

# Задание 1

## Цель

Подсчет числа единиц в двоичном представлении числа без массивов и операций деления

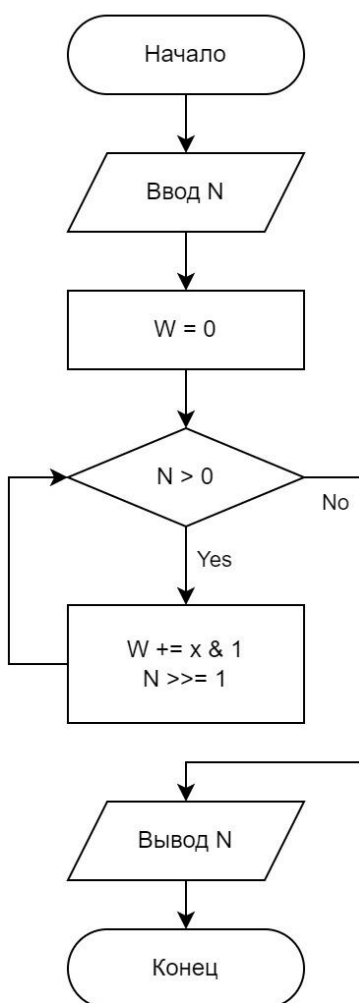
## Задачи

1. Написать программу, которая подсчитывает количество единиц в двоичной записи числа
2. Не использовать массивы и операции деления

## Описание рассматриваемого алгоритма

1. На вход функции подается число  $n$
2. Создается счетчик количества единиц  $w = 0$
3. Запускается цикл `while n > 0`, в котором:
  - а. С помощью побитовой операции умножения на маску  $1_2$  определяется младший бит числа, который прибавляется к счетчику
  - б. Происходит побитовый сдвиг исходного числа на 1 бит вправо
4. Функция возвращает полученное значение  $w$

## Описание программы в виде блок-схемы



# Описание результатов проводимых исследований

Результатом данной функции является количество единиц в двоичном представлении введенного числа

## Листинг программы

```
1 unsigned int count_ones (unsigned long long n) {
2     unsigned int w = 0;
3     while (n > 0) {
4         w += n & 1;
5         n = n >> 1;
6     }
7     return w;
8 }
```

## Задание 2

### Цель

Подсчет числа единиц в двоичном представлении числа с помощью вспомогательного массива

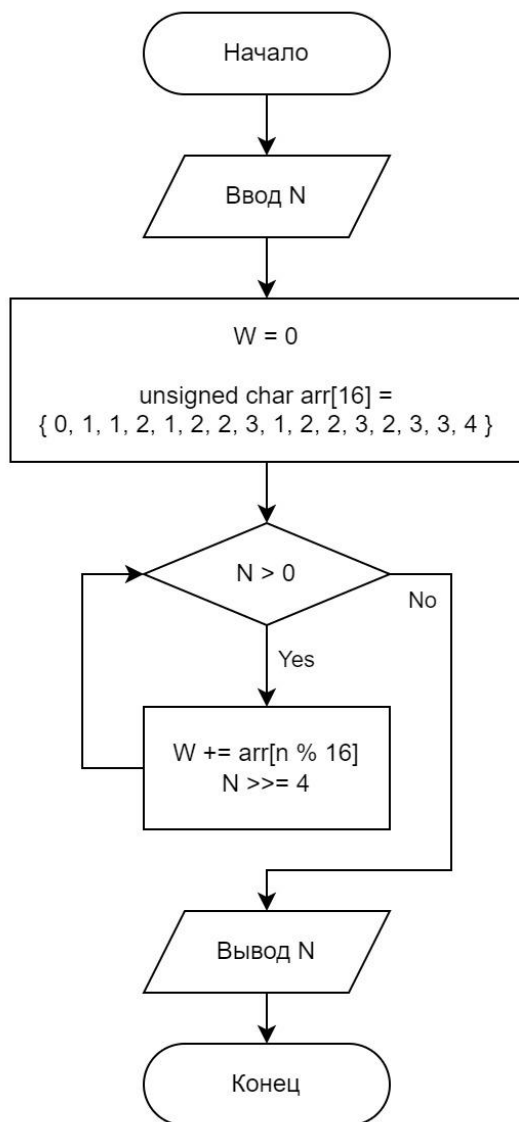
### Задачи

1. Написать программу, которая подсчитывает количество единиц в двоичной записи введенного числа
2. Использовать вспомогательный массив

### Описание рассматриваемого алгоритма

1. На вход функции подается число  $n$
2. Задается массив  $arr$  содержащий заранее рассчитанное количество единиц для чисел из промежутка  $[0; 15]$
3. Создается счетчик количества единиц  $w = 0$
4. Запускается цикл  $while\ n > 0$ , в котором:
  - а. С помощью побитовой операции умножения на маску  $15\ (1111_2)$  определяется четыре младших бита числа, которые рассматриваются как индекс массива
  - б. К счетчику прибавляется значение массива по этому индексу
  - в. Происходит побитовый сдвиг исходного числа на 4 бита вправо
5. Функция возвращает полученное значение  $w$

## Описание программы в виде блок-схемы



## Описание результатов проводимых исследований

Результатом данной функции является количество единиц в двоичном представлении введенного числа

## Листинг программы

```
1 unsigned char arr[16] = {
2     0, 1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 3, 2, 3, 3, 4
3 };
4
5 unsigned int count_ones (unsigned long long n) {
6     unsigned int w = 0;
7     while (n > 0) {
8         w += arr[n & 15];
9         n = n >> 4;
10    }
11    return w;
12 }
```

# Задание 3

## Цель

Статистика по количеству единиц в двоичном представлении чисел от 0 до  $2^k - 1$ , с помощью перебора и функции из 1 или 2 пункта

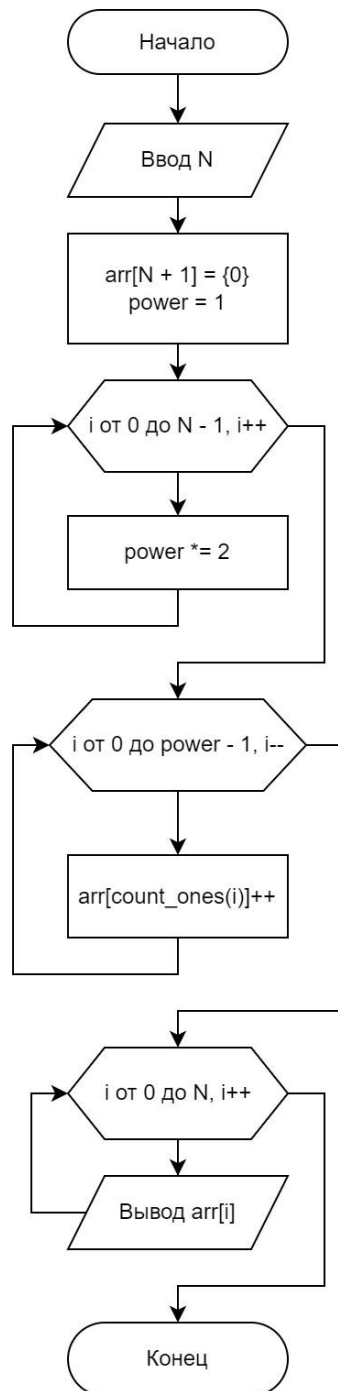
## Задачи

1. Написать программу, которая методом перебора подсчитывает статистику по количеству единиц в двоичном представлении чисел от 0 до  $2^k - 1$
2. Использовать функцию из задания 1 или 2

## Описание рассматриваемого алгоритма

1. На вход программы подается число  $k$
2. Создается массив `stats`, размера  $k + 1$ , изначально заполненный нулями, и объявляется переменная `power = 1`
3. В цикле от 0 до  $k - 1$ :
  - а. Переменная `power` умножается на 2
4. В цикле  $i$  от 0 до `power - 1`:
  - а. Инкремент элемента массива с индексом `count_ones(i)`
5. Полученный массив выводится на экран

## Описание программы в виде блок-схемы



## Описание результатов проводимых исследований

Результатом данной программы является вывод статистики по количеству чисел от 0 до  $2^k - 1$  для каждого возможного количества единиц от 0 до  $k$

## Листинг программы

```
1 #include "stdio.h"
2
3 unsigned char arr[16] = {0, 1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 3, 2, 3, 3, 4};
4
5 unsigned char count_ones (unsigned long long n) {
6     unsigned char w = 0;
7     while (n > 0) {
8         w += arr[n & 15];
9         n = n >> 4;
10    }
11    return w;
12 }
13
14 int main() {
15     int k;
16     scanf("%d", &k);
17     int stats[k + 1] = {0};
18
19     unsigned long long power = 1;
20     for (int i = 0; i < k; i++)
21         power *= 2;
22
23     for (unsigned long long i = 0; i < power; i++)
24         stats[count_ones(i)]++;
25
26     for (int i = 0; i <= k; i++)
27         printf("with %d ones: %d\n", i, stats[i]);
28 }
```

## Задание 4

### Цель

Статистика по количеству единиц в двоичном представлении чисел от 0 до  $2^k - 1$ , с помощью численного расчета

### Задачи

Написать программу, которая подсчитывает статистику по количеству единиц в двоичном представлении чисел от 0 до  $2^k - 1$  с помощью численного расчета

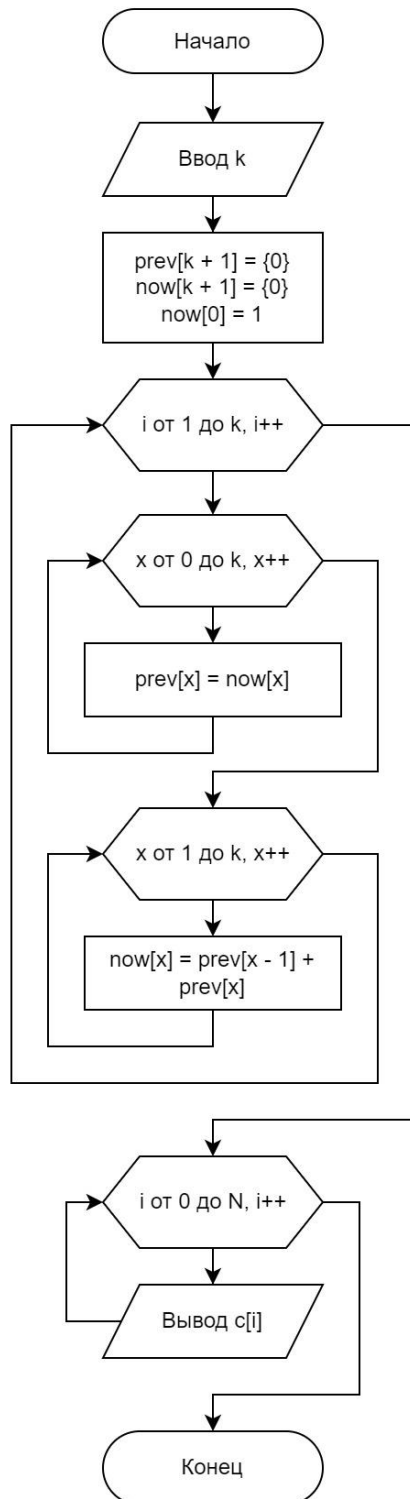
Использовать функцию вычисления биномиального коэффициента или треугольник Паскаля

### Описание рассматриваемого алгоритма

1. На вход программы подается число  $k$
2. Создается два массива: `prev` и `pow`, длины  $k + 1$ , заполненных нулями
3. Нулевой элемент массива `pow` задается равным 1

4. В цикле  $i$  от 0 до  $k$ :
  - а. Копирование всех элементов из  $now$  в  $prev$
  - б. В цикле  $x$  от 1 до  $k$ :
    - і. Элемент массива  $now$  с индексом  $x$  становится равен сумме элементов массива  $prev$  с индексами  $x$  и  $x - 1$
5. Массив  $now$  выводится на экран

## Описание программы в виде блок-схемы



## Описание результатов проводимых исследований

Результатом данной программы является вывод статистики по количеству чисел от 0 до  $2^k - 1$  для каждого возможного количества единиц от 0 до  $k$ , вычисленной с помощью треугольника Паскаля

## Листинг программы

```
1 #include "stdio.h"
2
3 int main() {
4     int k;
5     scanf("%d", &k);
6
7     unsigned long long prev[k + 1] = {0};
8     unsigned long long now[k + 1] = {0};
9     now[0] = 1;
10
11     for (int i = 1; i <= k; i++) {
12         for (int x = 0; x <= k; x++)
13             prev[x] = now[x];
14         for (int x = 1; x <= k; x++)
15             now[x] = prev[x - 1] + prev[x];
16     }
17
18     for (int i = 0; i <= k; i++)
19         printf("%llu\n", now[i]);
20 }
```