

Задача J. Последовательности

Рассмотрим последовательности чисел. Первая последовательность состоит из одного числа K. Каждая следующая последовательность чисел описывает предыдущую по такому правилу. Просматриваем описываемую последовательность слева направо и разбиваем на отрезки, состоящие из подряд идущих равных чисел (причем все идущие подряд одинаковые числа всегда объединяем в один отрезок). Далее каждый такой отрезок описываем двумя числами — первое число говорит, сколько раз повторяется одно и то же число, второе число говорит, какое именно число повторяется. Записываем эти пары последовательно в соответствии с отрезками слева направо, и получаем новую последовательность (см. примеры ниже).

Например, для K=2 последовательности получатся такими:

№ Последовательность Как ее читать (слова в описании соответствуют числам текущей последовательности слева направо, и описывают предыдущую последовательность)

1 2 Исходная последовательность

2 1 2 Одна «двойка»

3 1 1 1 2 Одна «единица», одна «двойка»

4 3 1 1 2 Три «единицы», одна «двойка»

5 1 3 2 1 1 2 Одна «тройка», две «единицы», одна «двойка»

6 1 1 1 3 1 2 2 1 1 2 Одна «единица», одна «тройка», одна «двойка», две «единицы», одна «двойка»

Напишите программу, которая по исходному числу K напечатает N-ую получающуюся последовательность.

Входные данные

Вводится число K (1 ≤ K ≤ 9) и число N (1 ≤ N ≤ 15).

Выходные данные

Ваша программа должна печатать N-ую последовательность, полученную из начальной последовательности, состоящей из одного числа K.

Числа при выводе следует разделять пробелами.

Примеры	1 1 1 3 1 2 2 1 1 2
входные данные	
2	3 1 1 3 1 1 2 2 2 1 1 2
6	

выходные данные	
1 1 1 3 1 2 2 1 1 2	

входные данные	
2	
1	

выходные данные	
2	

входные данные	
1	
3	

выходные данные	
2 1	



```
int massiv1[1000000] = {};  
int massiv2 [1000000] = {};  
void povtor (int k, int n)  
{  
    massiv1 [0] = k;  
  
    int t2;  
    int m;  
    for (int i = 0; i < n; i++)  
    {  
        t2 = 0;  
        m = 1;  
        for (int j = 0; massiv1[j] != 0; j++)  
        {  
            if (massiv1[j] == massiv1[j + 1])  
            {  
                m++;  
            }  
            else  
            {  
                massiv2[t2] = m;  
                massiv2[t2 + 1] = massiv1 [j];  
                t2 = t2 + 2;  
                m = 1;  
            }  
        }  
        cout << endl;  
        risov(massiv2, t2);  
        m = 0;  
        for (int j = 0; massiv2[j] != 0; j++)  
        {  
            if(massiv2[j] < 10)  
            {  
                massiv1[m] = massiv2[j];  
                m++;  
            }  
            else  
            {  
                while(massiv2[j] != 0)  
                {  
                    massiv1[m] = massiv2[j]%10;  
                    massiv2[j] = massiv2[j] / 10;  
                    m++;  
                }  
            }  
        }  
        risov(massiv1, m);  
    }  
}
```