

## Задача D1. ТРИЪГЪЛНИЦИ

В тази задача ще разглеждаме текстове, оформени като “равностранни” триъгълници, съставени от нули и единици, като първите ще наричаме 0-триъгълници, а вторите – 1-триъгълници. Всеки триъгълник има дължина на страната, определена от някакво цяло положително число  $N$ , по-голямо или равно на три и не по-голямо от 20. Първият ред на триъгълника се състои от една единствена цифра (0 или 1, в зависимост от вида му), вторият ред – от две цифри от съответния вид, разделени с един интервал, който е точно под цифрата от първия ред. Третият ред се състои от три цифри от съответния вид, разделени с по един интервал, като интервалите са точно под цифрите от втория ред и т.н., а последният ред е съставен от  $N$  цифри от съответния вид, разделени с по един интервал. 0-триъгълниците се изобразяват изправени – първият ред (с една цифра) отгоре, а последният ред (с  $N$  цифри отдолу), докато 1-триъгълниците се изобразяват обърнати – редът с една цифра е отдолу, а този с  $N$  цифри – отгоре. Задачата е да направите програма **TREE.EXE**, която по зададени цели положителни  $N$  и  $M$  отпечата следната конструкция: два 0-триъгълника с дължина на страните  $N$  са долепени един до друг в основите си (страните с по  $N$  цифри са една до друга, разделени с един интервал), а 1-триъгълник със страна  $M$  е спуснат в образувалото се между двата 0-триъгълника пространство (виж примерите).

Програмата трябва да прочете числата  $N$  и  $M$  от клавиатурата (**стандартния вход**), а да изведе резултата на екрана (**стандартния изход**), като най-лявата цифра на конструкцията трябва да бъде в първа позиция на съответния ред на екрана.

### ПРИМЕР 1

**Вход**

5 7

**Изход**

```
1 1 1 1 1 1 1
 1 1 1 1 1 1
  1 1 1 1 1
   0 1 1 1 1 0
    0 0 1 1 1 0 0
     0 0 0 1 1 0 0 0
      0 0 0 0 1 0 0 0 0
       0 0 0 0 0 0 0 0 0 0
```

### ПРИМЕР 2

**Вход**

5 3

**Изход**

```
      0              0
    0 0 1 1 1 0 0
  0 0 0 1 1 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0
```