CMPS 1500 Introduction to Computer Science I – Fall 13

2. Homework

Programming portion (problem 3) due **9/17/13** at 11:55pm on Blackboard.
Written portion (problems 1 and 2) due **9/18/13** at the beginning of class

1. **Binary Multiplication (6 points)**

   The goal of this exercise is to develop a logic circuit that multiplies two two-bit binary numbers. You are allowed to use AND, OR, and NOT gates, as well as half-adders. A half-adder is the first adder that we discussed in class, that has two input bits and two output bits.

   Let $x = x_2 x_1$ and $y = y_2 y_1$ be two two-bit binary numbers with bits x_1, x_2, y_1, y_2. The binary product of x and y is a three-bit number $z = z_3 z_2 z_1$.

   (a) (1 point) Fill in the blanks below for the bits of the two partial products with Boolean formulas in x_1, x_2, y_1, y_2.

$$
\begin{array}{ccc}
 & x_2 & x_1 \\
\times & y_2 & y_1 \\
\hline
 & \ldots & \ldots \\
+ \quad \ldots & \ldots & \\
\hline
z_3 & z_2 & z_1
\end{array}
$$

   (b) (5 points) Now, devise a logic circuit to add the two partial products, by stringing together half-adders. The inputs to your half-adders will mostly consist of the outputs of circuitry that computes the partial products. (Note: It suffices to use the simpler half-adder, because the number of carry bits and other inputs never exceeds two.)

2. **Machine Language (4 points)**

   Convert the sequence of 32-bit words below to a sequence of MIPS32 instructions. What does the code do?

   ```
   00100000000010000000000000000110
   00100001000010010000000000001100
   00000000000010010101000001000000
   00000001010010000101100000100000
   ```

3. **Stars (10 points)**
   The goal of this exercise is to write a Python script that for a given number n prints two triangles of stars: One solid triangle and one outlined triangle. Each triangle consists of n rows: the first row has length one, the second has length two, the third row has length three, and so on, until the last row of length n. In the solid triangle, the i-th row consists of i stars. The outlined triangle has the same shape as the solid triangle, but now the rows (except the first and last row) only have a star at the beginning and at the end, but spaces in between. So, the i-th row has one star, $i - 2$ spaces, and another star. An example for $n = 6$ is below:

```
*
* *
* * *
* * * *
* * * * *
* * * * * *


*
* *
*   *
*     *
*       *
* * * * * *
```

Please write a Python script that produces the exact output of the example above. You should proceed in the following steps:

   (a) (3 points) Write a function `printStars(k)` that prints a row of k stars. So, `printStars(4)` will print: `****`

   (b) (3 points) In the main body of your script, call `printStars` repeatedly in order to create a solid triangle. Try to not use the knowledge that $n = 6$ but make it work for a variable number n. (Side note: Can you create a solid triangle for $n = 20$ easily?)

   (c) (4 points) Now, create the outlined triangle. Note that the first row and the last row are the same as for the solid triangle. You could consider defining another function that prints a single row (other than the first or last) of the outlined triangle.

   Please name your file `lastName_firstName_hw2.py` and submit it on Blackboard.