Session 3 - The Finale

(UNDER CONSTRUCTION)

Good luck and happy coding!

Contents

Input

We are covering **input** so we can *finally* create interactive programs in **C**! We will focus on using **scanf** (in `stdio`) to read in user input.

Character Input

For **scanf** to read in a character, we need to pass it a pointer to a char variable! If we passed the value, then it wouldn't be able to fill in the variable for us.

```
char a;
scanf("%c", &a); // read in a character! Store in the variable 'a'
printf("%c\n", a); // print out that character!
```

We can also read in multiple characters!

```
char a,b,c;
scanf("%c%c%c", &a, &b, &c); // read in 3 characters!
printf("%c%c%c\n", a, b, c); // print out those characters!
```

You might wonder *why* scanf doesn't just return the value (or a pointer to the value). This is because `scanf` returns the number of items that were successfully read in! We can inspect this behaviour closely with the following:

```
char a,b,c,d;
int nReads = scanf("%c%c%c%c", &a, &b, &c, &d); // read in 4 characters!

printf("%d\n", nReads);
```

If you compile and run the program (located in `cFiles/input/nReplacements.c`), then you'll notice that trying to enter just one character won't work! The program simply waits until you enter more characters (even if it's a newline or a space) - to observe this behaviour, we'll have to pipe input from a file to the program. We can do this using the following command:

```
cat theFile | ./theProgram
```

So compile `cFiles/input/nReplacements.c`, and try piping in some input (try `txtFiles/input/nReplacementsInput1.txt`)!

String Input

Int Input

Retrying Int Input

Input Exercises

1. Bob is following the course and mistakenly has written:

```c
char a;
scanf("%c", a);
printf("%c\n", a);
```

What mistake has Bob made? What should he have written? What will happen, and what will the output be? What is this an example of?

Hint: Read the first paragraph of Character Input

2. Read the solution to Exercise 1.

3. A program requires you to input the length of the string, and then the string itself. Can you output the number of times the character 'e' appears in the string?

Test the program using the inputs in `txtFiles/InputEx3/Input` which has corresponding correct outputs in `txtFiles/InputEx3/Output`.

Makefiles

- Use https://youtu.be/GExnnTaBELk

Optional Exercises

Acknowledgements

Originally created by Edward Denton.