

## Faro Shuffle Sim

2

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Faro_Shuffle_Sim</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	Card Struct Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Member Data Documentation . . . . .	7
4.1.2.1	next . . . . .	7
4.1.2.2	number . . . . .	8
4.1.2.3	value . . . . .	8
<b>5</b>	<b>File Documentation</b>	<b>9</b>
5.1	faro_ll.c File Reference . . . . .	9
5.1.1	Function Documentation . . . . .	9
5.1.1.1	faro() . . . . .	10
5.1.1.2	faro_numerical() . . . . .	10
5.1.1.3	free_ll() . . . . .	11
5.1.1.4	get_cards_numerical() . . . . .	11
5.1.1.5	get_cards_ranksuit() . . . . .	11
5.1.1.6	getIntLength() . . . . .	11

5.1.1.7	<code>print_deck()</code>	12
5.1.1.8	<code>print_deck_numerical()</code>	12
5.1.1.9	<code>shuffle()</code>	12
5.1.1.10	<code>split()</code>	13
5.2	<code>faro_shuffle.c</code> File Reference	13
5.2.1	Function Documentation	14
5.2.1.1	<code>decToBinary()</code>	14
5.2.1.2	<code>main()</code>	14
5.3	<code>getBinary.c</code> File Reference	14
5.3.1	Function Documentation	15
5.3.1.1	<code>decToBinary()</code>	15
5.4	<code>headers.h</code> File Reference	15
5.4.1	Typedef Documentation	16
5.4.1.1	<code>Card</code>	16
5.4.2	Function Documentation	16
5.4.2.1	<code>faro()</code>	16
5.4.2.2	<code>faro_numerical()</code>	16
5.4.2.3	<code>get_cards_numerical()</code>	17
5.4.2.4	<code>get_cards_ranksuit()</code>	17
5.5	<code>README.md</code> File Reference	17

## 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:

<a href="#">Card</a>	This struct represents a single card in a deck of cards . . . . .	<a href="#">7</a>
----------------------	-------------------------------------------------------------------	-------------------





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">faro_ll.c</a>	9
<a href="#">faro_shuffle.c</a>	13
<a href="#">getBinary.c</a>	14
<a href="#">headers.h</a>	15



## 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 of the card 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 of the card 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

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

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

<i>current_node</i>	the head node of the linked list
<i>size</i>	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

<i>size</i>	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

<i>size</i>	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 malloced to contain a string for that number

#### Parameters

<i>n</i>	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

<i>deck</i>	the linked list of Cards being printed
<i>size</i>	the length of the linked list 'deck'
<i>prefix</i>	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

<i>deck</i>	the linked list of Cards being printed
<i>size</i>	the length of the linked list 'deck'
<i>prefix</i>	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

<i>top</i>	the elements of this linked list will make up the 1st, 3rd etc. elements of the list being returned.
<i>bot</i>	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

<i>deck</i>	the linked list representing the deck of cards. Half of the deck is split into a new linked list.
<i>size</i>	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

<i>number</i>	the number "k" that is to be converted
<i>*length</i>	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

<i>argc</i>	should be a minimum of 2
<i>argv[]</i>	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

<i>number</i>	the number "k" that is to be converted
<i>*length</i>	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

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

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

<i>size</i>	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**

<i>size</i>	the number of values being read in.
-------------	-------------------------------------

**5.5 README.md File Reference**



# Index

- Card, [7](#)
  - headers.h, [16](#)
  - next, [7](#)
  - number, [7](#)
  - value, [8](#)
- decToBinary
  - faro\_shuffle.c, [14](#)
  - getBinary.c, [15](#)
- faro
  - faro\_ll.c, [9](#)
  - headers.h, [16](#)
- faro\_ll.c, [9](#)
  - faro, [9](#)
  - faro\_numerical, [10](#)
  - free\_ll, [10](#)
  - get\_cards\_numerical, [11](#)
  - get\_cards\_ranksuit, [11](#)
  - getIntLength, [11](#)
  - print\_deck, [12](#)
  - print\_deck\_numerical, [12](#)
  - shuffle, [12](#)
  - split, [13](#)
- faro\_numerical
  - faro\_ll.c, [10](#)
  - headers.h, [16](#)
- faro\_shuffle.c, [13](#)
  - decToBinary, [14](#)
  - main, [14](#)
- free\_ll
  - faro\_ll.c, [10](#)
- get\_cards\_numerical
  - faro\_ll.c, [11](#)
  - headers.h, [17](#)
- get\_cards\_ranksuit
  - faro\_ll.c, [11](#)
  - headers.h, [17](#)
- getBinary.c, [14](#)
  - decToBinary, [15](#)
- getIntLength
  - faro\_ll.c, [11](#)
- headers.h, [15](#)
  - Card, [16](#)
  - faro, [16](#)
  - faro\_numerical, [16](#)
  - get\_cards\_numerical, [17](#)
  - get\_cards\_ranksuit, [17](#)
- main
  - faro\_shuffle.c, [14](#)
- next
  - Card, [7](#)
- number
  - Card, [7](#)
- print\_deck
  - faro\_ll.c, [12](#)
- print\_deck\_numerical
  - faro\_ll.c, [12](#)
- README.md, [17](#)
- shuffle
  - faro\_ll.c, [12](#)
- split
  - faro\_ll.c, [13](#)
- value
  - Card, [8](#)