**Homework (9/27/2014)**

1. **Ex4\_1**

The Square Within (modified from codercharts.com)

**Description**

Bobby is an interior designer on Times Square. As such he often faces the difficulty of dividing up the space to fill it with furniture. The goal of this puzzle is to help him determine how many possibilities he has.

We modelize the problem the following way.

The room is a square of size NxN unit 1 ft square

An object can occupy a single square, or a group of squares which form a square of size MxM (with M lesser than or equal to N in order to fit it in the room)

For example, if the room is of size 2x2:

a 1x1 square can be put at position (0,0) or at position (1,0) or at position (0,1) or at position (1,1)

a 2x2 square can be put at position (0,0) only.

Consequently there are 5 ways of placing squares in the room (4 ways for size 1x1, and 1 way for size 2x2).

Your program will take a list of room sizes and print their corresponding number of ways to place squares in them.

**Input**

The file name(data.in) of the input file containing the various room sizes

The input file contains one integer per line, to represent a room size N.

**For example:**

1

4

2

**Output**

Your program must print the number of ways to place squares in each of the room sizes given in input. Your program must print to the standard output (stdout/printf/echo/puts). It should print one integer per line.

**Using the above example, the output must be:**

1

30

5

1. **Ex4\_2**

The missing parenthesis(modified from codercharts.com)

# Description

# In an effort to help users who forget to put parenthesis around their expressions, we designed this puzzle to use artificial intelligence to find the most likely solution. The artificial assumption makes use of the common sense expression: Bigger is better.

# The puzzle will take expressions of the form:

# a op1 b op2 c

# where

# a, b, c are integers

# op1, op2 are operators: +, - or \*

# The artificial intelligence assumes that parenthesis are missing, e.g. that the expression could have been written either (a op1 b) op2 c or a op1 (b op2 c). Knowing that humans make mistakes it decides to place the parenthesis so that the value of the expression is the biggest (remember, bigger is better).

# Input

# The input file name is “data.in”, which is formatted as follows:

# the first line contains the number n of expressions to evaluate

# the n following lines each contain an expression of the form

# a op1 b op2 c where

# a, b, c are integers

# op1, op2 are operators: +,- or \*

# Example:

# 5

# 2 - 9 \* 8

# 6 \* 8 + 1

# 9 + 10 - 9

# 7 - 7 + 7

# 5 \* 5 \* 3

# Output

# The output should be printed on the standard output (printf, etc...)

# For each expression you should print its "bigger is better" value.

# For the previous example:

# -56

# 54

# 10

# 7

# 75

1. **Keep working on last week 2010 ProCo problem until 5.4 (Leave 9.\* problem as optional)**
2. **Finish following codeforecs.com problems,**

**5A – Triangle**

**7A - Kalevitch and Chess**

1. **Work on USACO training Section 1.1 problems**