**Упражнения: Loops**

Допълнителни задачи за упражнение в [“Programming Fundamentals” course @ SoftUni](https://softuni.bg/courses/programming-fundamentals).  
Тези задачи не са задължителни и не се броят към домашните за курса.

**1. Christmas Tree**

Напишете програма, която чете число **n** (1 ≤ **n** ≤ 100), въведено от потребителя, и печата **коледна елха** с размер **n** като в примерите по-долу:

**Примерен вход и изход**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 1 | |  \* | \* | 2 | |  \* | \*  \*\* | \*\* | 3 | |  \* | \*  \*\* | \*\*  \*\*\* | \*\*\* | 4 | |  \* | \*  \*\* | \*\*  \*\*\* | \*\*\*  \*\*\*\* | \*\*\*\* |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/155#6>.

**Подсказки**:

* В цикъл за **i** от **0** до **n** печатайте (за лявата част на елхата):
  + **n-i** интервала; **n** звездички; вертикална черта.
* Аналогично довършете дясната част на елхата.

**2. Butterfly**

*Пета задача от междинния изпит на 26 март 2016. Тествайте решението си* ***[тук](https://judge.softuni.bg/Contests/Practice/Index/179" \l "4)****.*

Да се напише програма, която прочита от конзолата **цяло число** **n**, въведено от потребителя, и чертае **пеперуда** с ширина **2 \* n - 1** **колони** и височина **2 \* (n - 2) + 1 реда** като в примерите по-долу. **Лявата** и **дясната** ѝ **част** са **широки** **n - 1**.

**Вход**

Входът е **цяло число** **n** в интервала [**3**…**1000**].

**Изход**

Да се отпечатат на конзолата **2 \* (n - 2) + 1** текстови реда, изобразяващи **пеперудата**.

**Примерен вход и изход**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **вход** | **изход** | **вход** | **изход** |
| 3 | \*\ /\*  @  \*/ \\* | 5 | \*\*\*\ /\*\*\*  ---\ /---  \*\*\*\ /\*\*\*  @  \*\*\*/ \\*\*\*  ---/ \---  \*\*\*/ \\*\*\* | 7 | \*\*\*\*\*\ /\*\*\*\*\*  -----\ /-----  \*\*\*\*\*\ /\*\*\*\*\*  -----\ /-----  \*\*\*\*\*\ /\*\*\*\*\*  @  \*\*\*\*\*/ \\*\*\*\*\*  -----/ \-----  \*\*\*\*\*/ \\*\*\*\*\*  -----/ \-----  \*\*\*\*\*/ \\*\*\*\*\* |

**3. Greatest Common Divisor (CGD)**

Напишете програма, която чете две цели положителни числа **a** и **b**, въведени от потребителя, и изчислява и отпечатва **най-големият им общ делител (НОД)**.

**Примерен вход и изход**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 24  16 | 8 | 67  18 | 1 | 15  9 | 3 | 100  88 | 4 | 10  10 | 10 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/156#6>.

**Подсказка**: имплементирайте **алгоритъма на Евклид**: [https://bg.wikipedia.org/wiki/алгоритъм-на-Евклид](https://bg.wikipedia.org/wiki/алгоритъм-на-Евклид%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D1%8A%D0%BC_%D0%BD%D0%B0_%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4).

**4. Number Pyramid**

Напишете програма, която чете цяло число **n**, въведено от потребителя, и отпечатва **пирамида от числа** като в примерите:

**Примерен вход и изход**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 7 | 1  2 3  4 5 6  7 | 10 | 1  2 3  4 5 6  7 8 9 10 | 12 | 1  2 3  4 5 6  7 8 9 10  11 12 | 15 | 1  2 3  4 5 6  7 8 9 10  11 12 13 14 15 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/156#12>.

**Подсказка**:

* С **два вложени цикъла** печатайте пирамида от числа: на първия ред едно число, на втория ред 2 числа, на третия ред 3 числа и т.н.
* В отделен **брояч** пазете колко числа сте отпечатали до момента (и кое е текущото число). Когато стигнете **n**, излезте внимателно от двата вложени цикъла с **break** или **return**.

**5. Diamond**

Напишете програма, която чете цяло число **n** (1 ≤ **n** ≤ 100), въведено от потребителя, и печата диамант с размер **n** като в примерите по-долу:

**Примерен вход и изход**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 1 | \* | 2 | \*\* | 3 | -\*-  \*-\*  -\*- | 4 | -\*\*-  \*--\*  -\*\*- | 5 | --\*--  -\*-\*-  \*---\*  -\*-\*-  --\*-- |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 6 | --\*\*--  -\*--\*-  \*----\*  -\*--\*-  --\*\*-- | 7 | ---\*---  --\*-\*--  -\*---\*-  \*-----\*  -\*---\*-  --\*-\*--  ---\*--- | 8 | ---\*\*---  --\*--\*--  -\*----\*-  \*------\*  -\*----\*-  --\*--\*--  ---\*\*--- | 9 | ----\*----  ---\*-\*---  --\*---\*--  -\*-----\*-  \*-------\*  -\*-----\*-  --\*---\*--  ---\*-\*---  ----\*---- |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/155#9>.

**6. Number Table**

Напишете програма, която чете цяло число **n**, въведено от потребителя, и отпечатва **таблица (матрица) от числа** като в примерите:

**Примерен вход и изход**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2 | 1 2  2 1 | 3 | 1 2 3  2 3 2  3 2 1 | 4 | 1 2 3 4  2 3 4 3  3 4 3 2  4 3 2 1 | 5 | 1 2 3 4 5  2 3 4 5 4  3 4 5 4 3  4 5 4 3 2  5 4 3 2 1 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/156#13>.