Code Report

This report will analyze the provided C code and provide detailed explanations of each part.

Part 1: Calculating Optimal Paths

In the first section, a function is implemented to calculate the shortest path on a two-dimensional grid. Three functions are defined here:

1. **find_factor(int n)**:
    - This function calculates the factorial of a given number **n**. It uses a recursive approach.
    - For example, for **n = 5**, it calculates 5!.
2. **numPathsHome(int street, int avenue, int fact, int street2, int avenue2)**:
    - This function calculates the shortest path on the grid from a given point **(street, avenue)** to the corner point **(0,0)**.
    - If both **street** and **avenue** are zero, it uses the combination formula to calculate the path: **fact / (street! * avenue!)**.
    - If **street** is not zero, it moves one step left and recursively calls **numPathsHome**.
    - If **avenue** is not zero, it moves one step down and recursively calls **numPathsHome**.
3. **main(int argc, const char * argv[])**:
    - It takes street and avenue numbers as input from the user.
    - It calls the **numPathsHome** function and prints the result.

Part 2: Hospital and City Relationships

In the second section, a structure (**struct**) is defined:

1. **Hospital**:
    - **name**: The name of the hospital.
    - **citiesServed**: The cities served by the hospital.

This section includes arrays **cities** and **locations** to represent which cities are served by which hospitals. However, the **Hospital** struct is defined but not utilized further in the code.

Part 3: Card Deck

In the third section, a deck of cards is created, shuffled, and printed. Three functions are defined here:

1. **Card**:
    - **face**: The face value of the card (e.g., "Ace", "King").
    - **suit**: The suit of the card (e.g., "Hearts", "Diamonds").
2. **init_cards(Card deck[52])**:
    - This function initializes a deck of 52 cards and assigns face and suit values to each card.
    - There are four suits and thirteen face values.
3. **shuffle_deck(Card deck[52])**:
    - This function shuffles the deck of cards.
    - It randomly selects an index and swaps cards.
4. **printDeck(Card deck[52])**:
    - This function prints the deck of cards in order.
5. **main(int argc, const char * argv[])**:

- It initializes a deck of cards, shuffles it, and prints the deck before and after shuffling.

some outputs are available below:

```
Enter the street number2
Enter the avenue number3
number of optimal paths to take back home 3
```

```
  Ace of Hearts
Deuce of Hearts
Three of Hearts
 Four of Hearts
 Five of Hearts
  Six of Hearts
Seven of Hearts
Eight of Hearts
 Nine of Hearts
  Ten of Hearts
 Jack of Hearts
Queen of Hearts
 King of Hearts
  Ace of Diamonds
Deuce of Diamonds
Three of Diamonds
 Four of Diamonds
 Five of Diamonds
  Six of Diamonds
Seven of Diamonds
Eight of Diamonds
 Nine of Diamonds
  Ten of Diamonds
 Jack of Diamonds
Queen of Diamonds
 King of Diamonds
  Ace of Clubs
Deuce of Clubs
Three of Clubs
 Four of Clubs
 Five of Clubs
  Six of Clubs
Seven of Clubs
Eight of Clubs
 Nine of Clubs
  Ten of Clubs
 Jack of Clubs
Queen of Clubs
 King of Clubs
  Ace of Spades
Deuce of Spades
Three of Spades
 Four of Spades
 Five of Spades
  Six of Spades
Seven of Spades
Eight of Spades
 Nine of Spades
  Ten of Spades
 Jack of Spades
Queen of Spades
 King of Spades
```

```
-----shufffffliiinggg----
 Four of Hearts
 Five of Diamonds
  Ten of Diamonds
 Four of Clubs
Three of Diamonds
 Nine of Hearts
Eight of Spades
Eight of Diamonds
 King of Diamonds
 King of Hearts
Deuce of Hearts
  Six of Hearts
 Nine of Clubs
 King of Spades
Deuce of Diamonds
  Ace of Hearts
Eight of Hearts
Seven of Diamonds
 King of Clubs
Eight of Clubs
Seven of Spades
Seven of Clubs
 Four of Diamonds
 Jack of Clubs
Deuce of Clubs
  Six of Spades
Three of Hearts
 Nine of Diamonds
  Ace of Diamonds
  Six of Clubs
Queen of Clubs
Queen of Spades
 Jack of Diamonds
 Five of Clubs
  Six of Diamonds
 Four of Spades
 Five of Spades
 Jack of Hearts
Queen of Hearts
 Jack of Spades
Three of Spades
Seven of Hearts
  Ten of Clubs
  Ten of Spades
Queen of Diamonds
  Ace of Clubs
Three of Clubs
 Nine of Spades
  Ten of Hearts
 Five of Hearts
Deuce of Spades
  Ace of Spades
```