CSCI-14 Assignment #5, simple looping (40 points) due 3/1/18

1. Write a C++ program that prints a table of the characters for the ASCII codes from `' '` (space) to `'~'` (tilde). Print the characters 16 to a line separated by spaces. Do not use the ASCII codes (e.g., space is 32, ~ (tilde) is 126), use the characters themselves (space is `' '`, tilde is `'~'`) to control the loop. (This is generally going to be the right way to do something with ASCII characters – use the characters not the codes!) <u>Do not print a trailing space on any line (except perhaps the last one after the `'~'`).</u> This illustrates the 'fence post' problem – you need 15 spaces and 1 end-of-line on each line. There is one space between any two ASCII characters on the lines. You may only use ONE loop for this program, which will count from `' '` to `'~'`. Do not test for printing the specific characters that end each line (e.g., `'/'` or `'?'`) to determine when to end the lines. Instead, count the number of characters you have printed on the lines. (20 points)

   The output should look like this:

```
  ! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~
```

2. A worker is paid one cent for the first day's work, 2 cents for the second, 4 cents for the third, and so on with, the pay in pennies being doubled each day. At the end of the job, she gets a single lump sum paycheck in dollars and cents. Write a program that prompts the user for a number of days, then prints each day's pay in pennies, followed by the total accrued pay in dollars with a fractional part for cents at the end. The only floating-point value in this program is the total pay. Add each day's pay (an int) to the total pay. You will need 10 digits of precision to get the correct results. Format the total pay to show two decimal places for cents. Test with several numbers of days. Work out (please – do not just try it until it fails, solve this first!) at how many days the program will stop working correctly with an int for the number of pennies, and include a test for that number of days and a test for a larger number of days. (20 points)

   A test run should look something like this for 8 days:

```
How many days? 8
Pay for day 1 = 1
Pay for day 2 = 2
Pay for day 3 = 4
Pay for day 4 = 8
Pay for day 5 = 16
Pay for day 6 = 32
Pay for day 7 = 64
Pay for day 8 = 128
Total pay is $2.55
```