CMPSC 297 - Introduction to C Programming

PennState

Systems and Internet
Infrastructure Security Labratory

Week #5 - Test

Professor Patrick McDaniel

Overview

- Two part exam:
  1. Creating a card-deck & Implementing 3 functions (`creating`, `printing`, and `sorting`)
     - For loops, printf, etc.
     - Manipulating string of characters and writing to memory
  2. Dynamically allocating memory to store series of suits symbols
     - Malloc, free, strcmp, etc.

Invitation Link: https://classroom.github.com/a/SCrTK2f2

Part #1 - Card Deck Preliminaries

PennState

- The `cmpsc297_deck` variable has an array of integers encoding the deck
- Each card type is represented as an integer:
  - 0=2, 1=3, 2=4, 3=5, 4=6, 5=7, 6=8, 7=9, 8=10, 9=jack, 10=queen, 11=king, 12=ace
  - Suits are defined by an offset of 13:
    - Spades = 0, Clubs = 13, Heart = 26, Diamond = 39
  - Example: 0 = "2 of ♠", 13 = "2 of ♣", 24 = "king of ♠" (symbols are provided)

Part #1 - Card Deck Functions

PennState

- Implementing three functions:
  - `int create_deck(int cards[], int num_cards);`
  - `int print_cards(int cards[], int num_cards);`
  - `int sort_cards(int cards[], int num_cards);`

Part #1 - Card Deck Functions

PennState

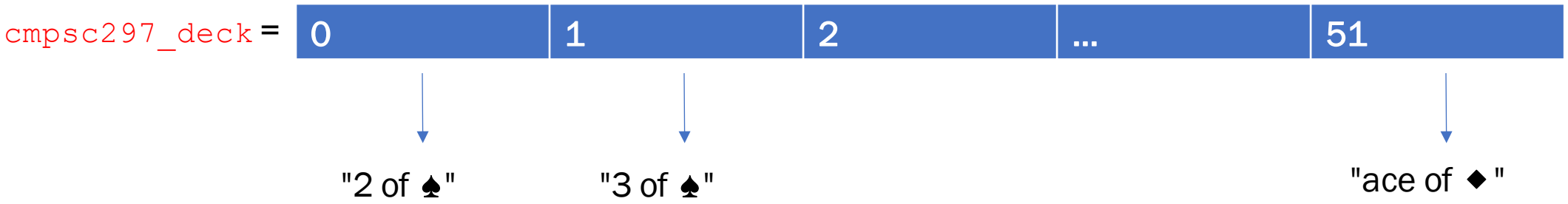
- `int create_deck(int cards[], int num_cards);`
  - You will assign card values to a predefined variable `cmpsc297_deck`
  - Hint: Use for loops to write to the `cmpsc297_deck`

`cmpsc297_deck` =

0	1	2	…	51

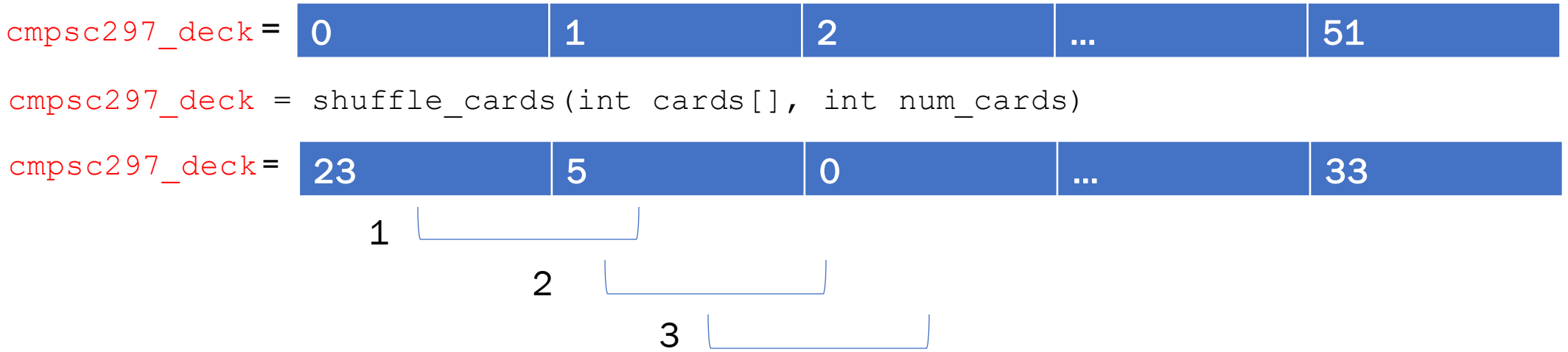
Part #1 - Card Deck Functions

- `int print_cards(int cards[], int num_cards);`
  - You will print the first `num_cards` from your own array
  - For each integer in your `num_cards`, you should print the actual card name (e.g., the value 0 should print "2 of ♠", 24 should print "king of ♣"
  - Use global defines to print out symbols
  - Hint: Use `printf` to write the card values to `STDOUT` & use modulus to extract the suit



Part #1 - Card Deck Functions

- `int sort_cards(int cards[], int num_cards);`
  - You will sort the cards from your own array
  - Call the `int shuffle_cards(int cards[], int num_cards);` that we provide for you
  - The cards should be sorted in groups based on number
  - Hint: Use bubble sort



Part #2 – Suits Generation

PennState

A generator generates 3 series of symbols of suits.

You need to use dynamic memory allocation to get and store these series to memory.

A validation function will check if your series are identical to the original ones.

You have to:

1. Init generator
2. Request the 3 series generated and copy them dynamically (malloc, realloc, etc.)
3. Run validation
4. Close generator

See `part2_generator.h` for the function prototypes of the generator.

```
Series #1

Original:♦ vs Yours: ♦  OK
Original:♣ vs Yours: ♣  OK
Original:♣ vs Yours: ♣  OK
Original:♥ vs Yours: ♥  OK
Original:♠ vs Yours: ♠  OK
Original:♣ vs Yours: ♣  OK
Original:♠ vs Yours: ♠  OK
...
```