CS/CE 1337 – PROJECT 4 – Roulette

Pseudocode Due:	CS/CE 1337.001 Mon/Wed class	10/14 beginning of class
(hard copy)	CS/CE 1337.002 Tue/Thu class	10/15 beginning of class
Code Due:	CS/CE 1337.001 Mon/Wed class	10/23 at 11:59 PM
	CS/CE 1337.002 Tue/Thu class	10/24 at 11:59 PM

Submission: The program is to be submitted in eLearning. Please submit the program's .cpp file. **Comments at the top of the program should include your name, course, section, and project type.** Projects can be submitted after the due date with a penalty of 3% per 2 hours late.

Problem: You are to develop a program that simulates playing roulette.

The roulette wheel will be a modified version of an actual roulette wheel. It will only contain the numbers 0–36 (no 00). 0 is neither even, nor odd, nor red, nor black. The numbers 1-36 are associated with a color; either red or black (see http://www.predictem.com/images/roulette.gif).

The gambler will have the option of placing a maximum of 8 bets on the table per "spin". The bets may be of any of the following types:

- Single number (0-36) – pays $35 for each $1 bet
- Red or black – pays $1 for each $1 bet
- Odd or even - pays $1 for each $1 bet
- First 12 (1-12), second 12 (13-24), or third 12 (25-36) - pays $2 for each $1 bet
- First half (1-18) or second half (19-36) - pays $1 for each $1 bet
- First column (1,4,7,10,13,16,19,22,25,28,31,34),
  second column (2,5,8,11,14,17,20,23,26,29,32,35),
  third column (3,6,9,12,15,18,21,24,27,30,33,36) – all columns pay $2 for each $1 bet

Please note that everything the gambler wins is in excess of the current bet. If a gambler bets $2 on a single number and wins he/she keeps the $2 bet and the casino gives him/her $70.

All information concerning the bets (type of bet, amount of bet, net gain, etc.) will be held in a dynamic array (possibly a multi-dimensional array). This array should be destroyed after each spin and a new one created for the next series of bets. The size of the array is determined by the number of bets placed per spin.

The gambler may never place a bet that would cause him/her to have negative dollars. The program should keep track of all bets placed for a spin and never let the gambler bet more than he/she has available. For example, if the gambler tries to place 4 bets and the third bet exceeds the amount of money available, that bet and all future bets that round are voided. The bet which exceeds the available amount should be set to 0 and the gambler should not be prompted for any more bets for that spin.

After collecting the bets from the gambler, spin the roulette wheel and determine the net gain of the gambler. Each spin should be a randomly generated number from 0 – 36. The gambler may leave the table after any spin. Once a bet is placed, the gambler must wait until the outcome of the spin to stop.

Input: To begin, the gambler should enter the amount of money available to play. Limit the maximum amount of money to begin at 1 million dollars. The gambler must enter a number greater than zero. For each spin, your program should ask for the number of bets to be placed. For each bet placed, you should ask the type of bet and the amount of the bet. I recommend some type of menu system for the type of bet. Every bet must be a positive, whole number greater than zero, and the maximum bet is $500. If the gambler enters a floating point number for the bet, your program should handle it gracefully. How you handle it is up to you; just make sure the program doesn't crash.

After a spin, give the gambler the option to bet again or walk away.

Output: After each bet, display the amount of money left. All output pertaining to money should display the amount in whole dollars (no decimals needed) with a dollar sign. After a spin, indicate the net gain from the bets for that spin and the current amount of money the gambler has.

When the gambler leaves the table, thank the gambler for playing and display the total amount of winnings (or losses). For example, if the player started with $1000 and left the table with $1200, indicate the player won $200.

Miscellaneous: This program should use functions. There is no minimum or maximum. Remember that functions are there to do specific tasks. There should be minimal code in your main function other than function calls. As stated previously, this function must use dynamic arrays for the bets. The array must be destroyed before another dynamic array is created.