Guess the number

Aim:

In this project, you have to write a C++ program to design a game that involves guessing a number between two randomly generated numbers. You cannot see the randomly generated at start. If you guess correctly, you earn points. If you guess incorrectly, you lose points. You can keep on guessing as long as you have points remaining. You can also choose to display one of the two numbers (either the lower bound or the upper bound) to make the game easier. If you choose to display a number to get help, you earn less points for correct guesses and lose more points for incorrect guesses. You CANNOT, of course, display both bounds. If you COULD display both, there will be no guessing.

```
Start
```

include all header files required for the program

start with decalring variables needed for your program

```
int Score
int LeftBound, RightBound
enum Menu
Menu MenuChoice
string UserName
int Choice
bool ValidUserName
int SuffixIndex
int FirstRandomNum
int SecondRandomNum
int Guessnum
```

Display basic welcome message as directed in project file.

cin >> UserName

if(alphabet or space) — Yes

No

set ValidUserName = false
Ask again to input username

set ValidUserName = true

Convert all the characters of UserName to lowercase using tolower

Then using for loop and if statement convert intials to uppercase using toupper

Now using if-else if statement find the index where "Senior" or "Junior" is found using .find()

Store that index in Suffixindex and use that to replace "Senior" or "Junior" with ", Sr." or ", Jr." accordingly, using .replace()

set srand(time(0))
Using a do-while loop generate assign and generate FirstRandomNum and SecondRandomNum with seeded random numbers until FirstRandomNum < SecondRandomNum

Display LeftBound and RightBound as -1 and -1.

Display Menu for user to select from and prompt to enter a choice using cin and cout

Convert Choice to MenuChoice uding static_cast<> so that we can use switch statement on MenuChoice

Now, comes the main logic of our program write switch case statement over MenuChoice.

case 1:
if RightBound == SecondRandomNum, then display a message stating that user cannot display both bounds at the same time.

else, display a message stating the point variation they will earn or lose as they chooses to display LeftBound

case 2:
if LeftBound == FirstRandomNum, then display a message stating that user cannot display both bounds at the same time.

else, display a message stating the point variation they will earn or lose as they chooses to display RightBound

case 3:
ask user to enter a GuessNum

if GuessNum >= FirstRandomNum and GuessNum <= SecondRandomNum, then update the Score accordingly as mentioned in project file i.e

if LeftBound or RightBound is shown then upadte Score by 1 for every correct ans.

else, update Score by 5 for every correct ans if none of the bounds are displayed.

else,

if LeftBound or RightBound is shown then upadte Score by 10 for every incorrect ans.

else, update Score by 5 for every incorrect ans if none of the bounds are displayed.

Display Remaining points after every guess so that user have an idea for guessing.

Make sure user can only guess until user has positive balance left, otherwise display a goodbye message.

case 4:

As user selects choice 4, do deccreament in Score by 1.

Now, Using a do-while loop generate assign and regenerate FirstRandomNum and SecondRandomNum with seeded random numbers until FirstRandomNum < SecondRandomNum

case 5:

if user enters choice 5, exit the loop and terminate the program by displaying a goodbye message and final point balance.

default :

if user enters anything other than 1,2,3,4 or 5, then display a message stating Wrong choice. Enter again.

end the switch case and return 0;

```
End
```