Assignment 4: Binomial Coefficients

Objective: To learn how to utilize basic recursion for a certain relation.

Description: You are to implement a program that calculates a given Binomial Coefficient. This is defined by the following formula:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

Where "!" represents a factorial, which is defined as such:

$$n! = n * (n-1) * (n-2) * (n-(n-1))$$

Ex: 5! = 5 * 4 * 3 * 2 * 1 = 120

6! = 6 * 5 * 4 * 3 * 2 * 1 = 720

0! = 1 (by the definition of factorial)

WARNING: LOTS OF MATH ON PAGE!

DO NOT PANIC! It is JUST to explain where this comes from

Feel free to skip this page if you do not care.

That said, it may help you understand the assignment better…

How is this useful in any way, shape, or form? Well, let us take a look at the following binomial:

$$(x+y)^3$$

What if we want to expand this out? Well, we could either just do it the old-fashioned way, or…

$$\binom{3}{0} x^3y^0 + \binom{3}{1} x^2y^1 + \binom{3}{2} x^1y^2 + \binom{3}{3} x^0y^3$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$\frac{3!}{0!(3-0)!} \qquad \frac{3!}{1!(3-1)!} \qquad \frac{3!}{2!(3-2)!} \qquad \frac{3!}{3!(3-3)!}$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$\frac{6}{1(3!)} \qquad \frac{6}{1(2!)} \qquad \frac{6}{2(1!)} \qquad \frac{6}{6(0!)}$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$1x^3y^0 \quad + \quad 3x^2y^1 \quad + \quad 3x^1y^2 \quad + \quad 1x^0y^3$$

This is why this relation is called a Binomial Coefficient, as it determines the coefficient of a certain binomial. This is useful for big coefficients (like (x+y)^50)

TRIVIA: Notice how each Binomial Coefficient's inverse Binomial Coefficient was *exactly the same* as itself? See: $\binom{3}{1}$ was equal to $\binom{3}{2}$. This is always true, no matter what.

READ THIS PAGE!

Function: What you will be programming is a single function that takes two inputs, **n** and **k**, and finds the Binomial Coefficient of $\binom{n}{k}$**.** While this *technically* can be done without recursion, it is highly, HIGHLY recommended you use recursion for this.

IMPORTANT: You will need to be aware of the **base cases** of this relationship.

HINT: On the previous page (if you read it), notice how the coefficient was **<u>equal to 1</u>** when **k** was either <u>equal to 0</u> or <u>equal to **n**</u>?

HINT 2: There are two base cases, both of which are based on the value of **k**.

HINT 3: Reread the above two hints, it *literally tells you* what the two base cases are.

The function is defined as such:

```
def Binomial_Coefficient(n, k):          #(Please spell this right)
    # code goes here
```

Oh yeah, one more thing…

READ THIS PAGE!

PASCALS IDENTITY: Given **n** objects, how many ways can we pick **k** of them?

For all of k and n when $1 \leq k < n$

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

REMEMBER: Your Binomial_Coefficient function is essentially $\binom{n}{k}$ in code form. Think about how **Pascals Identity** can be applied via recursive calls. Also, don't forget the **BASE CASES** mentioned on the previous page!

That's all folks, get to it.

Happy Coding :)