

Изпитни задачи за повторения от минали издания на курса

Практически упражнения към курса ["Увод в програмирането" за ученици](#).

Тествайте решенията си от тази тема в Judge:

<https://judge.softuni.bg/Contests/2642/Изпитни-задачи-за-повторения>

1. Хистограма

Четвърта задача от изпит в СофтУни от 6 март 2016.

Дадени са **n цели числа** в интервала **[1...1000]**. От тях някакъв процент **p1** са под 200, друг процент **p2** са от 200 до 399, друг процент **p3** са от 400 до 599, друг процент **p4** са от 600 до 799 и останалите **p5** процента са от 800 нагоре. Да се напише програма, която изчислява и отпечата процентите **p1, p2, p3, p4** и **p5**.

Пример: имаме **n = 20** числа: 53, 7, 56, 180, 450, 920, 12, 7, 150, 250, 680, 2, 600, 200, 800, 799, 199, 46, 128, 65. Получаваме следното разпределение и визуализация:

Диапазон	Числа в диапазона	Брой числа	Процент
< 200	53, 7, 56, 180, 12, 7, 150, 2, 199, 46, 128, 65	12	$p1 = 12 / 20 * 100 = 60.00\%$
200 ... 399	250, 200	2	$p2 = 2 / 20 * 100 = 10.00\%$
400 ... 599	450	1	$p3 = 1 / 20 * 100 = 5.00\%$
600 ... 799	680, 600, 799	3	$p4 = 3 / 20 * 100 = 15.00\%$
≥ 800	920, 800	2	$p5 = 2 / 20 * 100 = 10.00\%$

Вход

На първия ред от входа стои цялото число **n** ($1 \leq n \leq 1000$) – брой числа. На следващите **n реда** стои **по едно цяло число** в интервала **[1...1000]** – числата върху които да бъде изчислена хистограмата.

Изход

Да се отпечата на конзолата **хистограмата** – **5 реда**, всеки от които съдържа число между 0% и 100%, с точност две цифри след десетичната точка, например 25.00%, 66.67%, 57.14%.

Примерен вход и изход

Вход	Изход	Вход	Изход	Вход	Изход	Вход	Изход	Вход	Изход
3	66.67%	4	75.00%	7	14.29%	9	33.33%	14	57.14%
1	0.00%	53	0.00%	800	28.57%	367	33.33%	53	14.29%
2	0.00%	7	0.00%	801	14.29%	99	11.11%	7	7.14%
999	0.00%	56	0.00%	250	14.29%	200	11.11%	56	14.29%
	33.33%	999	25.00%	199	28.57%	799	11.11%	180	7.14%
				399		999		450	
				599		333		920	
				799		555		12	
						111		7	
						9		150	
								250	

									680	
									2	
									600	
									200	

2. Деление без остатък

Четвърта задача от изпит в СофтУни от 26 март 2016.

Дадени са **n**-на брой цели числа в интервала **[1...1000]**. От тях някакъв процент **p1** се делят без остатък на **2**, друг процент **p2** се делят без остатък на **3**, друг процент **p3** се делят без остатък на **4**. Да се напише програма, която изчислява и отпечатва процентите **p1**, **p2** и **p3**.

Пример: имаме **n = 10** числа: 680, 2, 600, 200, 800, 799, 199, 46, 128, 65. Получаваме следното разпределение и визуализация:

Деление без остатък на:	Числа в диапазона	Брой числа	Процент
2	680, 2, 600, 200, 800, 46, 128	7	$p1 = 7.0 / 10 * 100 = 70.00\%$
3	600	1	$p2 = 1 / 10 * 100 = 10.00\%$
4	680, 600, 200, 800, 128	5	$p3 = 5 / 10 * 100 = 50.00\%$

Вход

На първия ред от входа стои цялото число **n** ($1 \leq n \leq 1000$) – брой числа. На следващите **n** реда стои по едно цяло число в интервала **[1...1000]** – числата които да бъдат проверени на колко се делят.

Изход

Да се отпечата на конзолата **3 реда**, всеки от които съдържа процент между 0% и 100%, с точност две цифри след десетичната точка, например 25.00%, 66.67%, 57.14%.

- На **първият ред** – процентът на числата които се делят на **2**
- На **вторият ред** – процентът на числата които се делят на **3**
- На **третият ред** – процентът на числата които се делят на **4**

Примерен вход и изход

Вход	Изход	Вход	Изход
10	70.00%	3	33.33%
680	10.00%	3	100.00%
2	50.00%	6	0.00%
600		9	
200			
800			
799			
199			

46			
128			
65			

3. Чертане на крепост

Пета задача от изпит в СофтУни от 6 март 2016.

Да се напише програма, която прочита от конзолата **цяло число n** , въведено от потребителя, и чертае **крепост** с ширина $2 * n$ **колони** и височина n **реда** като в примерите по-долу. Лявата и дясната колона във вътрешността си са широки $n / 2$.

Вход

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

Изход

Да се отпечата на конзолата n текстови реда, изобразяващи **крепостта**, точно както в примерите.

Примерен вход и изход

вход	изход	вход	изход	вход	изход	вход	изход
3	<pre> / ^ \ / ^ \ \ ^ \ / ^ \ </pre>	4	<pre> / ^ ^ \ / ^ ^ \ \ ^ ^ \ / ^ ^ \ </pre>	5	<pre> / ^ ^ \ / ^ ^ \ \ ^ ^ \ / ^ ^ \ </pre>	8	<pre> / ^ ^ ^ \ / ^ ^ ^ \ \ ^ ^ ^ \ / ^ ^ ^ \ </pre>

4. Пеперуда

Пета задача от изпит в СофтУни от 26 март 2016.

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 отпечата на конзолата $2 * (n - 2) + 1$ текстови реда, изобразяващи **пеперудата**.

Примерен вход и изход

вход	изход	вход	изход	вход	изход

3	* \ /* @ * / \ *	5	*** \ /*** --- \ /--- *** \ /*** @ *** / \ *** --- / \ --- *** / \ ***	7	***** \ /***** ----- \ /----- ***** \ /***** ----- \ /----- ***** \ /***** @ ***** / \ ***** ----- / \ ----- ***** / \ ***** ----- / \ ----- ***** / \ *****
---	------------------------	---	--	---	--

5. Генератор за тъпи пароли

Шеста задача от изпит в СофтУни от 6 март 2016

Да се напише програма, която чете две цели числа n и L , въведени от потребителя, 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

6. Магически числа

Шеста задача от изпит в СофтУни от 26 март 2016.

Да се напише програма, която чете едно цяло „магическо“ число, въведено от потребителя, 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 -> $1 * 1 * 1 * 1 * 2 * 1 = 2$
- 111211 -> $1 * 1 * 1 * 2 * 1 * 1 = 2$
- 112111 -> $1 * 1 * 2 * 1 * 1 * 1 = 2$
- 121111 -> $1 * 2 * 1 * 1 * 1 * 1 = 2$
- 211111 -> $2 * 1 * 1 * 1 * 1 * 1 = 2$

Вход

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

Изход

На конзолата трябва да се отпечатаат 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 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

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "Обучение за ИТ кариера" на МОН за подготовка по професия "Приложен програмист".



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от фондация "Софтуерен университет" и се разпространява под свободен лиценз CC-BY-NC-SA (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).



SoftUni
Foundation

