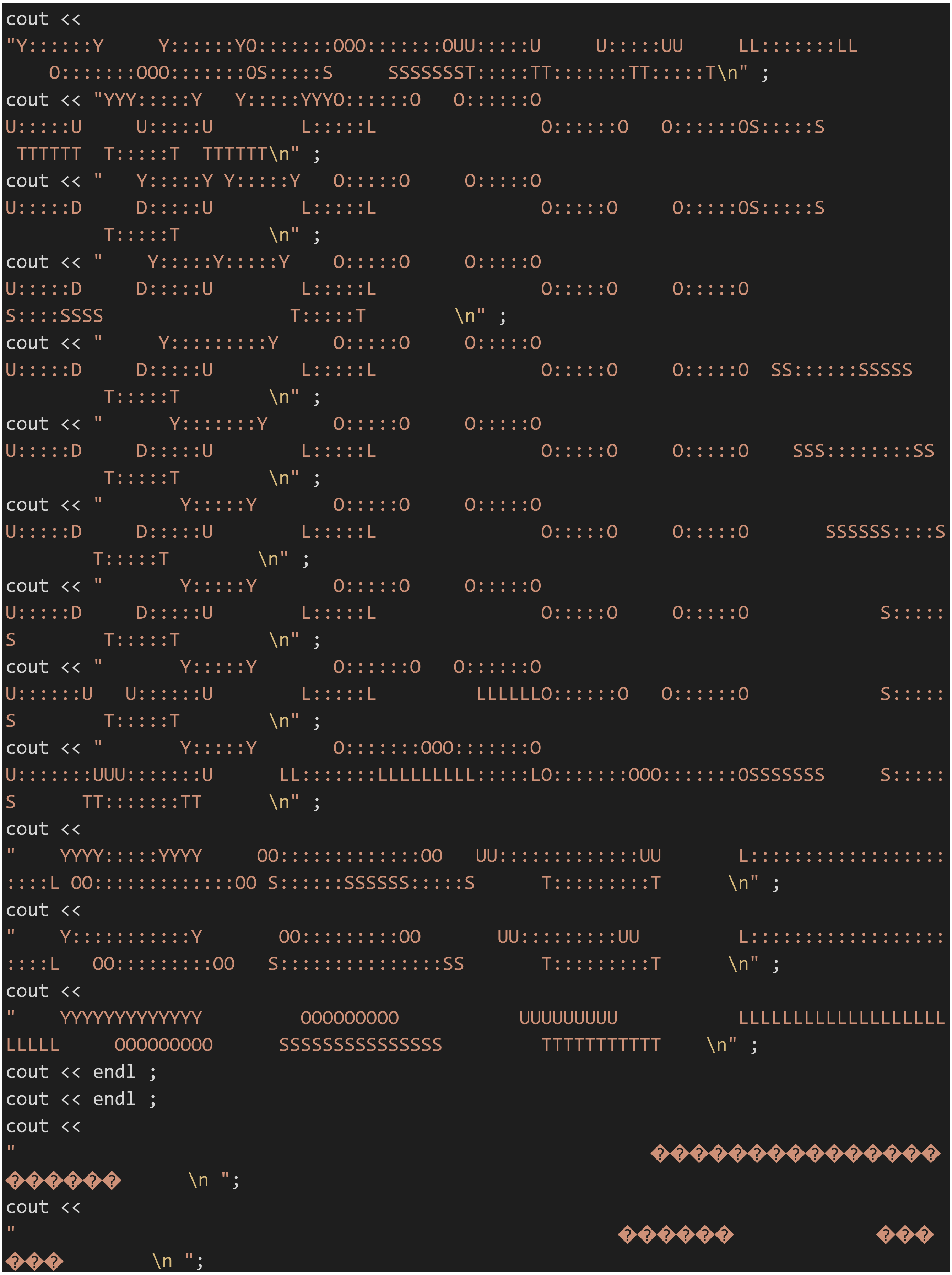


```
cout << "YYY:::::Y   Y:::::YYYO::::::O   O::::::O
U:::::U     U:::::U       W:::::W           WWWWW           W:::::W
O::::::O   O::::::ON::::::::::N    N::::::N\n";
cout << "   Y:::::Y Y:::::Y   O:::::O     O:::::O
U:::::D     D:::::U        W:::::W         W:::::W         W:::::W  O:::::O     O:::::
ON:::::::::::N   N::::::N\n";
cout << "    Y:::::Y:::::Y    O:::::O     O:::::O
U:::::D     D:::::U         W:::::W       W:::::::W       W:::::W   O:::::O     O:::::
ON::::::N:::::N  N::::::N\n";
cout << "     Y:::::::::Y     O:::::O     O:::::O
U:::::D     D:::::U          W:::::W     W:::::::::W     W:::::W    O:::::O     O:::::
ON::::::N N::::N N::::::N\n";
cout << "      Y:::::::Y      O:::::O     O:::::O
U:::::D     D:::::U           W:::::W   W:::::W:::::W   W:::::W     O:::::O     O:::::
ON::::::N  N::::N:::::::N\n";
cout << "       Y:::::Y       O:::::O     O:::::O
U:::::D     D:::::U            W:::::W W:::::W W:::::W
W:::::W      O:::::O     O:::::ON::::::N   N:::::::::::N\n";
cout << "       Y:::::Y       O:::::O     O:::::O
U:::::D     D:::::U             W:::::W:::::W   W:::::W:::::W       O:::::O     O:::::
ON::::::N    N::::::::::N\n";
cout << "       Y:::::Y       O::::::O   O::::::O
U::::::U   U::::::U              W:::::::::W     W:::::::::W        O::::::O   O::::::
ON::::::N     N:::::::::N\n";
cout << "       Y:::::Y       O:::::::OOO:::::::O
U:::::::UUU:::::::U               W:::::::W       W:::::::W         O:::::::OOO:::::::
ON::::::N      N::::::::N\n";
cout <<
"    YYYY:::::YYYY     OO:::::::::::::OO   UU:::::::::::::UU                 W:::::W
      W:::::W           OO:::::::::::::OO N::::::N       N:::::::N\n";
cout <<
"    Y:::::::::::Y       OO:::::::::OO       UU:::::::::UU                    W:::W
        W:::W              OO:::::::::OO   N::::::N        N::::::N\n";
cout <<
"    YYYYYYYYYYYYY         OOOOOOOOO           UUUUUUUUU                       WWW
        WWW                 OOOOOOOOO     NNNNNNNN         NNNNNNN\n";
cout << endl ;
cout << endl ;
cout <<
"                                                                            �����
����������                 \n" ;
cout <<
"                                                                           �����
������������                \n" ;
cout <<
"                                                                      ��������
�������������������           \n" ;
```

```
cout <<
"                                                                       �������
            �������          \n" ;
cout <<
"                                                                    �����
              �����       \n" ;
cout <<
"                                                                   �����
                �����      \n" ;
cout <<
"                                                                 ����
                  ����    \n" ;
cout <<
"                                                                 ����
                  ����    \n" ;
cout <<
"                                                                 ����     �����
��         �������     ����    \n" ;
cout <<
"                                                                ���       ������
�         �������       ������\n" ;
cout <<
"                                                              ����        �����
��         �������         ����\n" ;
cout <<
"                                                              ����
                        ����\n" ;
cout <<
"                                                              ����
                        ����\n" ;
cout <<
"                                                              ����
                        ����\n" ;
cout <<
"                                                              ����
                        ����\n" ;
cout <<
"                                                              ����
                        ����\n" ;
cout <<
"                                                              ����      �
              �       ����\n" ;
cout <<
"                                                              ����     ����
                ����      ����\n" ;
cout <<
"                                                                ���     �����
              �����     ������\n" ;
```

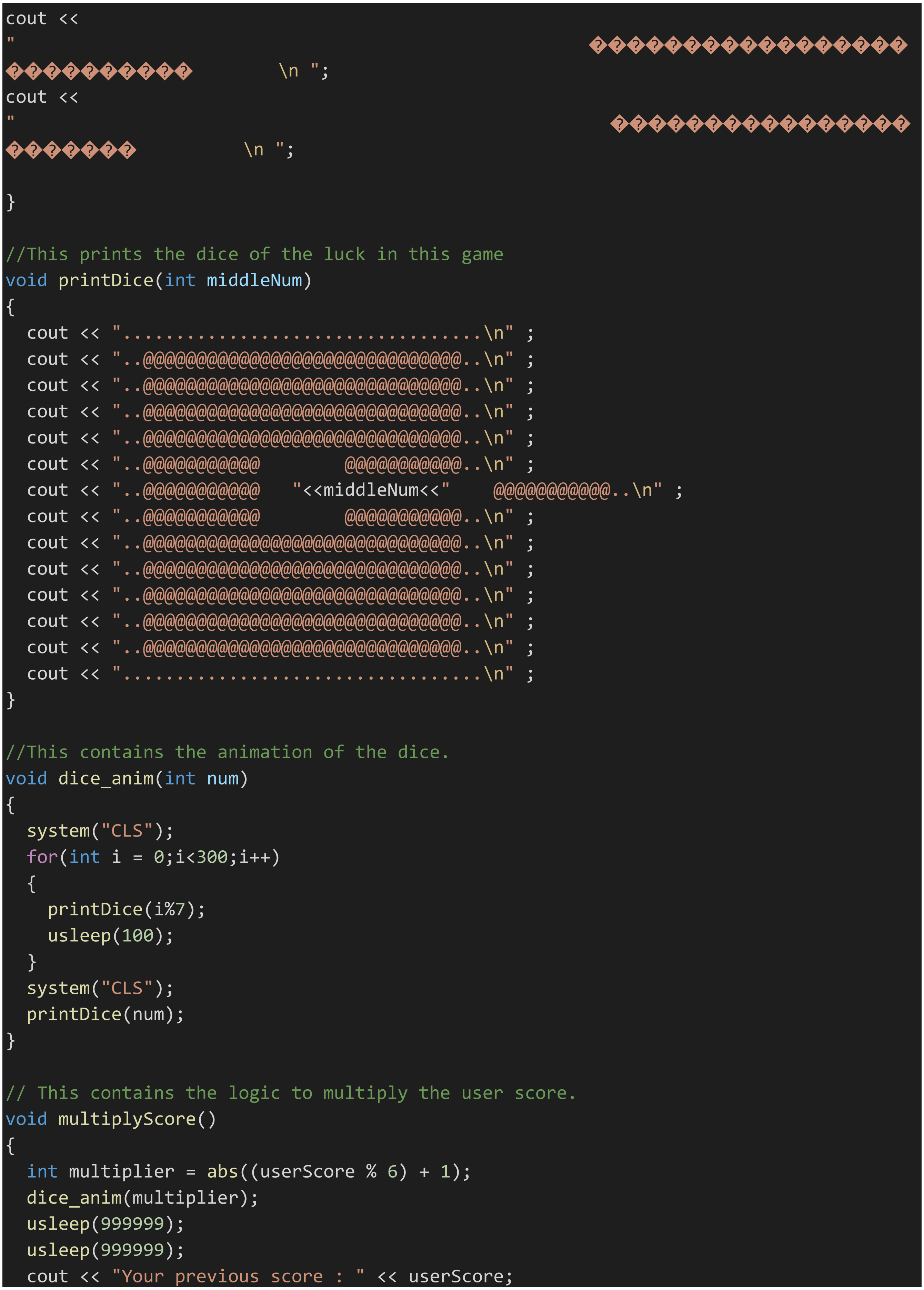


```
cout <<
"                                                                ����     ����
            ����     ����    \n" ;
cout <<
"                                                                ����       ����
���������������       ����    \n" ;
cout <<
"                                                                ����         ���
������������         ����    \n" ;
cout <<
"                                                                  �����
                �����      \n" ;
cout <<
"                                                                   �����
              �����       \n" ;
cout <<
"                                                                      �������
            �������          \n" ;
cout <<
"                                                                       ������
          ������           \n" ;
cout <<
"                                                                             �����
����������                 \n" ;
cout <<
"                                                                             �����
����������                 \n" ;


}




//This prints loss ascii art.
void youLoss()
{
usleep(999999);
usleep(999999);
system("CLS");
cout <<
"YYYYYYY       YYYYYYY      OOOOOOOOO     UUUUUUUU     UUUUUUUU     LLLLLLLLLLL
         OOOOOOOOO        SSSSSSSSSSSSSSS TTTTTTTTTTTTTTTTTTTTTTT\n" ;
cout <<
"Y:::::Y       Y:::::Y   OO:::::::::OO   U::::::U     U::::::U     L:::::::::L
       OO:::::::::OO    SS:::::::::::::::ST::::::::::::::::::::::T\n" ;
cout << "Y:::::Y       Y:::::Y OO:::::::::::::OO
U::::::U     U::::::U     L:::::::::L               OO:::::::::::::OO
S:::::SSSSSS::::::ST::::::::::::::::::::::T\n" ;
```



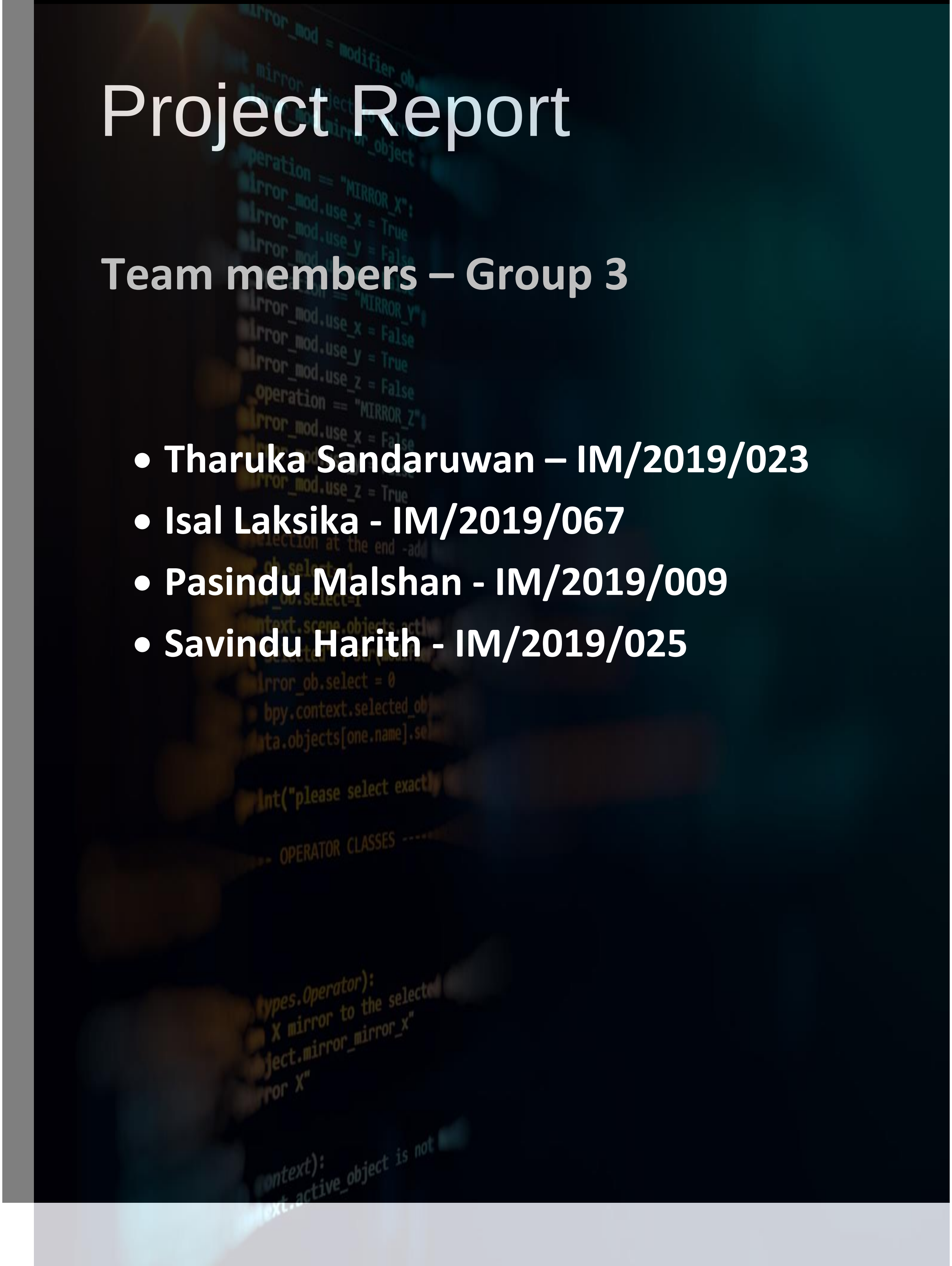
```
cout <<
"Y::::::Y     Y::::::YO:::::::OOO:::::::OUU:::::U    U:::::UU     LL:::::::LL
    O:::::::OOO:::::::OS:::::S     SSSSSSST:::::TT:::::::TT:::::T\n" ;
cout << "YYY:::::Y   Y:::::YYYO::::::O   O::::::O
U:::::U     U:::::U        L:::::L               O::::::O   O::::::OS:::::S
 TTTTTT  T:::::T  TTTTTT\n" ;
cout << "   Y:::::Y Y:::::Y   O:::::O     O:::::O
U:::::D     D:::::U        L:::::L               O:::::O     O:::::OS:::::S
         T:::::T        \n" ;
cout << "    Y:::::Y:::::Y    O:::::O     O:::::O
U:::::D     D:::::U        L:::::L               O:::::O     O:::::O
S::::SSSS                 T:::::T        \n" ;
cout << "     Y:::::::::Y     O:::::O     O:::::O
U:::::D     D:::::U        L:::::L               O:::::O     O:::::O  SS::::::SSSSS
         T:::::T        \n" ;
cout << "      Y:::::::Y      O:::::O     O:::::O
U:::::D     D:::::U        L:::::L               O:::::O     O:::::O    SSS::::::::SS
         T:::::T        \n" ;
cout << "       Y:::::Y       O:::::O     O:::::O
U:::::D     D:::::U        L:::::L               O:::::O     O:::::O       SSSSSS::::S
        T:::::T        \n" ;
cout << "       Y:::::Y       O:::::O     O:::::O
U:::::D     D:::::U        L:::::L               O:::::O     O:::::O            S:::::
S        T:::::T        \n" ;
cout << "       Y:::::Y       O::::::O   O::::::O
U::::::U   U::::::U        L:::::L         LLLLLLO::::::O   O::::::O            S:::::
S        T:::::T        \n" ;
cout << "       Y:::::Y       O:::::::OOO:::::::O
U:::::::UUU:::::::U      LL:::::::LLLLLLLLL:::::LO:::::::OOO:::::::OSSSSSSS     S:::::
S      TT:::::::TT      \n" ;
cout <<
"    YYYY:::::YYYY     OO:::::::::::::OO   UU:::::::::::::UU       L::::::::::::::::::
::::L OO:::::::::::::OO S::::::SSSSSS:::::S      T:::::::::T      \n" ;
cout <<
"    Y:::::::::::Y       OO:::::::::OO       UU:::::::::UU         L::::::::::::::::::
::::L   OO:::::::::OO   S:::::::::::::::SS       T:::::::::T      \n" ;
cout <<
"    YYYYYYYYYYYYY         OOOOOOOOO           UUUUUUUUU           LLLLLLLLLLLLLLLLLLL
LLLLL     OOOOOOOOO      SSSSSSSSSSSSSSS        TTTTTTTTTTT    \n" ;
cout << endl ;
cout << endl ;
cout <<
"                                                          ���������������
������      \n ";
cout <<
"                                                       ������             ���
���        \n ";
```

```
cout <<
"                                                       �����                 ��
����       \n ";
cout <<
"                                                      ����                     ��
���      \n ";
cout <<
"                                                     ����                        �
����     \n ";
cout <<
"                                                    ����     ����       ����
     ���     \n ";
cout <<
"                                                    ���     ������     ����
��     ����    \n ";
cout <<
"                                                   ����     ������     ����
��      ���    \n ";
cout <<
"                                                   ���       ����       ����
      ����   \n ";
cout <<
"                                                   ���
 ���   \n ";
cout <<
"                                                   ���
 ���   \n ";
cout <<
"                                                   ���
 ���   \n ";
cout <<
"                                                   ���            �����
   ����   \n ";
cout <<
"                                                   ����        ������������
        ���    \n ";
cout <<
"                                                    ���      �����     �����
     ����    \n ";
cout <<
"                                                    ����    ���           ���
   ����     \n ";
cout <<
"                                                     ����   ��             ��
����      \n ";
cout <<
"                                                      ����                     ��
���       \n ";
```



```
cout <<
"                                                      �����������������
����������        \n ";
cout <<
"                                                        ����������������
�������          \n ";

}

//This prints the dice of the luck in this game
void printDice(int middleNum)
{
  cout << "..................................\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@        @@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@   "<<middleNum<<"    @@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@        @@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@..\n" ;
  cout << "..................................\n" ;
}

//This contains the animation of the dice.
void dice_anim(int num)
{
  system("CLS");
  for(int i = 0;i<300;i++)
  {
    printDice(i%7);
    usleep(100);
  }
  system("CLS");
  printDice(num);
}

// This contains the logic to multiply the user score.
void multiplyScore()
{
  int multiplier = abs((userScore % 6) + 1);
  dice_anim(multiplier);
  usleep(999999);
  usleep(999999);
  cout << "Your previous score : " << userScore;
  usleep(999999);
  userScore *= multiplier;
  cout<< "\nYour current score : " << userScore<<"\n\n";
  usleep(999999);
  usleep(999999);
}
```

-End-



Project Report

Team members – Group 3

- Tharuka Sandaruwan – IM/2019/023
- Isal Laksika - IM/2019/067
- Pasindu Malshan - IM/2019/009
- Savindu Harith - IM/2019/025