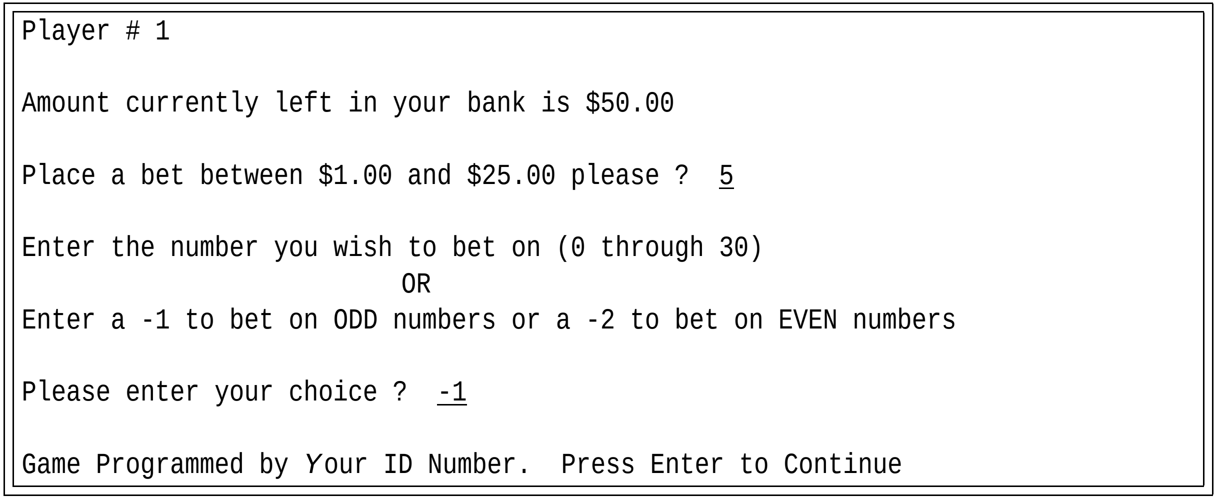
1992 South Jersey High School Programming Contest
Glassboro State College
BASIC Problem

The object of this exercise is to develop a two player game of Roulette called “Mini-Roulette”. For the sake of this contest, the scope of the program will be limited. The Roulette “wheel” will noly have numbers 0 through 30. The program will accept only two types of bets, “numeric” and “Odd/Even”. (Zero counts as an “even” number.) Both players place bets, then the “wheel” spin (simulated by a random number function) produces a result from 0 to 30, and finally the amount won or lost by each player is reported and credited to or deducted from their account (or bank), which starts at $50.00.

User interface:

A separate input screen will be presented to each of the two players. It will inform them of the amount left in their bank and request a bet between two limits. The player can wager no more than $25.00 and no less than $1.00. As no credit will be extended, please reset the upper limit to the amount in his bank if that is less than $25.00 before prompting for the bet. When you prompt for the number to bet on, a response of “-1” means to bet on the next number being odd, a response of “-2” indicates a bet that the next number will be even; all other responses (0 through 30) are bets on single numbers.

Input Screen:



```
Player # 1

Amount currently left in your bank is $50.00

Place a bet between $1.00 and $25.00 please ?  5

Enter the number you wish to bet on (0 through 30)
                          OR
Enter a -1 to bet on ODD numbers or a -2 to bet on EVEN numbers

Please enter your choice ?  -1

Game Programmed by Your ID Number.  Press Enter to Continue
```

The underlined numbers are user responses; the last line is used to help obtain the program output and will be explained when output is discussed.

Should the user make an error, it will be explained to him and the input re-prompted for.

The errors you must trap for are:

A bet that is too high (Over $25.00 or the money remaining in the bank.)

A bet that is too low. (Less than $1.00)

A number choice that is outside the range -2 to 30 inclusive (remember -2 means bet on even numbers, -1 means bet on odd numbers).

A number choice that is not an integer.

(Hint: if INT(number) does not equal the original number, the original number was not an integer.)

If for any reason you cannot accomplish all of this error trapping, continue with your program anyway. You will lose points, but you will not be disqualified.

Continuing play:

Once the bets have been made, the wheel will be spun. The code:

```
WinningNumber = INT(RND * 31)
```

will result in an integer (whole) number between 0 and 30 (Yes, the code says 31, this is no mistake).

For those who need a hint, the following code fragment will help you determine if the Winning number is odd or even:

```
WinningNumber MOD 2
```

This math function returns a zero if WinningNumber is even, a one if WinningNumber is odd. Please note that this code cannot be used on a line by itself.

Paying off bets:

If the player bets a number that matches the winning number, pay him 30 times his bet. If the player bets correctly on an Odd/Even wager, they win the amount of their bet. If they lose, the amount of their bet is subtracted from their bank.

Ending (or continuing) the program:

The program will continue until one player gets $55.00 (or more) in his bank - OR - one player's bank falls below $1.00. The game is then decided on the following rules:

1. If one (or both players) exceed $55.00 in their banks, the player with the highest total wins.

   (a) Should both players have the same amount the game is a tie.

2. Should one player's bank fall below $1.00, the other player wins.

   (a) Should both players go below $1.00 in the same round there is no winner.

A sample output screen is given here:

```
              The Wheel stopped on 12

Player # 1
Bank before Bet:  $50.00
Amount bet:  $10.00
Number bet:  Bet on EVEN numbers
Current Bank:  $60.00

Player # 2
Bank before Bet:  $50.00
Amount bet:  $10.00
Number bet:  7
Current Bank:  $40.00

Player #1 is the Winner

Game Programmed by Your ID Number.  Press Enter to Continue
```

Obtaining output:

As you will notice ever output screen contains "Game Programmed by Your ID Number. Press Enter to Continue". This line is generated by the following code:

```
PRINT
INPUT ”Game Programmed by Your ID Number. Press Enter to Continue”, Junk$
```

These lines are the last lines in both the input and output sections of your program. (Don't forget to replace Your ID Number with <u>the ID number given you</u> when you registered for the contest today).

Note: You will probably want to avoid using line drawn boxes in your actual program input and output screens since they often print slowly.

As this is a game and a game normally has no printed output, we will have to improvise. As directed in the following list, hit the Print Screen button on your keyboard.

When to use the Print Screen button:
(check each when you have printed)

[ ] The first player bets $5.00 on an even number (using -2).
Print the first player's "bet" screen.

[ ] The First player bets $5.00 on an even number (using -2).
The Second player bets $5.00 on an odd number (using -1).
Now, someone has to have won, so print the "win" screen.

[ ] Start the game again! The first player bets $35.00, an error message is shown,
The first player corrects bet to $15.00,
The first player bets on the number 13.
Print the screen with the error message.

[ ] The Second player bets $25.00,
The Second player bets on number 100 (illegal, wrong number), error message is shown,
The second player bets on the number 11.
Print the screen with the error message.

[ ] Continue the game with each player betting $25.00 until one player loses.
Print the final "Win/Lose" Screen.

[ ] Start the game again!
This time each player bets $20.00 on number 13 or number 11 (but not both on the same number) as in the last example until his bank falls below $20.00. The next $20.00 bet causes an error.
Print the Screen with the error message.

The numbers 13 and 11 were chosen arbitrarily. The odds against either number is 31 to one. The reason behind this play is to cause one (or both) players to use up their banks quickly.

This should now give you 6 screen dumps with your ID number on them. List the program and turn in the package.