Faro Shuffle Sim

2

Generated by Doxygen 1.8.14

Contents

Chapter 1

Faro_Shuffle_Sim

This program implements the faro shuffle magicians trick as a linked list.

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Card Struct Reference

this struct represents a single card in a deck of cards.

`#include <headers.h>`

Public Attributes

- char * value

  Member value stores the rank and suit ofthe vard being read in.
- unsigned int number

  Member 'number' stores the number if the NUMERICAL option is chosen.
- Card * next

  Member 'next' stores a pointer to the next card in a linked list of Cards.

4.1.1 Detailed Description

this struct represents a single card in a deck of cards.

4.1.2 Member Data Documentation

4.1.2.1 next

`Card::next`

Member 'next' stores a pointer to the next card in a linked list of Cards.

4.1.2.2 number

`Card::number`

Member 'number' stores the number if the NUMERICAL option is chosen.

4.1.2.3 value

`Card::value`

Member value stores the rank and suit ofthe vard being read in.

The documentation for this struct was generated from the following file:

- headers.h

Chapter 5

File Documentation

5.1 faro_ll.c File Reference

```
#include "headers.h"
#include "files/print_faro_val.h"
```

Functions

- Card * get_cards_ranksuit (int size)

  This function reads in a number of rank + suit values = to 'size' and returns the head of a linked list of the deck of cards.
- Card * shuffle (Card *top_ptr, Card *bot_ptr, int size)

  This function interweaves two equal length linked lists.
- Card * split (Card *deck, int size)

  This function iterates through a linked list and splits it at the halfway point.
- void print_deck (Card *current_node, int size, Stringplace prefix)

  This function prints the deck of cards when its a linked list.
- void free_ll (Card *current_node, int size, int k)

  This function takes the head node of the linked list of cards and iteratively frees all the allocated memory from the list.
- int getIntLength (int n)

  This function calculates the length of a decimal number.
- void faro (Card **deck, int size, int *k_binary, int k_length, int k)

  This function handles the faro shuffle by calling the other functions.
- void print_deck_numerical (Card *current_node, int size, Stringplace prefix)

  This function prints the deck of cards with numerical values.
- Card * get_cards_numerical (int size)

  This function reads in a number of numeric values = to 'size' and returns the head of a linked list of the deck of cards.
- void faro_numerical (Card **deck, int size, int *k_binary, int k_length)

  This function handles the faro shuffle by calling the other functions.

5.1.1 Function Documentation

5.1.1.1 faro()

```
void faro (
            Card ** deck,
            int size,
            int * k_binary,
            int k_length,
            int k )
```

This function handles the faro shuffle by calling the other functions.

Parameters

deck	is the deck of cards to be shuffled
size	is the length of the deck
k_binary	is the binary representation of the position k specified by the user.
k_length	is the length of the binary k

Returns

void

5.1.1.2 faro_numerical()

```
void faro_numerical (
            Card ** deck,
            int size,
            int * k_binary,
            int k_length )
```

This function handles the faro shuffle by calling the other functions.

Used for when the deck has numerical values.

Parameters

deck	is the deck of cards to be shuffled
size	is the length of the deck
k_binary	is the binary representation of the position k specified by the user.
k_length	is the length of the binary k

Returns

void

5.1.1.3 free_ll()

```
void free_ll (
            Card * current_node,
            int size,
            int k )
```

This function takes the head node of the linked list of cards and iteratively frees all the allocated memory from the list.

Parameters

current_node	the head node of the linked list
size	the length of the linked list

5.1.1.4 get_cards_numerical()

```
Card* get_cards_numerical (
            int size )
```

This function reads in a number of numeric values = to 'size' and returns the head of a linked list of the deck of cards.

Parameters

size	the number of values being read in.

5.1.1.5 get_cards_ranksuit()

```
Card* get_cards_ranksuit (
            int size )
```

This function reads in a number of rank + suit values = to 'size' and returns the head of a linked list of the deck of cards.

Parameters

size	the number of values being read in.

5.1.1.6 getIntLength()

```
int getIntLength (
            int n )
```

This function calculates the length of a decimal number.

It is used to calculate how much memory needs to be mallocd to contain a string for that number

Parameters

n	is the number being measured

5.1.1.7 print_deck()

```
void print_deck (
            Card * current_node,
            int size,
            Stringplace prefix )
```

This function prints the deck of cards when its a linked list.

Parameters

deck	the linked list of Cards being printed
size	the length of the linked list 'deck'
prefix	whether an IN shuffle or OUT shuffle was performed.

5.1.1.8 print_deck_numerical()

```
void print_deck_numerical (
            Card * current_node,
            int size,
            Stringplace prefix )
```

This function prints the deck of cards with numerical values.

Parameters

deck	the linked list of Cards being printed
size	the length of the linked list 'deck'
prefix	whether an IN shuffle or OUT shuffle was performed.

5.1.1.9 shuffle()

```
Card* shuffle (
            Card * top_ptr,
```

```
            Card * bot_ptr,
            int size )
```

This function interweaves two equal length linked lists.

Taking one node from top then from bottom, repeated until the end of the linked lists.

Parameters

top	the elements of this linked list will make up the 1st, 3rd etc. elements of the list being returned.
bot	the the elements of this linked list will make up the 2nd, 4th etc. elements of the list being returned.

Returns

new_deck the linked list made by interweaving the top and bot linked lists.

5.1.1.10 split()

```
Card* split (
            Card * deck,
            int size )
```

This function iterates through a linked list and splits it at the halfway point.

then it returns the bottom half of the list.

Parameters

deck	the linked list representing the deck of cards. Half of the deck is split into a new linked list.
size	the size of the deck of cards.

Returns

bottom_half the bottom half of the 'deck' linked list.

5.2 faro_shuffle.c File Reference

```
#include "headers.h"
#include <assert.h>
```

Functions

- int * decToBinary (int number, int *length)

  This function converts a decimal number to its binary representation.
- int main (int argc, char *argv[ ])

  This is the main controlling function of the faro shuffle.

5.2.1 Function Documentation

5.2.1.1 decToBinary()

```
int* decToBinary (
            int number,
            int * length )
```

This function converts a decimal number to its binary representation.

Parameters

number	the number "k" that is to be converted
∗length	an integer point that will contain the length of the binary number

Returns

the binary number stored in a integer array in reverse format (i.e. most significant bit stored last and least significant stored first)

5.2.1.2 main()

```
int main (
            int argc,
            char * argv[] )
```

This is the main controlling function of the faro shuffle.

Parameters

argc	should be a minimum of 2
argv[]	should contain either the option RANKSUIT or NUMERICAL

5.3 getBinary.c File Reference

```
#include <math.h>
#include <stdlib.h>
```

Functions

- int ∗ decToBinary (int number, int ∗length)

  This function converts a decimal number to its binary representation.

5.3.1 Function Documentation

5.3.1.1 decToBinary()

```
int* decToBinary (
            int number,
            int * length )
```

This function converts a decimal number to its binary representation.

Parameters

number	the number "k" that is to be converted
∗length	an integer point that will contain the length of the binary number

Returns

the binary number stored in a integer array in reverse format (i.e. most significant bit stored last and least significant stored first)

5.4 headers.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Classes

- struct Card

  this struct represents a single card in a deck of cards.

Typedefs

- typedef struct Card Card

Functions

- Card ∗ get_cards_ranksuit (int size)

  This function reads in a number of rank + suit values = to 'size' and returns the head of a linked list of the deck of cards.
- void faro (Card ∗∗deck, int size, int ∗k_binary, int k_length, int k)

  This function handles the faro shuffle by calling the other functions.
- Card ∗ get_cards_numerical (int size)

  This function reads in a number of numeric values = to 'size' and returns the head of a linked list of the deck of cards.
- void faro_numerical (Card ∗∗deck, int size, int ∗k_binary, int k_length)

  This function handles the faro shuffle by calling the other functions.

5.4.1 Typedef Documentation

5.4.1.1 Card

```
typedef struct Card Card
```

5.4.2 Function Documentation

5.4.2.1 faro()

```
void faro (
            Card ** deck,
            int size,
            int * k_binary,
            int k_length,
            int k )
```

This function handles the faro shuffle by calling the other functions.

Parameters

deck	is the deck of cards to be shuffled
size	is the length of the deck
k_binary	is the binary representation of the position k specified by the user.
k_length	is the length of the binary k

Returns

void

5.4.2.2 faro_numerical()

```
void faro_numerical (
            Card ** deck,
            int size,
            int * k_binary,
            int k_length )
```

This function handles the faro shuffle by calling the other functions.

Used for when the deck has numerical values.

Parameters

deck	is the deck of cards to be shuffled
size	is the length of the deck
k_binary	is the binary representation of the position k specified by the user.
k_length	is the length of the binary k

Returns

void

5.4.2.3 get_cards_numerical()

```
Card* get_cards_numerical (
            int size )
```

This function reads in a number of numeric values = to 'size' and returns the head of a linked list of the deck of cards.

Parameters

size	the number of values being read in.

5.4.2.4 get_cards_ranksuit()

```
Card* get_cards_ranksuit (
            int size )
```

This function reads in a number of rank + suit values = to 'size' and returns the head of a linked list of the deck of cards.

Parameters

size	the number of values being read in.

5.5 README.md File Reference

Index