C Practice

CS 0449: Introduction to System Software

CS0449 Teaching Assistants

University of Pittsburgh | School of Computing and Information

Agenda

- Course News!
- Lab 2 - C Programming
  - Parts 1, 2, and 3
  - How strings work in C

Course News

- TA office hours
  - See https://cs0449.gitlab.io/fa2023/general
- Lab 2 announced
  - Due: 17:59 Thursday, September 21th, 2023.

Lab 2: C Practice!

Practicing C with three functions

Part 1: Calculating PI!

- TODO: You want to calculate PI
  - We'll use Monte Carlo simulations to calculate the value of PI
    - If we randomly distribute points over a square, and draw a circle within the area, we can use the ratio of points within the square to points within the circle to calculate π
      - » The math is explained more on the lab description. Follow the guide and the video in the Lab
  - DO NOT make a main function!

Part 2: Convert String to Integer

- TODO: Write a program that converts a string to integer
  - Create a file (atoi.c) that has your implementation
- You'll want to iterate through the characters in the string and convert them into integers
  - **How can we change the position or value of a digit within a number…?**
- Some of things to note
  - The string may start with a '-' or '+'.
  - You can assume otherwise it only contains the numbers 0-9
  - Remember that **'0'** is different from **0**! Character 0 ('0') is the ASCII number that represents character 0.
- DO NOT make a main function!

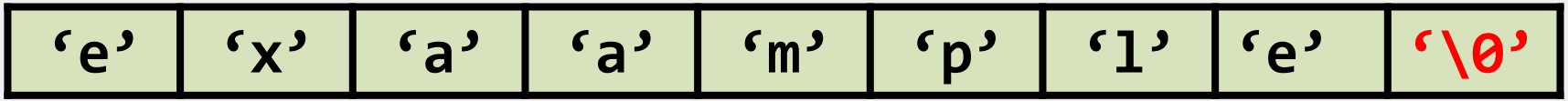
Input: "+523" Output: 523	Input: "124" Output: 124	Input: "-32" Output: -32

Understanding Strings in C

- Strings in C are effectively an **array of chars**
- The end of a string is denoted by the **null terminator '\0'**
  - '\0' is the ASCII character with the value zero

- Without the null terminator, an attempt to read the string will go won't terminate… (until it encounters another zero in memory!)

```
char[] myString = "example";
```



Part 3: Converting integers to strings

- TODO: Write a program that converts an integer to string
  - Create a file (itoa.c) that has your implementation

- You need to convert an integer into a string in accordance with that base that is given.
  - The base can be base 2, base 8, base 10, and base 16
  - Just like in part 2, the sign needs to be taken into consideration
    - This will only apply to numbers in base 10
  - You will be given a char array(aka string) to output your string in.
  - Remember that if number **xyz** is in base b, then **xyz/b = xy**
  - Remember that if number **xyz** is in base b, then **xyz%b = z**