2020/09/09 - Dealing Cards (Part 1)

2020年9月7日　15:55

SYNOPSIS

- Go over LAB 2A.

LAB 2A

- On Canvas, go to the "Lab 2.1" assignment and read the "lab2.html" file attached. All lab details will be there.

- There are 2 parts. There are 2 due dates:
  - Sept 12 (SAT): Prog2a (BASIC CARD DEALING)
  - Sept 19 (SAT): Prog2b (DITTO w/ LINKED LIST)

- Prog2a has a given template code file you need to copy to start the lab.

SUBMISSION COMMANDS

LAB 2.1

```
tar -cvf lab2_1.tar Prog2a.cpp
```

LAB 2.2

```
tar -cvf lab2_2.tar Prog2b.cpp
```

Prog 2A

- Pay attention to what you are given:
  - "const face[]", array of 13 card names
  - "const suit[]", array of 4 card suits
  - Valid card format: "FACE of SUIT". So cards like "Ace of Spades" are generated.
  - "random_card", which generates a valid card in the format above.
  - A while loop which calls random_card and throws the generated card into a string.
  - Some comments telling you exactly what to do.

- Make a 2D ARRAY called "table". It should be "table[4][13]" but, if flipped, I don't care.
  - Set every value to 0. You cannot use "table[][] = {0};" to do it.
  - This is an ARRAY, NOT a VECTOR.

- Decode the generated card string to figure out the face and suit indices used to construct it.

(Ex. ACE OF SPADES ⟹ face[0] & suit[3]

Others:

CARD NAME	FACE	SUIT
ACE OF SPADES	0	3
6 OF DIAMONDS	5	1
QUEEN OF HEARTS	11	2
KING OF CLUBS	12	0

- SUGGESTED IMPLEMENTATION (OPTIONAL)
  - make a function called "decode_card".

    ```
    void decode_card(const string &c, int &f, int &s);
    ```

  - #include <sstream> to use istringstream to extract string data.

  - Use istringstream, set it to "c" (card text), and read into 3 string variables.

    c = "Ace of Spades"

    ↑ FACE ↑ JUNK ↑ SUIT

  - Loop through ALL faces in "face[]" and set "f" (face index) if there is a match.
  - Loop through ALL suits in "suit[]" and set "s" (suit index) if there is a match.
  - Return

- main function stays clean:

```
string card = random_card(verbose);
int s, f;
decode_card(card, f, s);
table[s][f]++;
```

- Check if KING, QUEEN, and JACK of a suit is one or more. If so, break out of the loop.

- Print the table. Match the solution.
  - #include <iomanip> for "setw".
  - "**" on the row where the break earlier occurred.
    - If s is still in scope, just use it. The suit stored in it is guaranteed to be the correct answer so you can skip a step.

Prog 2B

- Stay tuned!