lab08

```
$ gcc lab08.c
lab08.c:108:1: warning: non-void function does not return a value in all control pat
hs [-Wreturn-type]
}
^
lab08.c:125:1: warning: non-void function does not return a value in all control pat
hs [-Wreturn-type]
}
^
lab08.c:162:1: warning: non-void function does not return a value in all control pat
hs [-Wreturn-type]
}
^
3 warnings generated.
$ ./a.out
Categories      Probability
Straight flush   0.0014%
Four of a kind   0.0236%
Full house       0.1443%
Flush            0.1967%
Flush            0.1961%
Straight         0.3520%
Three of a kind  2.1138%
Two pair         4.7533%
One pair         42.254%
High card        50.161%
High card        50.162%

Program output is incorrect
CPU time: 2.78555 sec
score: 60
o. Compilation warnings.
o. [Output] Program output is incorrect
o. [Format] Program format can be improved
o. [Coding] lab08.c spelling errors: amout(1), coditions(1), housem(1)
```

lab08.c

```c
// EE2310 lab08 Poker Hands
// 109061217, 林峻霆
// Date: 2020/11/23

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

int Straight(int num[13]);                      // function detect straight
```

This line has more than 80 characters

```c
int Flush(int col[4]);                          // function detect flush
int Common(int num[13]);                        // function detect other
                                                // conditions
int main(void)
{
    long int N = 10000000;                      // amout of loop
    int card[4][13];                            // array to check repeat
    int num[13];                                // array for num
    int col[4];                                 // array for color
    int number, color, poker;                   // information of cards
    double category[9] = {0};                   // array for category
    int i, j, k;                                // parameter for loop
    int find;                                   // parameter for break
```

This line has more than 80 characters

```c

    while (N) {
        find = 1;
        for (i = 0; i < 13; i++)                // reset array
            num[i] = 0;
        for (i = 0; i < 4; i++)
            col[i] = 0;
        for (i = 0; i < 4; i++) {
            for (j = 0; j < 13; j++) {
                card[i][j] = 0;
            }
        }

        for (j = 0; j < 5; j++) {               // draw 5 card
            poker = (rand() / 1000) % 52;
            number = poker % 13;
            color = poker % 4;
            if (card[color][number] == 0) {     // if not repeat,
                card[color][number] = 1;        // put them into array
                num[number] += 1;
                col[color] += 1;
            }
            else                                // redraw
                j = j - 1;
        }

        i = Straight(num);                      // check straight
        j = Flush(col);                         // check flush
        if (i == 1 && j == 1)                   // set category
            category[0] += 1;
        else if (i == 1)
            category[4] += 1;
        else if (j == 1)
            category[3] += 1;
        else {
            k = Common(num);
            category[k] += 1;
        }
        N = N - 1;                              // next input
    }

    printf("Categories      Probability\n");    // print result
    printf("Straight flush   ");
    printf("%.4lf%%\n", category[0] / 100000);
    printf("Four of a kind   ");
    printf("%.4lf%%\n", category[1] / 100000);
    printf("Full house       ");
    printf("%.4lf%%\n", category[2] / 100000);
    printf("Flush            ");
    printf("%.4lf%%\n", category[3] / 100000);
    printf("Straight         ");
    printf("%.4lf%%\n", category[4] / 100000);
    printf("Three of a kind  ");
    printf("%.4lf%%\n", category[5] / 100000);
    printf("Two pair         ");
    printf("%.4lf%%\n", category[6] / 100000);
    printf("One pair         ");
    printf("%.3lf%%\n", category[7] / 100000);
    printf("High card        ");
    printf("%.3lf%%\n", category[8] / 100000);
    return 0;                                     // end  program
}

int Straight (int num[13])
int Straight(int num[13])
    // To check whether there's a straight
    //   if yes, return 1
    //   else return 0
{
    int find = 1;                                 // parameter for break
    int i = 0;                                    // parameter for loop
    int j;                                        // parameter for loop
    while (i <= 8) {
        find = 1;
        for (j = 4; j >= 0 && find; j--) {
            if (num[i + j] != 1) {                // check straight property
                find = 0;
                i = i + j + 1;                    // move i to next index
            }
        }
        if (find == 0 && i > 8)                   // if not find, return 0
            return 0;
        else if (find) {                          // if find return 1, end loop
            return 1;
            i = 9;
        }
    }
}

int Flush (int col[4])
int Flush(int col[4])
    // To check if there's a Flush
    //   if yes, return 1
    //   else return 0
{
    int find = 1;                                 // parameter for break
```

```
This line has more than 80 characters
    int i;                                       // parameter for loop
    for (i = 0; i < 4 && find; i++) {
        if (col[i] == 5) {                       // if exist flush return 1
            return 1;
            find = 0;                            // if find, change find to 0
        }
    }
    if (find)                                    // if not find, return 0
        return 0;
}

int Common (int num[13])
int Common(int num[13])
    // To check for other coditions except flush and straight
    //   if there's a Four of a kind, return 1
    //   if there's a Full housem return 2
    //   if there's a Three of a kind, return 5
    //   if there's a Two pair return 6
    //   if there's a One pair return 7
    //   if there's a High card return 8
{
    int three = 0;                               // amount of three same num
    int two = 0;                                 // amount of two same num
    int i;                                       // parameter for loop
    int find = 1;                                // parameter for break
    for (i = 0; i < 13 && find; i++) {
        if (num[i] == 4) {                       // if amount = 4, return 1
            return 1;
            find = 0;
        }
        else if (num[i] == 3)                    // if amount = 3, three += 1
            three += 1;
        else if (num[i] == 2)                    // if amount = 2, two += 1
            two += 1;
    }
    if (find) {
        if (three == 1 && two == 1)              // check for Full house
            return 2;
        else if (two == 2)                       // check for Two pair
            return 6;
        else if (three == 1)                     // check for Three of a kind
            return 5;
        else if (two == 1)                       // check for One pair
            return 7;
        else                                     // High card condition
            return 8;
    }
}
```