SPRING 2019

ECE 103 ENGINEERING PROGRAMMING

EXPANDED HOMEWORK HW-6

ECE DEPARTMENT
PORTLAND STATE UNIVERSITY

ECE 103 Expanded Homework HW-6

Problem List

60 points

Finding combinations of twenty-one in a deck of cards

General Instructions

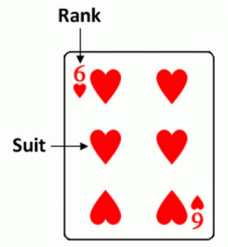
- Put a title block at the top of each source code file. It should include the course number, homework number, your full name, and a short program description.
- What you need to submit for grading:
  - Program Design Report (format: Word .doc/.docx or pdf)
  - Source code with this file name: **`hw6.c`**
- Store your completed files (source code, reports, etc.) in a single ZIP archive file.
  - Do **not** include the executable file (e.g., a.out or a.exe) or IDE project files.
  - WARNING: If you do not zip your files, you will receive a point deduction.
- Upload your ZIP file to the appropriate D2L Submission Folder by the deadline.

NOTE

For an example of what a typical Program Design Report should look like, refer to the file "Sample_Program_Design_Report.pdf", which is stored on D2L in the *Homework:Guide* module.

Problem Statement

A standard deck of playing cards contains 52 cards. There are four suits (heart ♥, diamond ♦, club ♣, spade ♠) and thirteen ranks (ace, 2 thru 10, jack, queen, king) per suit.



Rank	Numeric Value
ace (A)	1 or 11
2	2
3	3
⋮	⋮
9	9
10	10
jack (J)	10
queen (Q)	10
king (K)	10

Write a program that does the following:

1. Simulates a randomized shuffling of a 52 card deck.
2. Determines how many combinations of 21 exist in the deck by following this procedure:
    - Consecutively draws cards from the shuffled deck until either the combined value is 21 (win) or the combined value is over 21 (bust).
    - Continues drawing / displaying / combining cards until all the cards in the deck are used up.
2. Displays the total number of "wins" and "busts" for the given deck.

How should you handle an ace?

Since an ace card can be either 1 or 11 in value, you must decide which value to use in order to maximize the chance of getting a twenty-one combination.

For instance, suppose the order of the first few cards in the deck is:

AH AS 10C AC 8S ...

Choice	AH	AS	10C	AC	8S	Sum	Result
#1	1	11	10	-	-	**22**	bust
#2	11	1	10	-	-	**22**	bust
#3	11	11	-	-	-	**22**	bust
#4	1	1	10	11	-	**23**	bust
#5	1	1	10	1	8	**21**	win

Clearly, four of the possible choices for assigning a value to the ace lead to a bust situation. Only one option produces a winning combination. Your algorithm should incorporate logic to avoid the obviously poor choices. Of course, even if you choose the ace values wisely, sometimes you may still bust, as would be the case if the 8S was a 9S instead.

Hint: There are two situations to consider. Either you make all the aces have a value of 1, or else you assign one ace to be 11 and all other aces to 1.

As another example, let this be the order of the first few cards in a newly shuffled deck:

6H 3D KH 8C AC QH AS 9D 8H 5S 4H 5H 8D 2C 2S...

Draw	Value	Running Sum	Result
6H	6	6	
3D	3	9	
KH	10	19	
8C	8	27	bust
AC	1 or 11	1 or 11	
QH	10	**21**	win
AS	1 or 11	1 or 11	
9D	9	10 or 20	
8H	8	18	
5S	5	23	bust
4H	4	4	
5H	5	9	
8D	8	17	
2C	2	19	
2S	2	21	win
⋮	⋮	⋮	

Requirements

The program should:

1. Prompt the user to enter a seed value for a random number generator.
2. Display the contents of the card deck both before shuffling and after shuffling.
    - Arrange the deck of cards in four columns with thirteen cards per column.
    - For each card, show the rank first, followed by the suit.
        - Display the rank as '2' through '10' for card values from 2 to 10, 'A' for ace, 'J' for jack, 'Q' for queen, or 'K' for king.
        - Display the suit as 'H' for heart, 'D' for diamond, 'C' for club, or 'S' for spade.
    - Before shuffling, the deck should be in sorted order.
3. Display the combinations like this:
    - On the same line, show the sequence of cards that are drawn from the deck.
    - Once you know the final state of the combination:
        - Display "BUST" if over 21.
        - Display "WIN" if exactly 21.
        - Display "OUT OF CARDS" if you run out of cards before knowing whether it is bust or win.
    - Show each new draw sequence on a separate line.
4. Implement the card deck as a 1-D array (size 52) of structures.
    - A single structure represents a single card in the deck. The structure's member variables should include the card rank, the card suit, and the card value.
      Note: The structure does **not** contain the entire deck of cards!
    - If you want more of a challenge, you could implement the deck as an array of pointers to structures using dynamic memory allocation, but this is not required.
5. Implement a shuffling algorithm that uses the C library's random number generator function.
    - A common shuffling technique is to swap the first card with a card at a random location, then swap the second card with a card at another random location, and so on until all of the cards in the deck have been swapped at least once.
    - Look up the `srand()` library function for details on how to use the seed to initialize the random number generator. Note that if you use the same seed value each time, the generated sequence will be the same. This is useful for testing purposes.
    - **The shuffling procedure should be implemented as a separate C function.**
      If there are other areas in your program that would make sense to be placed in its own function, feel free to do so.
6. Implement at least one example of an enumerated type (e.g., to define the suit values).

Save your program using this filename: **`hw6.c`**

Sample Runs

```
Enter a seed value: 1234

Before shuffling:
 A H      A D      A C      A S
 2 H      2 D      2 C      2 S
 3 H      3 D      3 C      3 S
 4 H      4 D      4 C      4 S
 5 H      5 D      5 C      5 S
 6 H      6 D      6 C      6 S
 7 H      7 D      7 C      7 S
 8 H      8 D      8 C      8 S
 9 H      9 D      9 C      9 S
10 H     10 D     10 C     10 S
 J H      J D      J C      J S
 Q H      Q D      Q C      Q S
 K H      K D      K C      K S

After shuffling:
 8 H      9 H      8 D     10 H
 K H      7 D      Q H      2 S
 6 D      5 S      J H     10 S
 2 D      2 C      9 C      Q C
 J C     10 C      A D     10 D
 3 S      K D      K C      4 C
 8 C      4 H      K S      5 C
 6 H      3 C      3 D      A S
 5 H      Q S      4 D      4 S
 9 D      Q D      5 D      7 S
 7 C      J S      8 S      7 H
 3 H      A C      2 H      6 C
 A H      J D      9 S      6 S

Find combinations of 21:
8H KH 6D  BUST
2D JC 3S 8C  BUST
6H 5H 9D 7C  BUST
3H AH 9H 7D 5S  BUST
2C 10C KD  BUST
4H 3C QS QD  BUST
JS AC  WIN
JD 8D QH  BUST
JH 9C AD KC  BUST
KS 3D 4D 5D  BUST
8S 2H 9S 10H  BUST
2S 10S QC  BUST
10D 4C 5C AS 4S  BUST
7S 7H 6C 6S  BUST
OUT OF CARDS

Number of wins  = 1
Number of busts = 13
```

Just for fun, the instructor modified the test program to display graphical symbols for each suite by using the old IBM extended character set. You are NOT required to do this:

```
Enter a seed value: 5456

Before shuffling:
 A ♥      A ♦      A ♣      A ♠
 2 ♥      2 ♦      2 ♣      2 ♠
 3 ♥      3 ♦      3 ♣      3 ♠
 4 ♥      4 ♦      4 ♣      4 ♠
 5 ♥      5 ♦      5 ♣      5 ♠
 6 ♥      6 ♦      6 ♣      6 ♠
 7 ♥      7 ♦      7 ♣      7 ♠
 8 ♥      8 ♦      8 ♣      8 ♠
 9 ♥      9 ♦      9 ♣      9 ♠
10 ♥     10 ♦     10 ♣     10 ♠
 J ♥      J ♦      J ♣      J ♠
 Q ♥      Q ♦      Q ♣      Q ♠
 K ♥      K ♦      K ♣      K ♠

After shuffling:
 3 ♣      7 ♥      5 ♠      3 ♦
 3 ♥      4 ♥      J ♠      4 ♦
 K ♥      6 ♦      2 ♦      K ♣
 Q ♣      8 ♥      Q ♥      K ♠
 J ♦     10 ♣      3 ♠     10 ♦
10 ♥      2 ♣     10 ♠      2 ♠
 A ♣      6 ♥      6 ♠      7 ♠
 5 ♦      9 ♦      9 ♣      4 ♣
 8 ♣      7 ♣      2 ♥      J ♣
 A ♥      5 ♥      5 ♣      7 ♦
 J ♥      A ♠      Q ♠      9 ♠
 9 ♥      8 ♠      6 ♣      Q ♦
 A ♦      8 ♦      4 ♠      K ♦

Find combinations of 21:
3♣ 3♥ K♥ Q♣  BUST
J♦ 10♥ A♣  WIN
5♦ 8♣ A♥ J♥  BUST
9♥ A♦ 7♥ 4♥  WIN
6♦ 8♥ 10♣  BUST
2♣ 6♥ 9♦ 7♣  BUST
5♥ A♠ 8♠ 8♦  BUST
5♠ J♠ 2♦ Q♥  BUST
3♠ 10♠ 6♠ 9♣  BUST
2♥ 5♣ Q♠ 6♣  BUST
4♠ 3♦ 4♦ K♣  WIN
K♠ 10♦ 2♠  BUST
7♠ 4♣ J♣  WIN
7♦ 9♠ Q♦  BUST
K♦ OUT OF CARDS

Number of wins  = 4
Number of busts = 10
```