

Упражнения: Работа с по-сложни цикли

Задачи за упражнение в клас и за домашно към курса „[Основи на програмирането](#)“ @ СофтУни.

1. 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).

Примерен вход и изход:

вход	изход	вход	изход	вход	изход
10	1 4 7 10	7	1 4 7	15	1 4 7 10 13

Подсказки:

1. Създайте **нов файл**. Задайте подходящо име на файла, например “LoopByStep3”.
2. Можете да завъртите **for-цикъл със стъпка 3** по следния начин: `for i in range(1, n + 1, 3)`.
3. Отидете в началото на **файла** и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:

```
for i in range(1, n + 1, 3):  
    print(i)
```

4. **Стартирайте** програмата с [Alt+Shift+F10] и я **тествайте**.
5. **Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/156#0> . Трябва да получите **100 точки** (напълно коректно решение).

2. 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).

Примерен вход и изход:

вход	изход	вход	изход	вход	изход
2	2 1	3	3 2 1	5	5 4 3 2 1

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/156#1> .

3. Числа от 2^0 до 2^n

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

Примерен вход и изход:

вход	изход
3	1 2 4 8

вход	изход
4	1 2 4 8 16

вход	Изход
5	1 2 4 8 16 32

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/156#2>.

Подсказка: завъртете for-цикъл от 0 до n и започвайки от num = 1 на всяка стъпка умножавайте num по 2.

4. Четни степени на 2

Да се напише програма, която въвежда n и печата четните степени на 2 $2 \leq 2^n$: $2^0, 2^2, 2^4, 2^8, \dots, 2^n$.

Примерен вход и изход:

вход	изход
3	1 4

вход	изход
4	1 4 16

вход	изход
5	1 4 16

вход	изход
6	1 4 16 64

вход	изход
7	1 4 16 64

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/156#3>.

Подсказка: започнете от 1 и в цикъл умножавайте по 4 на всяка стъпка.

5. Редица числа $2k+1$

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

Примерен вход и изход:

вход	изход
3	1 3

вход	изход
8	1 3 7

вход	изход
17	1 3 7 15

вход	изход
31	1 3 7 15 31

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/156#4>.

Подсказки:

- Започнете от num = 1.
- В цикъл докато num не стигне n, печатайте, го умножавайте по 2 и прибавяйте 1.

6. 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.



Примерен вход и изход:

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

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/156#5>.

Подсказки:

- Въведете число.
- Повтаряйте в цикъл докато числото е невалидно: отпечатайте грешка и въведете число отново.

7. 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)**.

Примерен вход и изход:

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

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

Подсказка: имплементирайте [алгоритъма на Евклид](#).

8. Факториел

Напишете програма, която въвежда цяло число **n** ($1 \leq n \leq 12$) и изчислява и отпечатва $n! = 1 * 2 * \dots * n$ (**n факториел**).

Примерен вход и изход:

вход	изход	вход	изход	вход	изход	вход	изход	вход	изход
5	120	6	720	10	3628800	1	1	2	2

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/156#7>.

Подсказка: в цикъл умножете числата от 1 до n.

9. Сумиране на цифрите на число

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



Примерен вход и изход:

вход	изход	коментар
5634	18	$6+6+3+4 = 18$

вход	изход	коментар
19	10	$1+9 = 10$

вход	изход
5	5

вход	изход
17151	15

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/156#8>.

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

10. 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".

Примерен вход и изход:

вход	изход
2	Prime

вход	изход
3	Prime

вход	изход
4	Not Prime

вход	изход
5	Prime

вход	изход
7	Prime

вход	изход
1	Not Prime

вход	изход
0	Not Prime

вход	изход
-1	Not Prime

вход	изход
149	Prime

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/156#9>.

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).

11. Въвеждане на четно число

Напишете програма, която **въвежда четно число**. Ако потребителят въведе **грешно число**, трябва да се въвежда ново такова.

Примерен вход и изход:

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

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/156#10>.

Подсказки:



- В цикъл въвеждайте число и проверявайте дали е четно. При коректно число излезте от цикъла.

12. 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.

Примерен вход и изход:

вход	изход
0	1

вход	изход
1	1

вход	изход
2	2

вход	изход
5	8

вход	изход
10	89

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/156#11>.

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**.

13. Пирамида от числа

Напишете програма, която въвежда цяло число **n** и отпечатва **пирамида от числа**.

Примерен вход и изход:

вход	изход
7	1 2 3 4 5 6 7

вход	изход
10	1 2 3 4 5 6 7 8 9 10

вход	изход
12	1 2 3 4 5 6 7 8 9 10 11 12

вход	изход
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/492#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**.

14. Таблица с числа

Напишете програма, която въвежда цяло число **n** и отпечатва **таблица (матрица) от числа**.

Примерен вход и изход:

вход	изход
2	1 2 2 1

вход	изход
3	1 2 3 2 3 2 3 2 1

вход	изход
4	1 2 3 4 2 3 4 3 3 4 3 2 4 3 2 1

вход	изход
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/492#13>.

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**.

Изпитни задачи от минали издания на курса

15. Генератор за тъпи пароли

Шеста задача от изпита на 6 март 2016. Тествайте решението си [тук](#).

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 отпечатаат **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

16. Магически числа

Шеста задача от изпита на 26 март 2016. Тествайте решението си [тук](#).

Да се напише програма, която въвежда едно цяло „магическо“ число и изкарва **всички** възможни **6-цифрени числа**, за които **произведението на неговите цифри е равно** на „магическото“ число.

Пример: „Магическо число“ -> 2

- 111112 -> 1 * 1 * 1 * 1 * 1 * 2 = 2



- $111121 \rightarrow 1 * 1 * 1 * 1 * 2 * 1 = 2$
- $111211 \rightarrow 1 * 1 * 1 * 2 * 1 * 1 = 2$
- $112111 \rightarrow 1 * 1 * 2 * 1 * 1 * 1 = 2$
- $121111 \rightarrow 1 * 2 * 1 * 1 * 1 * 1 = 2$
- $211111 \rightarrow 2 * 1 * 1 * 1 * 1 * 1 = 2$

Вход

Входът се чете от конзолата и се състои от **едно цяло число** в интервала $[1...600000]$.

Изход

На конзолата трябва да се отпечатат **всички “магически” числа**, разделени с **интервал**.

Примерен вход и изход

вход	изход
2	111112 111121 111211 112111 121111 211111
8	111118 111124 111142 111181 111214 111222 111241 111412 111421 111811 112114 112122 112122 112141 112212 112221 112411 114112 114121 114211 118111 121114 121122 121141 121212 121221 121411 122112 122121 122211 124111 141112 141121 141211 142111 181111 211114 211122 211141 211212 211221 211411 212112 212121 212211 214111 221112 221121 221211 222111 241111 411112 411121 411211 412111 421111 811111
531441	999999

17. Специални числа

Шеста задача от изпита на 24 април 2016. Тествайте решението си [тук](#).

Да се напише програма, която **въвежда едно цяло число N** и генерира всички възможни **“специални” числа** от **1111** до **9999**. За да бъде **“специално”** едно число, то трябва да отговаря на **следното условие**:

- **N** да се дели на всяка една от неговите цифри без остатък.

Пример: при **N = 16**, **2418** е специално число:

- $16 / 2 = 8$ без остатък
- $16 / 4 = 4$ без остатък
- $16 / 1 = 16$ без остатък
- $16 / 8 = 2$ без остатък

Вход

Входът се чете от конзолата и се състои от **едно цяло число** в интервала $[1...600000]$

Изход

На конзолата трябва да се отпечатат **всички “специални” числа**, разделени с **интервал**

Примерен вход и изход

вход	изход	коментари
3	1111 1113 1131 1133 1311 1313 1331 1333 3111 3113 3131 3133 3311 3313 3331 3333	3 / 1 = 3 без остатък 3 / 3 = 1 без остатък 3 / 3 = 1 без остатък

															3 / 3 = 1 без остатък		
11	1111																
16	1111	1112	1114	1118	1121	1122	1124	1128	1141	1142	1144	1148	1181	1182	1184	1188	
	1211	1212	1214	1218	1221	1222	1224	1228	1241	1242	1244	1248	1281	1282	1284	1288	
	1411	1412	1414	1418	1421	1422	1424	1428	1441	1442	1444	1448	1481	1482	1484	1488	
	1811	1812	1814	1818	1821	1822	1824	1828	1841	1842	1844	1848	1881	1882	1884	1888	
	2111	2112	2114	2118	2121	2122	2124	2128	2141	2142	2144	2148	2181	2182	2184	2188	
	2211	2212	2214	2218	2221	2222	2224	2228	2241	2242	2244	2248	2281	2282	2284	2288	
	2411	2412	2414	2418	2421	2422	2424	2428	2441	2442	2444	2448	2481	2482	2484	2488	
	2811	2812	2814	2818	2821	2822	2824	2828	2841	2842	2844	2848	2881	2882	2884	2888	
	4111	4112	4114	4118	4121	4122	4124	4128	4141	4142	4144	4148	4181	4182	4184	4188	
	4211	4212	4214	4218	4221	4222	4224	4228	4241	4242	4244	4248	4281	4282	4284	4288	
	4411	4412	4414	4418	4421	4422	4424	4428	4441	4442	4444	4448	4481	4482	4484	4488	
	4811	4812	4814	4818	4821	4822	4824	4828	4841	4842	4844	4848	4881	4882	4884	4888	
	8111	8112	8114	8118	8121	8122	8124	8128	8141	8142	8144	8148	8181	8182	8184	8188	
	8211	8212	8214	8218	8221	8222	8224	8228	8241	8242	8244	8248	8281	8282	8284	8288	
	8411	8412	8414	8418	8421	8422	8424	8428	8441	8442	8444	8448	8481	8482	8484	8488	
	8811	8812	8814	8818	8821	8822	8824	8828	8841	8842	8844	8848	8881	8882	8884	8888	