Computing Powers

Write a program called powers that takes two command line arguments (call them e and n. It will print the values of i^e for i from 1 to n.

You are not permitted to use the pow function or any other library function that computes powers.

You must write and use a function

```
int power(int a, int e) {
  ...
}
```

that computes a^e. Do this by multiplying a by itself e times.

You must check for the presence of two command line arguments and print a proper usage message if they are not supplied.

Print the output in two columns, one of width 3 and the other of width 12.

Here are some sample runs:

```
> ./powers 3 10
  0           0
  1           1
  2           8
  3          27
  4          64
  5         125
  6         216
  7         343
  8         512
  9         729
 10        1000
> ./powers 3
usage: powers e n
```

Recursive Powers

Write a program called rpowers1 that is like powers, except that the function that computes powers is recursive. Use the following formula:

$$a^e = \begin{cases} 1 & \text{if e = 0} \\ a \cdot a^{e-1} & \text{otherwise} \end{cases}$$

More Efficient Recursive Powers

Write a program called rpowers2 that is like rpowers1, except that it uses the following recursive formula.

$$a^e = \begin{cases} 1 & \text{if } e = 0 \\ a & \text{if } e = 1 \\ a \cdot a & \text{if } e = 2 \\ (a^{e/2})^2 & \text{if } e > 2 \text{ and } e \text{ is even} \\ a \cdot (a^{(e-1)/2})^2 & \text{if } e > 2 \text{ and } e \text{ is odd} \end{cases}$$

Use a switch statement to handle the cases.

Pascal's Triangle

$n! = 1 \cdot 2 \cdot 3 \ldots n$. Factorials are important because $n!$ is the number of permutations (rearrangements of n items. $0!$ is defined to be 1.

Factorials have been studied at least since 300BC in Indian literature.

The binomial coefficients are the numbers $\binom{n}{m} = \dfrac{n!}{m!(n-m)!}$. We read $\binom{n}{m}$ as n choose m because it is the number of ways of selecting m objects from a collection of n objects.

Note that the notation is not division. It's just that we write n over m enclosed in parentheses. Don't blame me. The notation was made up by the Austrian mathematician Andreas von Ettingshausen in 1826. However, binomial coefficients go back to the Indian mathematician Bhāskara II in the 12th century.

Pascal's triangle of size n is an arrangement of the binomial coefficients $\binom{i}{j}$ for j from 0 to i on each line, where i goes from 0 to i.

For example, the triangle of size 4 looks like this:

$$\begin{array}{lllll} \binom{0}{0} \\ \binom{1}{0} & \binom{1}{1} \\ \binom{2}{0} & \binom{2}{1} & \binom{2}{2} \\ \binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} \\ \binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4} \end{array}$$

Filling in the numbers gives this triangle:

$$\begin{array}{ccccc} 1 \\ 1 & 1 \\ 1 & 2 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 4 & 6 & 4 & 1 \end{array}$$

Write a program called pascal that prints Pascals triangle, but formatted a shown below:

```
                          1
                       1     1
                    1     2     1
                 1     3     3     1
              1     4     6     4     1
           1     5    10    10     5     1
        1     6    15    20    15     6     1
```

Each number is printed in a field of width 6, but the rows are indented as shown.

Include a factorial function

```
int factorial(int n) {
   ...
}
```

and a choose function

```
int choose(int n, int m) {
  ...
}
```

that computes $\binom{n}{m}$.