https://pages.mini.pw.edu.pl/~aszklarp/p1/task10?lang=en

4 min read

mgr inż. Paweł Aszklar

Programming 1 - Laboratories: Exercise Task 6

Before you start, download main.c file available at the bottom of the page and copy its content into the source file where you will be solving this task. For each part of the task uncomment given code and fill missing code.

In this task you will be implementing operations on a deck of playing cards represented as a doubly-linked list. You are given three data structures:

- **Card** - a single card with its suit and rank,
- **DeckElem** - linked list element storing a card,
- **Deck** - linked list representing a deck of cards.

Part 1

Write the following functions:

- DeckElem * deck_create_elem(Card c);

  Allocates a new DeckElem and initializes it with a given Card. Returns a pointer to the element or NULL if allocation failed.

- void deck_append_elem(Deck *d, DeckElem *el);

  Adds a card to the end of the Deck.

- void deck_print(const Deck *d);

Prints the contents of the Deck to the standard output. You can use card_print function to print individual Cards.

- void deck_destroy(Deck *d);

  Clears the Deck of all cards.

Part 2 (1.5 points)

Write a function:

DeckElem * deck_unlink_last(Deck *d);

which removes a card from the end of the Deck. Function should return the address of the element holding the card (without freeing it).

Write a function:

void deck_reverse(Deck *d);

which reverses the order of cards in the Deck. The function should not alocate or dealocate any memory. *Hint!* Use some of the previously defined functions to simplify implementation.

Part 3 (2 points)

Write a function:

void deck_prepend(Deck *dst, Deck* src);

which adds the cards from src Deck to the beginning of dst. No loops or allocations are allowed.

Write a function:

Deck deck_build();

which creates a stripped 24-card Deck (see example at the bottom of the page).

Part 4 (1 points)

Write a function:

void deck_deal(Deck* d, Deck hands[], unsigned n);

which deals cards from Deck d to a number of hands. Cards should be dealt from the back: last card goes to the first hand, second to last to the second hand etc. Operation should continue until all cards have been dealt and d is empty. See example at the bottom of the page for details. No allocations are allowed.

Part 5 (1 point)

Write a function:

void deck_shuffle(Deck *d);

which shufles a Deck of cards. Shuffling should be done as follows: Deck d is dealt out to a randomly selected number of hands - between 2 and 10 (inclusive) - as described in Part 4. Afterwards all hands should be joined together in reverse order, i.e.: last hand, ..., first hand forming new contents of d. *Hint!* Use some of the previously defined functions to simplify the implementation.

Example

Compare your program's output with the following:

**********Part 1********** [QS][10D][10C][10H][AC][JS][AD][JS][AS] [QS] **********Part 2********** Last: [QS] Last: [AS] Last: [JS] Last: [AD] Reversing the rest... [JS][AC][10H][10C][10D][QS] **********Part 3********** d2: [QS][10S][KS][KD][JH][10C][9S][9H][10D][9H] Prepending d2 to d d : [QS][10S][KS][KD][JH][10C][9S][9H][10D][9H] d2: [QH] Prepending d2 to d d : [QH] [QS][10S][KS][KD][JH][10C][9S][9H][10D][9H] d2: [JC][AH][9D][JH][JS] Prepending d2 to d d : [JC][AH][9D][JH][JS][QH][QS][10S][KS][KD][JH][10C][9S][9H][10D][9H] Building deck... [9S][10S][JS][QS][KS][AS][9C][10C][JC][QC][KC][AC][9D][10D][JD][QD][KD][AD] [9H][10H][JH][QH][KH][AH] **********Part 4********** Dealing cards... Hand 1: [AH][JH][AD][JD][AC][JC][AS][JS] Hand 2: [KH][10H][KD][10D][KC][10C][KS]

[10S] Hand 3: [QH][9H][QD][9D][QC][9C][QS][9S] **********Part 5********** [9S][10S][JS][QS][KS][AS][9C][10C][JC][QC][KC][AC][9D][10D][JD][QD][KD][AD][9H][10H][JH][QH][KH][AH] Shuffling 10 times... [JH][JS][10H][JD][QH][10D][10S][9C][JC][QC][QD][KC][9S][AC][10C][9D][KS][KH][9H][AH][KD][AD][AS][QS]

Downloads

- main.c - initial code

Generated with Reader Mode