cc-practice-io

Table of Contents

- 1. READ README
- 2. TODO Identify yourself
- 3. DONE Conversion specification
- 4. DONE Integer decimal d"
- 5. DONE Integer decimal precision p
- 6. DONE Scientific notation e
- 7. DONE Variable floating point g

1. READ README

- This file is a practice file for C input/output functions `printf` and `scanf`.
- You will learn how to:
    1. …x
- Time: approx. 30-60 min.
- When you're done with a section move the cursor on the section heading and type `S-<right>` (or SHIFT+ <right-arrow>).

2. TODO Identify yourself

- replace the placeholder `[yourName]` in the header of this file by your name and save the file (`C-x C-s`).

3. DONE Conversion specification

Recreate the output below exactly, using only format specifiers (no extra white space).

```
: ....|....|....|
:   100100
: 200200
:        3.1416
: 3.141593
```

—— SOLUTION ——

```
printf("....|....|....|\n");
printf("%8d\n", 100100);
printf("%-10d\n", 200200);
printf("%13.4f\n", 3.141593);
printf("%-.6f\n", 3.141593);
```

```
....|....|....|
  100100
200200
       3.1416
3.141593
```

4. DONE Integer decimal d"

Show that the default for `d` is p=1. Print the numbers 1, 1, 100 and 10000 with the specifiers `%d`, `%.1d`, `%.5d`, `%.2d`. Print each expression on its own line, but use only ONE `printf` statement.

—— SOLUTION ——

```
printf("....|....|....|\n");
printf("%d\n %.1d\n %.5d\n %.2d\n", 1, 1, 100, 10000);
```

```
....|....|....|
1
 1
 00100
 10000
```

5. DONE Integer decimal precision p

Print the number 42 on a space of 10 characters with precision 5.

The result should look like this:

```
: ....|....|....|
:      00042
```

—— SOLUTION ——

#+name spec1

```
printf("....|....|....|\n");
printf("%10.5d\n", 42);
```

```
....|....|....|
     00042
```

6. DONE Scientific notation e

- Print 1, 1000.100, and 1,000,000,000,000,000 using `%e`.
- Provide for the required number of decimal positions (but not more)
- Print each expression on its own line with its own `printf` function.
- Add the header-argument `:results output` to the code block

Desired output:

```
: 1e+00
: 1.0001e+03
: 1e+15
```

—— SOLUTION ——

```
printf("%1.e\n", 1.);
printf("%.4e\n", 1000.1);
printf("%.e\n", 1000000000000000.);
```

```
1e+00
1.0001e+03
1e+15
```

7. DONE Variable floating point g

- Use the format specifier `g` to display the following numbers: `200`, `3.142574654` with p=8, `2.71`, and `!5`.
- print each on a line of its own, but use only **one** `printf` statement to do it!
- `!N` is defined as the factorial of `N`.

—— USE THIS CODE BLOCK ——

```
printf("....|....|....|\n");
...
```

—— SOLUTION ——

```
printf("....|....|\n");
printf("%g\n%.8g\n%g\n%g\n", 200., 3.142574654, 2.71, 5.*4.*3.*2.*1.);
```

```
....|....|
200
3.1425747
2.71
120
```

Author: [yourName] (pledged)
Created: 2022-06-06 Mon 12:01