## Числата от 1 до N през 3

Напишете програма, която въвежда число n и отпечатва **числата от 1 до** n **през 3** (със стъпка 3). Примери:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 10 | 1  4  7  10 | 7 | 1  4  7 | 15 | 1  4  7  10  13 |

## Числата от N до 1 в обратен ред

Напишете програма, която въвежда цяло положително число n и печата **числата от** n **до** **1** **в обратен ред** (от най-голямото към най-малкото). Примери:

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

## Числа от 1 до 2n

Напишете програма, която чете от конзолата цяло число n и **печата числата от 1 до 2n**. Примери:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 3 | 1  2  4  8 | 4 | 1  2  4  8  16 | 5 | 1  2  4  8  16  32 |

## Четни степени на 2

Да се напише програма, която въвежда n и **печата четните степени на 2** **≤** **2n**: **20**, **22**, **24**, **28**, …, **2n**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 3 | 1  4 | 4 | 1  4  16 | 5 | 1  4  16 | 6 | 1  4  16  64 | 7 | 1  4  16  64 |

## Редица числа 2k+1

Напишете програма, която въвежда число nи отпечатва **всички числа ≤ n от редицата**: 1, 3, 7, 15, 31, …. Всяко следващо число се изчислява като **предишното число \* 2 + 1**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 3 | 1  3 | 8 | 1  3  7 | 17 | 1  3  7  15 | 31 | 1  3  7  15  31 |

## Число в диапазона [1…100]

Напишете програма, която въвежда цяло положително **число** n **в диапазона [1…100]**. При въвеждане на число извън посочения диапазон, да се отпечата съобщение за грешка и потребителят да се подкани **да въведе ново число**. Примери:

|  |
| --- |
| **вход / изход** |
| Еnter a number in the range [1...100]: **35**  The number is: 35 |
| Еnter a number in the range [1...100]: **105**  Invalid number!  Еnter a number in the range [1...100]: **0**  Invalid number!  Еnter a number in the range [1...100]: **-200**  Invalid number!  Еnter a number in the range [1...100]: **77**  The number is: 77 |

## Най-голям общ делител (НОД)

Напишете програма, която въвежда две цели положителни числа a и **b** и изчислява и отпечатва **най-големият им общ делител (НОД)**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

**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).

## Факториел

Напишете програма, която въвежда цяло число n (1 ≤ n ≤ 12) и **изчислява и отпечатва** **n!** = 1 \* 2 \* … \* **n** (**n** **факториел**). Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 5 | 120 | 6 | 720 | 10 | 3628800 | 1 | 1 | 2 | 2 |

## Сумиране на цифрите на число

Напишете програма, която въвежда цяло число num и отпечатва **сумата от цифрите му**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **коментар** |  | **вход** | **изход** | **коментар** |  | **вход** | **изход** |  | **вход** | **изход** |
| 5634 | 18 | 6+6+3+4 = 18 | 19 | 10 | 1+9 = 10 | 5 | 5 | 17151 | 15 |

**Подсказка**: в цикъл докато не стигнете до 0 сумирайте последната цифра на числото (num % 10) и го разделяйте след това на 10 (така изтривате последната му цифра).

## Проверка за просто число

Напишете програма, която въвежда цяло число n и **проверява дали е** **просто число** (дали се дели само на себе си и на единица). Да се отпечата “**Prime**” или “**Not prime**”. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2 | Prime | 3 | Prime | 4 | Not Prime | 5 | Prime | 7 | Prime |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 1 | Not Prime | 0 | Not Prime | -1 | Not Prime | 149 | Prime |

**Подсказки**:

* Ако числото е по-малко от 2, значи не е просто.
* Всяко друго число първоначално се приема за **просто** и се проверява в цикъл дали се дели на числата 2, 3, 4, …, до корен квадратен от n (цялата част).

## Въвеждане на четно число (с обработка на грешен вход)

Напишете програма, която **въвежда четно число**. Ако потребителят въведе **грешно число** (нечетно число или стринг, който не е цяло число), трябва да му излиза **съобщение за** **грешка** и да **въвежда отново**. Примери:

|  |
| --- |
| **вход / изход** |
| Enter even number: **34**  Even number entered: 35 |
| Enter even number: **35**  The number is not even.  Enter even number: **hello**  Invalid number!  Enter even number: **12.85**  Invalid number!  Enter even number: **3464232636536513**  Invalid number!  Enter even number: **8**  Even number entered: 8 |

**Подсказки**:

* В цикъл въвеждайте число и проверявайте дали е четно. При коректно число излезте от цикъла.
* С try **{ … }** catch **{ … }** конструкция прихванете грешните числа, които не могат да се обърнат в int.

## Числа на Фибоначи

Напишете програма, която въвежда цяло число **n** и пресмята **n-тото** **число на Фибоначи**. Нулевото число на Фибоначи е 1, първото е също 1, а всяко следващо е сумата от предходните две. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 0 | 1 | 1 | 1 | 2 | 2 | 5 | 8 | 10 | 89 |

**Подсказка**:

* При **n < 2** отпечатайте **1**.
* Започнете от **f0**=**1** и **f1**=**1** и в цикъл сумирайте последните две числа. Записвайте последните две числа след всяка стъпка в **f0** и **f1**.

## Пирамида от числа

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

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

**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.

## Таблица с числа

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

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

**Подсказка**:

* С **два вложени цикъла** за **row** (ред) и **col** (колона) печатайте число по формулата **num** = **row + col + 1**.
* За долната дясна половина на таблицата ще се получат грешни резултати. Там използвайте формулата **2\*n - num**.
* Как се сетихме за тези формули? Математическа досетливост: наблюдаваме числата, предполагаме каква е формулата, тестваме и ако не се получи, измисляме друга формула и пак пробваме. В случая имаме различни формули за горната лява и долната дясна половина от матрицата.

## Генератор за тъпи пароли

Да се напише програма, която въвежда две цели числа *n* и *l* и генерира по азбучен ред всички възможни **“тъпи” пароли**, които се състоят от следните **5 символа**:

* Символ 1: цифра от 1 до *n*.
* Символ 2: цифра от 1 до *n*.
* Символ 3: малка буква измежду първите *l* букви на латинската азбука.
* Символ 4: малка буква измежду първите *l* букви на латинската азбука.
* Символ 5: цифра от 1 до n, по-голяма от първите 2 цифри.

### Вход

Входът се чете от конзолата и се състои от две **цели числа** *n* и *l* в интервала [**1**…**9**], по едно на ред.

### Изход

На конзолата трябва да се отпечатат **всички “тъпи” пароли** по **азбучен ред**, разделени с **интервал**.

### Примерен вход и изход

|  |  |
| --- | --- |
| **вход** | **изход** |
| 2  4 | 11aa2 11ab2 11ac2 11ad2 11ba2 11bb2 11bc2 11bd2 11ca2 11cb2 11cc2 11cd2 11da2 11db2 11dc2 11dd2 |
| 3  1 | 11aa2 11aa3 12aa3 21aa3 22aa3 |
| 3  2 | 11aa2 11aa3 11ab2 11ab3 11ba2 11ba3 11bb2 11bb3 12aa3 12ab3 12ba3 12bb3 21aa3 21ab3 21ba3 21bb3 22aa3 22ab3 22ba3 22bb3 |
| 4  2 | 11aa2 11aa3 11aa4 11ab2 11ab3 11ab4 11ba2 11ba3 11ba4 11bb2 11bb3 11bb4 12aa3 12aa4 12ab3 12ab4 12ba3 12ba4 12bb3 12bb4 13aa4 13ab4 13ba4 13bb4 21aa3 21aa4 21ab3 21ab4 21ba3 21ba4 21bb3 21bb4 22aa3 22aa4 22ab3 22ab4 22ba3 22ba4 22bb3 22bb4 23aa4 23ab4 23ba4 23bb4 31aa4 31ab4 31ba4 31bb4 32aa4 32ab4 32ba4 32bb4 33aa4 33ab4 33ba4 33bb4 |