- CPSC 275: Introduction to Computer Systems

CPSC 275: Introduction to Computer Systems

Fall 2025

- Syllabus
- Schedule
- Resources
- Upload
- Solution

Lab 5W

This laboratory is a graded, individual assignment. You may use the lecture slides and your own prior lab work, but you may not use other electronic resources or the work of other students. Only the computers in the laboratory may be used.

Exercise 1

Using recursion, write a C function:

```
void dec2bin(unsigned int n);
```

This function accepts a non-negative integer `n` as a parameter and prints its binary representation. You may not use the `%b` format specifier in `printf`; instead, construct the binary output manually. Using this function, write a C program named `convert.c` that reads a non-negative integer from standard input and prints its binary representation.

Sample Input

```
2025
```

Sample Output

```
11111101001
```

Compile your program with:

```
$ gcc -Wall -o convert convert.c
```

Run your program with:

```
$ ./convert < convert.in
```

where `convert.in` is the input file to your program.

Exercise 2

Write a function that, given a string, returns the total number of words:

```
int numwords(char line[]);
```

Using this function, write a C program named `count.c` that reads one or more lines of text with a maximum length of 100 characters from standard input and prints the number of lines, the number of words, and the number of characters.

Sample Input

```
Hello, New England!
Fall has arrived.
```

Sample Output

```
2 6 38
```

Here, there are 2 lines, 6 words, and 38 characters, including spaces and newline characters.

Compile your program with:

```
$ gcc -Wall -o count count.c
```

Run your program with:

```
$ ./count < count.in
```

where `count.in` is the input file to your program.

Exercise 3

Using pointers (no array indexing `[]`), write a C function:

```
int mystrcmp(char *s1, char *s2);
```

which compares the two strings `s1` and `s2` lexicographically and returns `0` if they are equal, -1 if `s1` < `s2`, and 1 if `s1` > `s2`.

Using this function, write a C program named `compare.c` that reads two strings each with a maximum length of 100 characters from standard input and prints the comparison result.

Sample Input

```
Apple Banana
```

Sample Output

```
-1
```

Compile your program with:

```
$ gcc -Wall -o compare compare.c
```

Run your program with:

```
$ ./compare < compare.in
```

where `compare.in` is the input file to your program.

Hand In

Upload your source code (`convert.c`, `count.c`, `compare.c`) **no later than 4:10 p.m.**

- **Welcome: Sean**
  - LogOut