

CS 2022 Homework 1

Due:
Tuesday September 6th 2011 at 11:59 PM
on <http://cms.csuglab.cornell.edu>.

In this assignment you are asked to write three small C programs to demonstrate command of the basic principles we have discussed in class so far.

Factorials

Write two C programs `fact1.c` and `fact2.c` that calculate the factorial of a number that will be passed to your program as a command line argument.

`fact1.c` should use recursive function calls to compute the answer, while `fact2.c` should not make any recursive calls and should use loops instead.

After your programs are compiled, the user should be able to run them as follows:

```
> ./fact1 0
1
> ./fact1 6
720
> ./fact2 4
24
> ./fact2 8
40320
```

Zipper

Your third small program, `zipper.c`, will take-in multiple command-line arguments and “zip them together”. The output of the program will be several strings printed onto the screen, one per line. The length of each output string will be equal to the number of command-line arguments passed to the program. The output strings are constructed as follows:

- the first string is composed of the first characters of all the command-line arguments (in order).

- the second string is composed of the second characters of all the command-line arguments (in order). ... and so on.

The number of output strings is equal to the length of the longest passed-in argument. If the passed-in arguments are of different lengths, then after exhausting the letters of a shorter argument, an empty space will be printed in its slot. The following examples should make things clearer:

```
> ./zipper aa bb
```

```
ab
```

```
ab
```

```
> ./zipper abcd efgh
```

```
ae
```

```
bf
```

```
cg
```

```
dh
```

```
> ./zipper hello green 12345
```

```
hg1
```

```
er2
```

```
le3
```

```
le4
```

```
on5
```

```
> ./zipper hi hello bye
```

```
hhb
```

```
iey
```

```
le
```

```
l
```

```
o
```

```
> ./zipper ThisIsOk
```

```
T
```

```
h
```

```
i
```

```
s
```

```
I
```

```
s
```

```
O
```

```
k
```

Tips

Here are some tips that you might find helpful in completing this assignment:

- Use the `atoi` function to convert a string argument into an integer.
- Use the `printf` function to print to the screen.
- Useful string manipulation routines are available by *including* the `string.h` header file. An important one in this assignment is `strlen(argument)` that computes the length of a string.
- The last slide of lecture 2 has a useful hint.
- If you run into an infinite-loop and your program does not terminate, use `<ctrl>-c` to terminate your program forcefully. If that does not work, start another terminal and type “`killall -9 name_of_your_app`”.

Submission and Testing

Submit your work on CMS (<http://cms.csuglab.cornell.edu/>) by the deadline. Make sure you have been added to CMS early, and contact me via email if you do not have a CMS account. You should submit your **source code** (the `.c` files) of your application and **not** the compiled binaries.

Your programs will be compiled with `gcc` and tested in a Linux environment. You are free to choose the environment of your liking to develop the solutions, but keep in mind that testing will be on a fixed environment, and your application is expected to run on that.

Academic Integrity Reminder

Remember that you may have general discussions about how to approach this problem with your peers, but you should work on the final solution by yourself, alone. If you are stuck or are having trouble, you may email me, talk to me after class, or come to office hour on Wednesday. Make sure to email me before coming to office hour because I might be in my lab downstairs.

Good Luck!