lab05

```
$ gcc lab05.c

$ ./a.out
 Points  Probability  #Cards
    4        0.59%      2.00
    5        1.17%      2.00
    6        1.80%      2.02
    7        2.48%      2.05
    8        3.19%      2.09
    9        3.99%      2.12
   10        4.85%      2.15
   11        5.72%      2.20
   12       11.20%      2.18
   13       12.21%      2.25
   14       12.61%      2.35
   15       13.38%      2.46
   16       13.96%      2.58
   17       14.24%      2.67
   18       15.27%      2.79
   19       15.87%      2.92
   20       21.89%      2.80
   21       17.77%      3.17
```

---

score: 89.0
o. [Output] Program output is correct, good.
o. [Coding] lab05.c spelling errors: arbitary(2), probabilaty(1)
o. [Format] Program format can be improved.
o. [Codes] Logic can be simplified.

lab05.c

```c
// EE231002 Lab05. Blackjack Probabilities
// 111061234, 呂恆毅
// Date: 2022/10/17

#include <stdio.h>
#include <stdlib.h>

int main(void)                                  // I/O library
{
    int i, j;                                   // looping index
    int p, sum, determ;                         // points, currently sum,
                                                //     a determination
    int A, ca;                                  // Currently, how many Ace,
                                                //     how many card I have
    double to, canu;                            // total successful times,
                                                //     and sum of the points

    printf(" Points  Probability  #Cards\n");    // output the title
    for (i = 4; i <= 21; i++) {                 // arrange the outputs by
                                                //     its points
        canu = 0;                               // reset variable to 0
        to = 0;                                 // reset variable to 0
        for (j = 0; j < 100000; j++) {          // test for 100000 times
            sum = 0;                            // reset variable to 0
            A = 0;                              // reset variable to 0
            ca = 0;                             // reset variable to 0
            p = rand() % 13 + 1;                // ask for an arbitary point
            ca++;                               // change the amount of cards
                                                // the points J Q K A should
                                                //     represent
            if (p > 10) p = 10;
            else if (p == 1) {
                p = 11;
                A++;                            // change how many A I have
            }
            sum = sum + p;                      // change the sum of points
            do {
                p = rand() % 13 + 1;            // ask for an arbitary point
                                                // the points J Q K A should
                                                //     represent
```

```
41                  if (p > 10) p = 10;
42                  else if (p == 1) {
43                      p = 11;
44                      A++;                            // change how many A I have
45                  }
46                  ca++;                               // change the amount of cards
47                  sum = sum + p;                      // change the sum of points
48                                                      // modify the point of A to 1
49                  if (sum > 21 && A != 0) {           //     necessary
50                      sum = sum - 10;                 // change the sum of points
51                      A--;                            // change how many A I have
52                  }
53                  if (sum == i) {                     // if the sum of points match
54                                                      //     the target card number
55                      to++;                           // change the successful times
56                      canu = canu + ca;               // change the total card
57                                                      //     numbers
58                      determ = 1;                     // change the determination
59                  }
60                  else if (sum > i) determ = 1;       // change the determination
61                  else determ = 0;                    // change the determination
62              }
                } while (determ == 0);                  // determine whether staying
63              while (determ == 0);                    // determine whether staying

    Note 'do' loop coding.
64                                                      //     in the loop
65          }
66          canu = canu / to;                           // calculate the average
67                                                      //     amount of cards
68          to = to / 1000;                             // calculate the probabilaty
69                                                      // output the result
70          printf("%4d%12.2lf%%%10.2lf\n", i, to, canu);
71      }
72      return 0;
73  }
```