

Handout 4

while loop

4.1 Terminal tip: `ftp`

It is possible to connect to your computer in the lab from other places. To get files use `sftp` “secure file transfer protocol”

```
sftp student@simpson7.lclark.edu
<enter password: student>
```

You can now move about the files on the computer using `ls`, `cd`, etc. To retrieve a file from the computer, use the command `get`

```
sftp> get myfile.c
```

Once you're done, you can exit

```
sftp> exit
```

Note: The file that you get will be placed in whatever directory you were in when you used the `sftp` command. First navigate to where you want to be, then connect using `sftp`.

There is also the `put` command that allows you to put files on the other machine.

Take some time right now and exchange files with your neighbor using `sftp`.

4.2 A simple **while** loop

```

/*
program to print squares of integers
*/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

int main()
{
    // obtain max
    double max;
    printf("Enter a positive integer ");
    scanf("%lf", &max);
    printf("\n");

    // print squares up to max^2
    double k =1;
    while (k <=max)
    {
        printf("%lf \n", k*k);
        k = k+1;
    }
}

```

Note:

- You can replace `k = k+1;` with `k++;`
- You can force the program to stop by entering CTRL+C

4.3 Practice problems

1. Write a program `sum-squares.c` that takes in a positive integer n and returns the sum

$$1^2 + 2^2 + \dots + n^2.$$

2. Write a program `factorial.c` that takes in a non-negative integer n and returns $n!$. To be especially fancy, have your program include the possibility that $n = 0$. (Remember that $0! = 1$.)
3. Write a program `isprime.c` that takes in a positive integer and determines whether the number is prime or not.

4.4 Homework: `doublep.c`

Suppose you are given one penny on day 1, two pennies on day 2, four pennies on day 3, etc., so that each day your payout is twice that of the previous day. Write a program `doublep.c` that computes how much (in total) will you have received after 30 days?

4.5 Lab: `forks.c`

Suppose that every day, 1% of forks in the dining hall are “lost.” Suppose also that every day, one of the dining hall workers opens a box of 10 new forks, adding them to the collection of forks available to use. Finally, suppose that at the beginning of the semester there are 3000 forks available in the dining hall. Write a program `forks.c` that makes a table showing how many forks are present in the dining hall every 10 days. Your table should look something like this (but with the gaps and question marks filled in):

Day	Forks
0	3000
10	???
20	???
100	???