## Диплома от НБУ

Все още не сте чули за статистиката, която гласи, че **40 от 300 студента** взимат **диплома** и сте ентусиазирани да проверите колко трудно може да бъде.

За да получите диплома ви трябват **240 кредита**.

Напишете програма, която прочита **име на студент (текст**), брой курсове (цяло число), и след това n на брой редове с **кредити от курсове (цяло число)**, където **n e броя на курсовете**.

Накрая проверете дали студента е успял да вземе диплома.

Ако студент има **240 или повече** кредита, то **принтирайте**:

`{име на студента} gets diploma from NBU with {брой кредити}`

В **противен случай**, принтирайте:

`As expected, {име на студента} started a job and left NBU behind. {брой кредита нужни до диплома} credits needed to graduate.`

**Примерен вход и изход**

|  |  |
| --- | --- |
| **вход** | **изход** |
| Dian  7  10  12  10  3  5  7  2 | 0 |

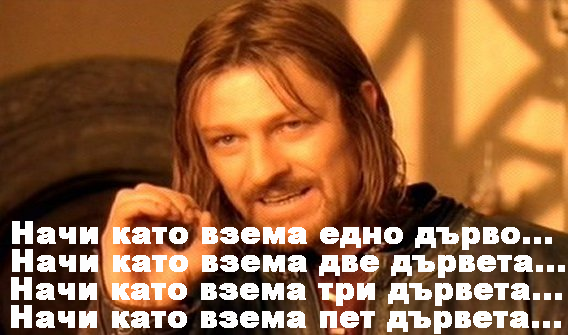
## Фибоначи

Напишете програма, която чете **положително число n** и генерира **числа на Фибоначи**, докато те не станат **по-големи от n.**

Числата на Фибоначи се образуват по следната формула:

Първите две числа винаги са **0 и 1**, всяко следващо число е сбор на предишните две.

Пример: 0, 1, 1, 2, 3, 5, 8, 13, 21….



**Примерен вход и изход**

|  |  |
| --- | --- |
| **вход** | **изход** |
| 217 | 0  1  1  2  3  5  8  13  21  34  55  89  144 |

|  |  |
| --- | --- |
| **вход** | **изход** |
| 63245987 | 0  1  1  2  3  5  8  13  21  34  55  89  144  233  377  610  987  1597  2584  4181  6765  10946  17711  28657  46368  75025  121393  196418  317811  514229  832040  1346269  2178309  3524578  5702887  9227465  14930352  24157817  39088169  63245986 |

## Диамант

Харесали сте най-готината колежка/колега в курса ви, като програмисти нямате много добри социални умения и затова решавате, че ще й/му подарите подарък.

Plot twist: освен, че нямате комуникативни умения, нямате и пари, а знаете, че той/тя харесва **диаманти**. Затова решавате да направите програма, която приема **цяло число** и принтира **диамант** със следните **размери**.

**Примерен вход и изход**

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

**Подсказки**:

* Разделете ромба на горна и долна част и ги печатайте с два отделни цикъла.
* За **горната част** завъртете цикъл за row от 1 то n:
  + Отпечатайте n-row интервала.
  + Отпечатайте “\*”.
  + Отпечатайте row-1 пъти “ \*”.
* **Долната част** отпечатайте аналогично на горната с цикъл от 1 до n-1.

## Слънчеви очила

Оказва се, че диамантите не са нещото на вашата бъдеща половинка. Минус точки за вас, вече изобщо не ви харесват и за да излезете от депресията, решавате да си направите розови очила, с които гледайки се в огледалото, уверено да се самозалъгвате, че не вие сте проблема.

Напишете програма, която чете цяло число n (3 ≤ n ≤ 100), въведено от потребителя, и печата **слънчеви очила** с размер 5\*n x n като в примерите:

**Примерен вход и изход**

|  |  |
| --- | --- |
| **вход** | **изход** |
| 3 | \*\*\*\*\*\* \*\*\*\*\*\*  \*////\*|||\*////\*  \*\*\*\*\*\* \*\*\*\*\*\* |
| 4 | \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*  \*//////\*||||\*//////\*  \*//////\* \*//////\*  \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* |
| 5 | \*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*  \*////////\* \*////////\*  \*////////\*|||||\*////////\*  \*////////\* \*////////\*  \*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\* |

**Подсказки**:

* Отпечатайте **най-горния ред** от очилата:
  + 2\*n звездички; n интервала; 2\*n звездички
* Отпечатайте **средните** **n-2** **реда**:
  + звездичка; 2\*n-2 наклонени черти; звездичка; n интервала; звездичка; 2\*n-2 наклонени черти; звездичка
  + когато редът е (n-1) / 2 - 1, печатайте n вертикални черти вместо n интервала
* Отпечатайте **най-долния ред** от очилата:
  + 2\*n звездички; n интервала; 2\*n звездички