

Introduction to C - Programming Assignment #6

Due Date: Monday, December 7, 2009, 11:59pm

Objective

1. To give students practice writing functions.
2. To give students practice using structs

Problem: Arup's Solitaire

Though Arup is rarely early, he still wants a contingency plan just in case he ever makes it early to any sort of appointment, because he really hates waiting. Playing cards are easy to carry around, so his plan is that he will always have his cards with him so that he can play a game if he has to wait. But, regular solitaire is getting boring, so he'd like you to implement a new card game that can be played with one person.

The idea is very simple:

Initially, the player is dealt 4 random cards from the full deck of regular playing cards. From this point on, the player can choose to exchange any single card she holds with a randomly chosen card from the deck.

This process continues until the player has four of a kind.

The player's score is simply the number of times she had to exchange cards to complete the task.

You are to write a program that simulates this game and tells the player their final score.

Your program should also allow a user to play as many times as she wants, and tally her best score. (At the end of the program, when the user no longer plays, prompt her with her best score of all the games she played.)

Implementation Requirements

At least two structs must be declared:

- 1) To hold a single playing card
- 2) To hold a deck of playing cards.

It is recommended (but not required) to have a third struct that holds a hand of playing cards. Two alternatives to this third struct are:

- 1) Using an array of the card struct of size 4 to hold a hand.
- 2) Using the same struct that you use for a deck for a hand of cards.

References

Textbook: Chapters 8, 9, 16

Notes: Lectures on functions, arrays, structs

Restrictions

Name the file you create and turn in ***solitaire.c***. Although you may use other compilers, your program must compile and run using Dev C++. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. You should also include comments throughout your code, when appropriate. If you have any questions about this, please see a TA.

Deliverables

A single source file named ***solitaire.c***. turned in through WebCourses. **Remember: NO LATE ASSIGNMENTS!!!**

Sample Output

Note: The actual output for this game is insanely long. I will just show the beginning and ending of a game so you can see what format I would like you to use. In this sample, two games will be played. (User input is in bold and italics.)

Here is your hand:

- 0. 5 of S
- 1. 8 of H
- 2. 8 of S
- 3. 2 of S

Which card would you like to exchange(0,1,2,3)?

3

Here is your hand:

- 0. 5 of S
- 1. 8 of H
- 2. 8 of S
- 3. K of D

Which card would you like to exchange(0,1,2,3)?

0

Here is your hand:

- 0. T of H
- 1. 8 of H
- 2. 8 of S
- 3. K of D

Which card would you like to exchange(0,1,2,3)?

Note: Now we skip many lines of output!

Here is your hand:

- 0. 8 of C
- 1. 8 of H
- 2. 8 of S
- 3. K of D

Which card would you like to exchange(0,1,2,3)?

3

Congrats, you finished with 71 turns!

Here is your final hand:

- 0. 8 of C
- 1. 8 of H
- 2. 8 of S
- 3. 8 of D

Would you like to play again?(yes/no)

yes

Here is your hand:

- 0. 3 of H
- 1. 2 of C
- 2. 7 of D
- 3. 7 of S

Which card would you like to exchange(0,1,2,3)?

Note: Now we skip many lines of output!

Here is your hand:

- 0. K of D
- 1. 7 of C
- 2. 7 of D
- 3. 7 of S

Which card would you like to exchange(0,1,2,3)?

0

Congrats, you finished with 52 turns!

Here is your final hand:

- 0. 7 of H
- 1. 7 of C
- 2. 7 of D
- 3. 7 of S

Would you like to play again?(yes/no)

no

Thanks for playing. Your best score was 52.

Extra Credit

Write a program that reads in the total number of seconds since midnight, and then breaks the time into hours (0-23), minutes (0-59), and seconds (0-59).

To implement this program, create the following function:

```
struct time split_time(long total_seconds);
```

An example output would be

Enter Time in Seconds: 1203

Hours: 0 Minutes: 20 Seconds: 3

Turn in this program as `split_time.c` and add it to Webcourses in addition to `solitaire.c`.