CS 2413 001: Data Structures, Summer 2000
Programming Project #2 Bonus: Counting Bridge Points
Worth up to 15% of the value of the regular project
Due in class Friday 7 July 2000

Note: no bonus sections will be accepted after class Friday 7 July

`http://www.cs.ou.edu/~cs2413/`

The game of bridge consists of two phases: bidding and play. This bonus section will be concerned with part of the bidding phase.

A game of (contract) bridge has four players, split into two teams. Often the teams are referred to as north-south and east-west. All of the cards are dealt, so each player starts with 13 cards.

During the bidding phase, players go around the table, in round-robin fashion, bidding higher and higher numbers of "tricks" that they expect their team will be able to win (each trick consisting of a card from each of the four players). For example, a player might open the bidding by saying "one heart," meaning that the bidder expects his or her team to take 7 tricks (the first 6 aren't counted in the bidding), using hearts as the *trump* suit (a special suit whose cards always beat all cards of other suits).

Since a player does not know their partner's cards during the bidding phase (and often will not know until all the tricks are played), bridge experts have developed an elaborate system of bidding codes that give a partner (and the opponents) useful information about the bidder's cards.

Players decide what to bid based on "point counting," which will be the subject of this bonus section. Point counting is divided into two categories: *high card points* and *distribution points*.

High card points encode the numbers of aces, kings, queens and jacks (the high cards, sometimes called *honor* cards) that a player holds, and are counted like so:

Card	Points
Ace	4
King	3
Queen	2
Jack	1

So, a bridge hand that contains two aces and a king but no queens or jacks would have

$$2 \cdot 4 + 1 \cdot 3 + 0 \cdot 2 + 0 \cdot 1 = 11$$

high card points.

Distribution points refer to the number of cards a player has in each suit, and are counted like so:

Name	Meaning	Points
Doubleton	Only two cards in a particular suit	1
Singleton	Only one card in a particular suit	2
Void	No cards in a particular suit	3

So, a bridge hand that contains two clubs, three diamonds, one heart and seven spades would have $1 + 0 + 2 + 0 = 3$ distribution points.

However, some bridge players have adopted a modification to this system, encoding the concept of *protection*: a high card is not counted for high card points if it is not *protected* by a sufficient number of other cards in the same suit. The number of cards required to protect each high card is:

Card	Number of other cards of the same suit needed for protection
Ace	0
King	1
Queen	2
Jack	3

Thus, for example, if a hand has queen, jack and 10 of hearts, then the queen is protected (by the jack and the 10) but the jack is not.

Under this modified counting system, if a high card is not protected, then its high card points are not counted.

Seeding the Random Number Generator

In this project, you will use the random number generator provided as part of the standard C++ library, `random`.

Random number generators actually produce deterministic sequences of numbers; that is, if the most recently generated random value was *x*, then the random number generator will **always** generate *y* as its next random number. Therefore, random number generators are often referred to as *pseudo*-random.

Most random number generators can be *seeded* with a value that, for our purposes, we can think of as imposing a most-recently-generated value on the pseudo-random sequence; that is, it skips the RNG ahead to some new place in the sequence.

In the case of `random`, for example, the standard C++ library also includes a function called `srandom` that takes an `unsigned int` argument as a seed.

The Bonus Task

Modify the program described in Programming Project #2 in the following ways:

1. It should input and apply a seed for the random number generator.
2. If the game is named bridge (case insensitive; e.g., "Bridge" and "BRIDGE" should also work), then the program should output the players' points, like so:
   `Henry:  5C 7C TD JD AD 2H 4H 6H 8H QH JS QS KS (12 points)`

Run your program using the inputs in the Project #2 description, as well as several extra bridge games (using the same player names), using each of the following seeds:

1234567890, 987654321, 135792468, 2468013579, 2486248624

You should use the same player names as listed in the bridge example in the Programming Project #2 description.

You may turn in your project (with bonus section) early; otherwise, turn it in during class on Friday 7 July. If you turn it in after the close of class (3:20pm), it will be considered late, at which point not only will you lose 10% of the maximum value, you will also receive absolutely no credit for the bonus section. Submissions after 3:20pm Monday 10 July will receive no credit at all.

References

S. Radhakrishnan, L. Wise & C. N. Sekharan, *Object-Oriented Data Structures Featuring C++*, 1999.
`http://www.bridgeworld.com/beginners/bbeg5.html`