COP3330 Programming Assignment 1 (Warm up)

Rank of Hands in Texas Holdem

Objective:

Using procedural programming constructs (arrays, loops, branches, subroutines, basic IO, etc) that you learnt in COP3014 for problem solving. This assignment is a warm up for things to come, and is easier than most of the rest of assignments in this course.

Details:

The program will read from the standard input (redirect to a file) a list of encoded hands of 5 cards, and identify the rank of the hands. The suits and ranks are encoded as follows.

Encoding of Suits:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Suit | Spade | Heart | Diamond | Club |
| Encoding | S | H | D | C |

Encoding of ranks:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rank | A | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K |
| Encoding | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

A card is represented by a letter (S, H, D, or C) followed by a number in the range 1-13. For example, S13 represents the King of Spade. Each hand consists of 5 cards. To simplifying programming, we will assume that the cards are sorted by encoded ranks. S13S12S11S10S1 is an example representing the hand with a Spade King, a Spade Queue, a Spade Jack, a Spade 10, and a spade Ace, which forms a royal flush.

The following list is from the best to worst rank of hands in Texas Holdem. The program would need to recognize all of the ranks.

**Royal Flush**: An Ace-high straight of one suit. Example: S13S12S11S10S1

    

**Straight Flush**: A straight of entirely one suit. Example: D8D7D6D5D4

    

**Four of a kind**: Four cards of the same rank. Example: S11H11D11C11D8

    

**Full house**: Three-of-a-kind and a pair. Example: D12S12D11S11C11

    

**Flush**: Five cards of the same suit. Example: D10D9D3D2D1

    

**Straight**: Five cards of sequential rank (Ace can be high or low). Example: C12H11S10D9D8

    

**Three of a kind**: Three cards of the same rank. Example: C13H13S13D2S1

    

**Two pair**: two two cards of the same rank. Example: S12D11C11S2D2

    

**One pair**: two cards of the same rank. Example C13S10D6S1D1

    

**High card**: None of above. Example: C13D10S7S4D2

    

Your program should read from the standard input (redirect to a file) a list of encoded hands of 5 cards (one hand on each line), identify the rank of the hands, print the name of the rank for each hand and the cards in the hand. The output for a given input file should be identical to that of the sample executable provided.

The following is a sample input (in file testcase0.txt):

S13S12S11S10S1

D5D4D3D2D1

C8C7C6C5C4

S11D11C11H11D8

S13J13S12D12H12

C13C4C10C2C5

C12H11S10D9D8

S5D4D3S2D1

C13H13S13D2S1

S12D11C11D2H2

C13S10D6S1D1

C13D10S7S4S2

The corresponding output of the program should be (see the execution of proj1\_linprog):

Loyal flush: S13S12S11S10S1

Straight flush: D5D4D3D2D1

Straight flush: C8C7C6C5C4

Four-of-a-kind: S11D11C11H11D8

Full house: S13J13S12D12H12

Flush: C13C4C10C2C5

Straight: C12H11S10D9D8

Straight: S5D4D3S2D1

Three-of-a-kind: C13H13S13D2S1

Two pair: S12D11C11D2H2

One pair: C13S10D6S1D1

High card: C13D10S7S4S2

**Submission**

The due time for this assignment is May 22 (Wendesday), 2013. 11:59pm.

Name your program as proj1.cpp (and proj1.h if you have a header file). Tar all of your files for this assignment (which should include at least two files proj1.cpp and bug\_fixing\_log.txt, your bug fixing log file), name the tar file yourlastname\_firstinitial\_proj1.tar and submit the tar file in blackboard. The bug fixing log file must follow the template given in the class website.

**Grading policy:**

The program must work on linprog. O point for programs with any g++ compiler error on linprog. You will need to track compiling errors that **you fixed by yourselves** in a log for fixed compiler bugs that needs to have at least two entries for 5 points/each entry (10 points max).

* Program with no compiler error (20 points)
* Log for fixed compiler bugs (10 points)
* Program able to display the input cards (20 points)
* Program able to identify all loyal flush correctly (6 points)
* Program is able to identify all straight flushes correctly (12 points)
* Program able to identify all of the rest ranks (4 points each rank, 32 points total)

**Hints:**

1. Start the project as soon as possible.
2. You can store each hand in two arrays, a character array for the suit and a character array for the rank.
3. You should write one routine for checking one rank (you will need to write ten such routines). Routine for checking whether the hand is a loyalflush can be as follows. Other routines are similar.

int isloyalflush(char suit[5], int rank[5]) {

 if (suit[0] != suit[1]) return 0;

 if (suit[0] != suit[2]) return 0;

 if (suit[0] != suit[3]) return 0;

 if (suit[0] != suit[4]) return 0;

 if (rank[0] != 13) return 0;

 if (rank[1] != 12) return 0;

 if (rank[2] != 11) return 0;

 if (rank[3] != 10) return 0;

 if (rank[4] != 1) return 0;

 return 1;

}

1. The main program structure would be as follows:

While (there are more inputs) {

 Read one line of hand into the suit and rank arrays

 If (isloyalflush(…)) output ‘loyal flush: ‘;

 else if (isstraightflush(…)) output ‘straight flush: ‘;

 else if ….

 …

 Output the hand read

 }