

Homework 7

Due: 2:30 PM Apr. 24, 2008

For this assignment, you should save all your code in the directory

```
/u/<login>/course/cs004/hw7
```

To run the handin script, enter the following command in a terminal:

```
cs004_handin hw7
```

No stencil code is provided for this homework. You should refer to the lectures and your lab assignments to get you started if you do not remember exactly how to structure a program in C.

Problem 7.1

Implement this program in a file named `print2d.c`. It should populate a 5 x 5 matrix of integers with a row-by-row input from the user, and then print out the matrix, like this:

```
Please enter row 1: 1 2 3 4 5
Please enter row 2: 6 7 8 9 10
Please enter row 3: 11 12 13 14 15
Please enter row 4: 16 17 18 19 20
Please enter row 5: 21 22 23 24 25
```

Matrix:

```
 1  2  3  4  5
 6  7  8  9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```

Your matrix does not have to look exactly like this example, though it should be formatted nicely. The tab character (`'\t'`) will be useful in printing out your matrix.

Use `gcc -Wall -o print2d print2d.c` to compile, and `./print2d` to run.

Problem 7.2

In this problem, you'll be finding all prime numbers less than 100. You will have to loop through numbers 2 to 99, and for each number, test if it is prime, and then print a message that tells the user the result. You can test the primality of a number x by seeing if every integer greater than 1 and less than that number does not divide x . You can tell that a number a does not divide a number x if $x \% a \neq 0$. Here, `%` is the “modulus” operator, which returns the remainder after x is divided by a .

A sample run of this program would look like:

```
./prime
2: IS PRIME
3: IS PRIME
4: NOT PRIME
5: IS PRIME
6: NOT PRIME
....
....
99: NOT PRIME
```

Save your work in the file `prime.c`. Compile this problem with the command `gcc -Wall -o prime prime.c`.

Problem 7.3

Write a program that reads in an arbitrary-length series of digits separated by white space and then converts the digits to a single integer and prints that integer. The user will signify the end of the string of digits by entering a negative number. One run of the program might look like this:

```
Enter your digits: 5 6 2 9 -1
The number you have entered is 5629
```

You may assume that the user will not enter more than 10 digits. **Do not** simply print out the digits with no space between them - you must convert

CS004

Homework 7 **2:30 PM Apr. 24, 2008**

the digits to an actual integer value and store that value in an `int` variable!
This program is tougher than it may seem, so come to TA hours if you need help! Write your code in a file named `digits.c`.

Compile using `gcc -Wall -o digits digits.c`, and run by typing `./digits`.