

## Junior Knights Programming Assignments: Nested Loops

### Part A: Parallelogram (para.py)

Write a program that prints out a parallelogram of stars with a desired length and width, entered by the user. Carefully think about how many spaces and how many stars go on each line.

#### Sample Run #1

What is the length of your parallelogram?

3

What is the width of your parallelogram?

7

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

### Part B: Right-Justified Triangle (triangle-v2.py)

Write a program that prints out a right-justified isosceles triangle of stars with a desired number of rows, entered by the user. Once again, carefully think about how many spaces go on each line.

#### Sample Run #1

How many rows do you want in your right-justified triangle?

6

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

### Part C: Multiplication Table (mult.py)

Write a program that prints out a right-justified isosceles triangle of stars with a desired number of rows, entered by the user. Once again, carefully think about how many spaces go on each line.

#### Sample Run #1

How many rows do you want in your table?

6

1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24
5	10	15	20	25	30
6	12	18	24	30	36

### **Part D: Abundant Numbers (abundant.py)**

An abundant number is one whose proper divisors sum to a greater number than itself. For example, 12 has the proper divisors 1, 2, 3, 4 and 6. Adding these we get 16, which is bigger than 12. (A proper divisor is any number smaller than a given number that divides into it. Thus, all of 12's divisors are proper divisors except for 12 itself.) In this program, you will ask the user to enter a low and high bound, and you will print out all the abundant numbers in that range, inclusive.

**Note: If you want more of a challenge, try printing the list out with commas in between in the right places. This will require processing the list of items twice.**

### **Sample Run #1**

What is your low bound?

**10**

What is your high bound?

**30**

The abundant numbers from 10 to 30 are 12 18 20 24 30.