CS246 lab Notes #5 String.h functions and command-line arguments

- String functions
  - #include <string.h>
  - sprintf
    - Say you want to convert a number into the string representation of the number
    - Like fprintf going to a stream, sprintf goes to a string
    - Prototype: int sprintf(char *buf, char *fmt, …)
    - Example: int sprintf(buffer, "%d", num);
  - strchr, strrchr
    - char *strchr(char *str, int ch)
      - Returns a pointer to the first occurrence of ch
    - char *strrchr(char *str, int ch)
      - Returns a pointer to the last occurrence of ch
  - strstr
    - char *strstr(char *str1, char *str2)
      - Returns a pointer to the first occurrence of str2 in str1.
  - strcspn, strspn
    - size_t strcspn(char *str1, char *str2)
      - Returns the starting index of initial segment in str1 consisting entirely of str2.
    - size_t strspn(char *str1, char *str2)
      - Returns the starting index of the first nonmatching segment of str2 in str1.
  - strtok
    - Used to parse string into tokens, kinda like the Java StringTokenizer.
    - Allocates a static copy of the original string
    - Prototype: char *strtok(char *str, char *tokens)
    - For each token delimiter it finds, it returns a pointer to the piece of the string that is broken by that token, i.e. the located token in the target string is replaced with a ‘\0’.
    - Example:
      - strtok("Dianna,Xu,cs246", ",")
      - After the first call, it will return a pointer to the string containing "Dianna"
      - Then you should call it with NULL as the first argument, it will return "Xu". On a third call it will return “cs246”
    - This function MODIFIES its first argument!!!
    - Because it allocates a static buffer while paring, it is dangerous and should be used with caution
  - Two more random string-y functions
    - #include <stdlib.h>
    - int atoi(char *str), long atol(char *str)
      - Returns a int or a long from the given string, respectively
    - double atof(char *str)
- Returns a float from the given string

- Exercises
  - Write a function that counts and returns the number of vowels in a string.
  - Write a program `reverse.c` that echos its command-line arguments in reverse order. i.e. running the program by typing:
    ```c
    reverse void and null
    ```
    should produce the following output:
    ```
    null and void
    ```