2.3 binomial coefficient

#include <iostream>  
#include <math.h>  
*//2.3 Binomial coefficient*int binom(int n, int k)  
{  
 if ((n < 0) || (k < 0) || (k > n)) return 0;  
 if ((k == 0) || (k == n)) return 1;  
 int i = 0;  
 int denom = 1;  
 for (i = 1; i <= k; ++i) {  
 denom \*= i;  
 }  
 int num = 1;  
 for (i = 1; i <= k; ++i) {  
 num \*= (n+1-i);  
 }  
 int b = num/denom;  
 return b;  
}  
  
int factorial(int n)  
{  
 return (n == 1 || n == 0) ? 1 : factorial(n - 1) \* n;  
}  
  
int main(){  
  
 int n = 10;  
 int k = 2;  
  
 std::cout<<binom(n,k)<<'\n';  
 std::cout<<(factorial(n)/(factorial(k)\*factorial(n-k)));  
}

This is the original code and it works. I’ve tested it against my own factorial function.

Binom(10,2) returns 45,

Factorial(10) / (factorial(8)\*factorial(2)) also returns 45.

It’s not optimized because there is 2 loops and since the loops are independent, we can combine it into one for loop. Also the simplification of canceling n! with (n-k)! is not necessarily the best choice for a given n,k. I will explain on the following page.

Ink Drawings
Ink Drawings
￼
￼￼
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
￼
￼
￼


Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
￼
￼
￼￼￼
￼
￼
￼
￼
￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼
￼
￼
￼
￼
￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼
￼
￼
￼￼￼
￼
￼
￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼
￼￼￼￼￼￼￼￼￼￼￼￼
￼￼￼￼￼
￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼


Ink Drawings
￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼
￼￼￼￼￼￼￼￼￼￼￼
￼
￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼
￼
￼￼￼￼￼￼￼￼￼￼￼
￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼
￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼￼
￼￼￼


#include <iostream>  
#include <math.h>  
*//2.3 Binomial coefficient*int binom(int n, int k)  
{  
 if ((n < 0) || (k < 0) || (k > n)) return 0;  
 if ((k == 0) || (k == n)) return 1;  
 int i = 0;  
 int result = 1;  
  
 if (k < n-k)  
 k = n-k;  
 *//since k is smaller, the loop is smaller.* for (i = 1; i <= k; ++i) {  
 result /= i;  
 result \*= (n+1-i);  
 }  
 return result;  
}  
  
int factorial(int n)  
{  
 return (n == 1 || n == 0) ? 1 : factorial(n - 1) \* n;  
}  
  
int main(){  
  
 int n = 10;  
 int k = 2;  
  
 std::cout<<binom(n,k)<<'\n';  
 std::cout<<(factorial(n)/(factorial(k)\*factorial(n-k)));  
}

Console output

45

45

I’ve combined the two for loops into 1 loop. The main optimization made to the code is that we pick k to be the min( k, n-k) . since the k is the loop counter, the smaller, the faster the loop ends. Given n=10, k =2, the loop will pick k = 2 and not 10-2 = 8. The loop will calculate result = 1\*10, result / 1, result \* 9, result /2. That’s the end of the program. If the input is n= 10, k = 8. We change k to 2 because min(8, 10-8=2) is 2. The loop will be the same as n=10, k =2.