Voici trois programmes python différents en application de la loi binomiale :

Modèle 1 : avec une table

```python
def pascal(n,k) :
   L1 =[ 1 , 1 ]
   L2 =[ 1 , 1 ]
   for i in range ( 2 , n+1) :
      for p in range ( 1 , i ) :
         L2[p]=L1[p-1]+L1[p]
      L2 . append ( 1 )
      L1=L2[ : ]
   return ( L2[k] )


print("Saisir les paramètres de la loi binomiale :")
rang = int(input("n ="))
proba = eval(input("p ="))
coeff = int (input("k ="))


coeff_bin = pascal(rang, coeff)
loi_binom = coeff_bin * proba**coeff * (1-proba)**(rang-coeff)
print("p(X = ", coeff , ") = " , loi_binom)
```

Modèle 2 : avec une fonction auto-référente

```python
def coeff_bin(k,n):
   if n == k or k == 0:
      resultat = 1
   elif k > 0 and k < n:
      resultat = coeff_bin(k,n-1) + coeff_bin(k-1,n-1)
   return resultat


n = int (input("n ="))
p = eval(input("p ="))
k = int (input("k ="))


coeff_binomial = coeff_bin(k,n)
valeur = coeff_binomial * p**k * (1-p)**(n-k)
print("p( X =" , k , ") =" , valeur)
```

Modèle 3 : calcul des factorielles pour résoudre p(5<=X<=12) pour B(43,0.2)

```python
# factorielle
def fact(k):
   i = 1
   if k == 0:
      return 1
   else:
      for j in range (1,k+1):
         i = i * j
      return i
# loi binomiale
```

```
def binom(n,p,k):
    return fact(n)/(fact(k)*fact(n-k))*p**k*(1-p)**(n-k)
# loi binomiale sur un intervalle
def binom_inter(n,p,a,b):
    s = 0
    for i in range (a,b+1):
        s += binom(n,p,i)
    return s
#calcul de p(5<=X<=12) pour B(43,0.2)
n,p,a,b = 43,0.2,5,12
val = binom_inter(n,p,a,b)
print("p(",a,"<=X<=",b,")=",val)
```

→on obtient : p(5 <=X<= 12)= 0.8761068491282018