lab05

1 // EE231002 Lab05. Blackjack probabilities

2 // 108061112, 林靖

3 // Oct. 18, 2019

4

5 #include <stdio.h> // Standard input and output library

6 #include <stdlib.h> // Standard library for performing rand() 7

8 int main(void) // The function called at program startup 9 {

10 short target; // For each particular target points 11 long numExpr; // Number of experiments performed for each target pt 12 long ttlSuccess; // Total number of successes out of 100k experiments 13 long ttlDeal; // Total number of cards dealt upon all successful tries 14 float probSuccess; // Probability of success in 100k experiments 15 float avgDeal; // Average number of cards dealt upon successful tries 16 short randomVal; // Face value of the card randomly drawn 17 short cumuDeal; // Cumulative number of cards dealt in single experiment 18 short sumPt; // Summation of points in single experiment 19

20 printf("Points Probability #Cards\n"); // Fixed table header 21 for (target = 4; target <= 21; target++) { // Compute from 4 to 21 22 ttlSuccess = 0; // Reinitialize counters 23 ttlDeal = 0; // for each target pt. 24 for (numExpr = 0; numExpr < 100000; numExpr++) {// Play many times

for (numExpr = 0; numExpr < 100000; numExpr++) { // Play many times 25 sumPt = 0; // Initialize for single experiment 26 randomVal = rand() % 13 + 1; // Randomly draw a card 27 switch (randomVal) { // To determine value of drawn card: 28 case 11: // Jacks,

29 case 12: // queens, and

30 case 13: // kings

31 sumPt += 10; // are treated as 10 pt. 32 break;

33 case 1: // Ace is always 11 pt in first deal 34 sumPt += 10; // Add 10 here, and add 1 in line 36 35 default: // The cards 2 to 10 36 sumPt += randomVal; // have the face values. 37 } // The loop below terminates when a conclusive result 38 for (cumuDeal = 2; sumPt < target; cumuDeal++) { // is obtained. 39 randomVal = rand() % 13 + 1; // Randomly draw a card 40 switch (randomVal) { // To determine the value: 41 case 11: // Jacks,

42 case 12: // queens, and 43 case 13: // kings

44 sumPt += 10; // are treated as 10 pt. 45 break;

46 case 1:

47 if (sumPt + 11 <= 21) { // If the total would not exceed 1

48 sumPt += 10; // 21, ace can have 11 pt. (10 49 } // extra over the original 1 pt) 50 default: // The cards 2 to 10 51 sumPt += randomVal; // have the face values. 52 }

53 if (sumPt == target) { // If reach exactly the target, 54 ++ttlSuccess; // then it's a success. 55 ttlDeal += cumuDeal; // Add cumulated cards into the 56 } // total number of cards dealt. 57 }

58 }

59 probSuccess = 100.f \* ttlSuccess / numExpr; // Percentage of success 60 avgDeal = 1.f \* ttlDeal / ttlSuccess; // Calculate the average 61 printf("%3.hi%12.2f%%%10.2f\n", target, probSuccess, avgDeal); 62 } // Print out the results in columns aligning the table head 63 return 0; // Indicates normal program termination

64 }

[Format] can be improved.

[Each experiment] needs to draw at least two cards.

[Coding] can be more concise.

Score: 89

2