

Lab 8

Out: April 13, 2008

To get started, copy the stencil code to your `lab08` directory with the following command:

```
cp /course/cs004/pub/lab08/* ./
```

Make sure you are in your `lab08` directory before running this command.

Problem 8.1

In this task you will write a program that will read in positive numbers from the command line and add them, printing the sum to the terminal. Your program should repeatedly prompt the user to enter a number until the user enters a negative number. Then it should print the sum of the positive numbers. Here's an example of what your program should look like in a terminal window:

```
Enter a number: 6
Enter a number: 8
Enter a number: 10
Enter a number: -1
The sum is: 24
```

Hint: You need to ask the user to input a number an arbitrary number of times, so what kind of loop should you use?

Modify the `sum.c` file to implement this program. Compile it by running the command `gcc -Wall -o sum sum.c`, and run it with `./sum`.

Problem 8.2

In the `isbn.c` file, write a program that checks the validity of ISBN numbers. Every published book has an ISBN number, normally right above the barcode. An example ISBN number is 0-393-96945-2. To avoid bad scans of ISBN numbers, the 10 digits have the following property:

$$(10d_1 + 9d_2 + 8d_3 + \dots + 2d_9 + d_{10}) \bmod 11 = 0$$

Where d_1 through d_{10} are the 10 digits of the ISBN. Your program should take in a 10-digit ISBN and determine whether or not the number is valid. Two sample runs are shown below:

```
> ./isbn
Please enter an ISBN number (no dashes or spaces): 0393969452
VALID
> ./isbn
Please enter an ISBN number (no dashes or spaces): 0393969451
INVALID
```

Hint: You'll need read in each of the ten digits of the ISBN one by one into a variable of type `char`. You can scan these as a string into an array or one at a time as a single character into a variable. To get the integer value of the digit that a `char` represents, you can subtract `'0'` (the the character representing the number zero) from it.

Compile using `gcc -Wall -o isbn isbn.c` and run with `./isbn`.

Problem 8.3

The program you will write for this problem should ask the user to input 5 integers, store those numbers in reverse order in a `int` array, and then print the entries of the array. The input can be read in all at once or in pieces. Either way the input must be seperated by spaces. Below is an example of what your program should output when run in a terminal:

```
Please enter 5 integers: 7 1 24 3 8
Reverse: 8 3 24 1 7
```

Modify the `reverse.c` file to implement your program. To compile, type `gcc -Wall -o reverse reverse.c`, and type `./reverse` to run.