Exercise 3: Compiling on the Command Line

Linux provides tools to compile C, C++, and many other languages. For this exercise, create and open up a file with vim or gedit, for instance, if you're using C,

```
vim my_file.c
```

or

```
gedit my_file.c &
```

Or if you're using C++ you'd enter,

```
gedit my_file.cpp &
```

Write out a Hello World program in C or C++. For instance in C you'd write,

```
#include <stdio.h>
int main(){
	printf("Hello world!\n");
}
```

or in C++ you'd write,

```
#include <iostream>
int main(){
	std::cout << "Hello world!" << std::endl;
}
```

Save the file and exit the text editor. To do this in gedit click the Save button and exit. In vim hit the escape button, then type ":wq" (which stands for **w**rite **q**uit). Now, at the command line, if you wrote your program in C, type,

```
gcc my_file.c
```

Or if you wrote your program in C++, type,

```
g++ my_file.cpp
```

The compiler will output a binary file named a.out. To run it, type in

```
./a.out
```

You should see "Hello world!" as output! Congratulations!

I/O Redirection

I/O redirection is useful for testing your program's output against a test case.

1) ./a.out > output1
2) cat output1
What did > do?

3) ./a.out >> output1
4) cat output1
What did >> do?

Change your file to take an input of two numbers and output the sum,
For instance in C you'd write,

```
#include <stdio.h>
int main(){
        int num1,num2;
        scanf("%d",&num1);
        scanf("%d",&num2);
        printf("%d\n",num1+num2);
}
```

or in C++ you'd write,

```
#include <iostream>
using namespace std;
int main(){
        int num1,num2;
        cin >> num1;
        cin >> num2;
        cout << num1+num2 << endl;
}
```

5) gedit input1
6) Put this into the file:
1
2
7) Save the file

After you compile your program,
5) ./a.out < input1 > output1
6) cat output1
What does it show you?