

## Пермутации от $k$ елемента

Вариациите без повторение от  $k$  елемента от  $k$ -ти клас ще наричаме *пермутации от  $k$  елемента*, т.е. това са такива редици от по  $k$  елемента, във всяка от които влизат всички елементи. Редиците се различават само по местата на елементите.

*Комбинации от  $k$  елемента от  $n$ -ти клас*

*Дефиниция.* Ако  $v_1, v_2, \dots, v_k$  са  $k$  на брой различни елементи ( $k \geq 1$ ), а  $n$  е дадено естествено число ( $n \leq k$ ), всяка редица от точно  $n$  елемента от дадените, като всеки елемент може да участва не повече от веднъж и мястото на елемента в редицата не е от значение, се нарича *комбинация от  $k$ -елемента от  $n$ -ти клас*.

*Пример:* Ако  $k = 4$ ,  $v_1 = 1$ ,  $v_2 = 2$ ,  $v_3 = 3$ ,  $v_4 = 4$ , а  $n = 2$ , всички комбинации от тези елементи от 2-ри клас се получават от вариациите с повторение от 1,2,3 и 4 от 2-ри клас

(1,1) (2,1) (3,1) (4,1)

**(1,2)** (2,2) (3,2) (4,2)

**(1,3)** **(2,3)** **(3,3)** (4,3)

**(1,4)** **(2,4)** **(3,4)** (4,4)

се изберат редиците с растящи стойности например. Означени са удебелено и наклонено (Редиците (1,2) и (2,1);(1,4) и (4,1) и т.н. са еквивалентни).

**Задача 4.** (комбинации) Да се напише програма, която намира всички комбинации от елементите  $0, 1, \dots, k - 1$  от  $n$ -ти клас ;  $n \leq k$ .

*анализ:*

Трябва да бъдат генерирани всички вариации с повторение от елементите  $0, 1, \dots, k - 1$  и се изведат онези, които образуват например строго монотонно растящи редици.

За целта може да се използва функцията *bool mon( )*, която установява дали дадена редица от цели числа е строго монотонно растяща.

```
bool mon(size_t* a, int n)
{
    for (int i = 1; i <= n; i++)
    {
        int curr = a[i];
        for (int j = i + 1; j <= n; j++) if (a[i] >= a[j]) return false;
    }
    return true;
}
```