

INTRODUCTION TO COMPUTER PROGRAMMING: FUNDAMENTALS OF C

Instructor: Gilbert Pajela

Summer 2017

→ Lab 5: Strings

1. Write and test a function `deblank` that takes a string output and a string input argument and returns a copy of the input argument with all blanks removed.
2. Write a function `fact_calc` that takes a string output argument and an integer input argument `n` and returns a string showing the calculation of $n!$. For example, if the value supplied for `n` were 6, the string returned would be `"6! = 6 x 5 x 4 x 3 x 2 x 1 = 720"`. Write a program that repeatedly prompts the user for an integer between 0 and 9, calls `fact_calc` and outputs the resulting string in a box of asterisks of the right size to surround the result. If the user inputs an invalid value, the program should display an error message and reprompt for valid input. Input of the sentinel -1 should cause the input loop to exit. Sample run:

```
Enter an integer between 0 and 9 or -1 to quit => 6
*****
* 6! = 6 x 5 x 4 x 3 x 2 x 1 = 720 *
*****
Enter an integer between 0 and 9 or -1 to quit => 12
Invalid entry
Enter an integer between 0 and 9 or -1 to quit => 0
*****
* 0! = 1 *
*****
Enter an integer between 0 and 9 or -1 to quit => -1
```

3. Write and test a function `hydroxide` that returns a 1 for true if its string argument ends in the substring `OH`.
Try the function `hydroxide` on the following data:
`KOH H2O2 NaCl NaOH C9H8O4 MgOH`

4. Write a program that takes nouns and forms their plurals on the basis of these rules:

- (a) If noun ends in “y”, remove the “y” and add “ies”.
- (b) If noun ends in “s”, “ch”, or “sh”, add “es”.
- (c) In all other cases, just add “s”.

Print each noun and its plural. Try the following data:

chair dairy boss circus fly dog church clue dish

5. Write a program that stores lists of names (the last name first) and ages in parallel arrays and sorts the names into alphabetical order keeping the ages with the correct names. Sample output:

Original list

Ryan, Elizabeth 62
McIntyre, Osborne 84
DuMond, Kristin 18
Larson, Lois 42
Thorpe, Trinity 15
Ruiz, Pedro 35

Alphabetized list

DuMond, Kristin 18
Larson, Lois 42
McIntyre, Osborne 84
Ruiz, Pedro 35
Ryan, Elizabeth 62
Thorpe, Trinity 15

6. Write a program that takes data a line at a time and reverses the words of the line. For example,

Input: birds and bees
Reversed: bees and birds

The data should have one blank between each pair of words.

7. Write and test a function that finds and returns through an output parameter the longest common suffix of two words (e.g., the longest common suffix of “procrastination” and “destination” is “stination”, of “globally” and “internally” is “ally”, and of “gloves” and “dove” is the empty string).

8. Write a program that processes a data file of names in which each name is on a separate line of at most 80 characters. Here are two sample names:
Hartman-Montgomery, Jane R.
Doe, J. D.
- On each line the surname is followed by a comma and a space. Next comes the first name or initial, then a space and the middle initial. Your program should scan the names into three arrays— **surname**, **first**, and **middle_init**. If the surname is longer than 15 characters, store only the first 15. Similarly, limit the first name to ten characters. Do not store periods in the **first** and **middle_init** arrays. Write the array's contents to a file, aligning the contents of each column:

| | | |
|-----------------|------|---|
| Hartman-Montgom | Jane | R |
| Doe | J | D |

Exercises are adapted from the following source(s):

1. Hanly, J. R. (2012). *Problem solving and program design in C*. Pearson Education.