DISTRIBUCION BINOMIAL

El siguiente programa permite calcular la distribución binomial acumulada generando la tabla correspondiente a partir de n > 0.

El programa esta desarrollado en lenguaje c a continuación se describe su funcionamiento.

Formula para calcular la distribución binomial

$$P[X = k] = \binom{n}{k} \cdot p^k \cdot (1 - p)^{n - k}$$

Formula para calcular combinaciones

$$C_k^n = \frac{n!}{k! (n-k)!}$$

En esta funcion es donde se realiza el calculo de forma recursiva

```
n es el número de pruebas.
k es el número de éxitos.
p es la probabilidad de éxito.
q es la probabilidad de fracaso
.
double b(long k, long n, double p){
    if ( k == 0) return pow(1.0 - p, n);
    return ( (double)(n - k + 1) / k * p / (1.0 - p) ) * b(k - 1, n, p);
}
```

Para realizar los cálculos y realizar la impresión correspondiente de la tabla de distribucion binomial acumulada se tiene la siguiente función.

La función recibe n como parámetro

void printfBinomial(long n){

```
printf("p | ");
```

Se tiene el ciclo para imprimir los intervalos:

```
for(i = 0; i != 14; i++){
    printf("%.3f ", array[i]);
}
printf("\n");
printf("\n");
```

Ciclo que genera la tabla de distribucion acumulada a partir de n

```
while (col != n+1){
    printf(" ");
    if (col<10)
        printf("%lu | ",col );
    else
        printf("%lu | ",col );
    for(i = 0; i != 14; i++){
        binomial[i] += b(col,n,array[i]);
        printf("%.3f ",binomial[i]);
    }
    col++;
    printf("\n");
}</pre>
```

Ejemplo tabla de distribucion acumulada con n = 10

Entrada n = 10

Salida:

```
    p | 0.010 0.050 0.100 0.200 0.250 0.300 0.400 0.500 0.600 0.700 0.750 0.800 0.900 0.950
    0 | 0.904 0.599 0.349 0.107 0.056 0.028 0.006 0.001 0.000 0.000 0.000 0.000 0.000 0.000
    1 | 0.996 0.914 0.736 0.376 0.244 0.149 0.046 0.011 0.002 0.000 0.000 0.000 0.000 0.000
    2 | 1.000 0.988 0.930 0.678 0.526 0.383 0.167 0.055 0.012 0.002 0.000 0.000 0.000 0.000
    3 | 1.000 0.999 0.987 0.879 0.776 0.650 0.382 0.172 0.055 0.011 0.004 0.001 0.000 0.000
    4 | 1.000 1.000 0.998 0.967 0.922 0.850 0.633 0.377 0.166 0.047 0.020 0.006 0.000 0.000
    5 | 1.000 1.000 1.000 0.994 0.980 0.953 0.834 0.623 0.367 0.150 0.078 0.033 0.002 0.000
    6 | 1.000 1.000 1.000 0.999 0.996 0.989 0.945 0.828 0.618 0.350 0.224 0.121 0.013 0.001
    7 | 1.000 1.000 1.000 1.000 1.000 1.000 0.998 0.988 0.945 0.833 0.617 0.474 0.322 0.070 0.012
    8 | 1.000 1.000 1.000 1.000 1.000 1.000 1.000 0.998 0.989 0.954 0.851 0.756 0.624 0.264 0.086
    9 | 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
    1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.0
```