

Міністерство освіти і науки України
Житомирський державний університет імені Івана Франка
Кафедра прикладної математики та інформатики

С.В. Матвійчук, С.С.Жуковський



Практикум програмування Python / C++ на e-olymp.com

Житомир
2019

Затверджено вченою радою Житомирського державного університету імені Івана Франка, протокол №___ від _____2019 р.

Рецензенти:

Морозов А.В. – кандидат технічних наук, доцент кафедри комп'ютерної інженерії та кібербезпеки, проректор з науково-педагогічної роботи, Житомирського державного технологічного університету

Шевчук П.Г – кандидат педагогічних наук, вчитель інформатики КУ Миропільської селищної ради "Опорний навчальний заклад "Миропільська гімназія"

М43 Практикум програмування Python / C++ на e-olymp.com (збірник задач з рекомендаціями до їх розв'язання) / С.В. Матвійчук, С.С.Жуковський – Житомир: Вид-во ЖДУ, 2019 - 235 с.

Автори:

Матвійчук Сергій Володимирович – вчитель інформатики Ружинської гімназії

Жуковський Сергій Станіславович – доцент кафедри прикладної математики та інформатики Житомирського державного університету імені Івана Франка, методист ЖОІППО, учитель інформатики ліцею №25 м. Житомира

В збірнику задач з олімпіадного програмування «Практикум програмування Python / C++ на e-olymp.com» містить теоретичний матеріал з базових основ мов програмування Python та C++, перелік задач з олімпіадного програмування що розміщені на сайті e-olymp, рекомендації щодо їх розв'язання та коди програм мовами програмування Python / C++.

Збірник задач може бути використаний учителями та учнями загальноосвітніх шкіл а також викладачами та студентами вищих навчальних закладів під час вивчення програмування та підготовки до олімпіад з програмування.

ЗМІСТ

ПЕРШІ КРОКИ З РУТНОН	8
ЗАМІСТЬ ВСТУПУ	8
ЛІНІЙНІ ПРОГРАМИ	10
Довідник РУТНОН	11
1024. HELLO WORLD!	12
2400. ТРИКУТНИКИ	12
248. ЮНИЙ САДІВНИК	13
6277. ПОКУПКА ВОДИ	13
7817. ГАРНЕ ЧИСЛО	14
1355. КІЛЬКІСТЬ N-ЗНАЧНИХ ЧИСЕЛ, ЩО МІСТИТЬ 7	15
7336. ПИРІЖКИ	15
7460. ПОЇЗДКА НА ЕКСКУРСІЮ.	16
933. СУМА ЦИФР ДВОЦИФРОВОГО ЧИСЛА	17
943. ОБМІН КРАЙНІХ	18
955. КВАДРАТ СУМИ	18
7459. НЕПАРНІ РОЗРЯДИ	19
1359. СТОРОНА КВАДРАТА	19
2806. ЧИСЛА	20
РОЗГАЛУЖЕННЯ	21
Довідник РУТНОН	22
6278. НОМЕРИ БУДИНКІВ	22
4736. ЧИ ДІЛИТЬСЯ НА 11?	23
6279. КІЛЬКІСТЬ ДНІВ У МІСЯЦІ	23
2036. ВІК	24
1955. КАЛЕНДАР	25
1357. НА СКІЛЬКИ НУЛІВ ЗАКІНЧУЄТЬСЯ ЧИСЛО	26
108. СЕРЕДНЄ З ЧИСЕЛ	26
1351. НАЙБІЛЬША ЦИФРА	27

2392. ЦІКАВА СУМА.....	27
7337. ЗНИЖКИ	28
107. КОМПАКТ-ДИСКИ	29
7670. СТЕПАН І ПОХІД В МАГАЗИН	30
7375. БОЧКА	31
7330. ПОДІЛЬНІСТЬ НА 3	32
ЦИКЛ WHILE.....	34
6274. ПОДІЛ СКАРБУ.....	34
388. ПЕРЕТВОРЕННЯ	35
8298. ЛИШИТЬСЯ ТІЛЬКИ ОДНЕ ЧИСЛО.....	35
1993. ЗВАЖУВАННЯ	36
4196. ШОКОЛАДНІ ПЛИТКИ.....	37
1684. МОБІЛЬНИЙ ТАРИФ "КОПІЙКА"	37
77. ГІРЛЯНДА	38
2. ЦИФРИ	40
1603. СУМА ЦИФР ЧИСЛА	41
1605. ДРУГА ЦИФРА ЧИСЛА	42
1607. ЧИСЛО У ЗВОРОТНОМУ ПОРЯДКУ	42
1609. КІЛЬКІСТЬ ДАНИХ ЦИФР В ЧИСЛІ	42
1601. НСД ДВОХ ЧИСЕЛ	43
1602. НСК ДВОХ ЧИСЕЛ	44
3917. ПЕРЕВІРКА НА ПРОСТОТУ	44
123. НУЛІ В КІНЦІ ЗАПИСУ N!	45
3918. РОЗКЛАДАННЯ НА МНОЖНИКИ.....	46
ЦИКЛ FOR.....	48
ДОВІДНИК РYТНОН.....	48
5325. ПО ПОРЯДКУ.....	49
4192. ОЛІМПІАДА	49
271. ФАКТОРІАЛ!	50
62. ФАКТОРІАЛ.....	50
140. ФІНАНСОВА ПІРАМІДА	51

2862. КІЛЬКІСТЬ ДІЛЬНИКІВ	52
446. РІВНІ ДІЛЬНИКИ	52
128. ЩАСЛИВІ КВИТКИ	53
4730. ФІБОНАЧЧІ	54
115. ДВІ ЦИФРИ	54
ПРОСТО МАСИВИ.....	56
Довідник РҮТНОН.....	57
7829. СУМА ЕЛЕМЕНТІВ МАСИВУ.....	57
7830. НАЙБІЛЬШИЙ ЕЛЕМЕНТ МАСИВУ	58
7832. КІЛЬКІСТЬ МАКСИМАЛЬНИХ	58
7831. СУМА БЕЗ МАКСИМАЛЬНОГО	58
7841. НЕПАРНІ ЕЛЕМЕНТИ	58
7842. ПАРНІ ІНДЕКСИ	58
7843. БІЛЬШІ ПОПЕРЕДЬОГО	59
7844. СУСІДИ ОДНОГО ЗНАКА	59
7845. БІЛЬШІ СВОЇХ СУСІДІВ	59
7846. НАЙБІЛЬШИЙ ЕЛЕМЕНТ	60
7847. КІЛЬКІСТЬ РІЗНИХ ЕЛЕМЕНТІВ	60
7848. ПЕРЕСТАВИТИ СУСІДНІ	60
7849. ОБМІНЯТИ МАХ І МІН	60
7850. УНІКАЛЬНІ ЕЛЕМЕНТИ	61
7833. БІЛЬШІ ЗА СЕРЕДНЄ АРИФМЕТИЧНЕ	61
7834. ДВА НАЙБІЛЬШИХ	61
5337. РІЗНІ-РІЗНІ.....	62
2440. N-ТЕ НАЙБІЛЬШЕ ЗНАЧЕННЯ.....	62
5721. ПОШУК ЕЛЕМЕНТА	62
3935. РЕВЕРС	63
2099. ДВА МАСИВИ	63
ОБРОБКА РЯДКІВ.....	64
Довідник РҮТНОН.....	64
4718. ПРИВІТ, ГАРРІ!	65

622. Одиниці	65
3254. 01110001, ОСЬ ЗАПИТАННЯ.....	66
7340. Поле-чудес.....	66
1427. Калькулятор	66
4724. Робимо зрізи	67
963. Перестановка слів	68
494. Голосні	68
4722. Квадрат числа	68
4726. Перше та останнє входження	69
5049. Видали пропуски.....	69
2164. Шифр Юлія.....	70
2704. Римська система числення	71
1966. Великий плюс	72
119. Степінь двійки	72
116. Довге число	73
2796. Зайва цифра	74
7377. Послідовність	74
7509. Одиниці і сімки	75
4835. Без двох одиниць підряд	76
ЗАДАЧІ НА МАСИВАХ	77
7822. Відсутній номер.....	77
113. Кульки	78
145. Квадрати	78
226. Олімпіада та Інтернет	79
8296. Комп'ютери в школи	80
7503. Олімпіада	80
7226. День календаря	81
7380. Шкільний вальс.....	82
1356. SMS голосування	83
4197. Кінофестиваль.....	84
7823. Кран	85
44. Одиниці	86

7819. КОРУПЦІЮ ГЕТЬ	88
18. PAINT2D	89
4831. РЮКЗАК	91
832. ЗРОСТАЮЧА ПІДПОСЛІДОВНІСТЬ	92
РІЗНІ ЗАДАЧІ	94
7507. ДОБУТОК	94
4193. ЯЛИНКИ	94
46. ВІДРІЗКИ	95
7232. КОМПЛЕКТАЦІЯ КОМП'ЮТЕРІВ	96
8297. ЧИСЛА З СІРНИКІВ	97
7508. НОМЕР ТИЖНЯ	98
193. СУМА ЦИФР	99
1681. СУМИ ЦИФР	100
7228. СКІЛЬКИ ШІСТОК?	101
2798. ПОСЛІДОВНІСТЬ	102
7376. ОБМІН ЦИФР	103
7824. БЕЗ ПОВТОРЕНЬ	104
194. ДОБУТОК ЦИФР	105
2794. ДІЛЬНИКИ	106
7505. ПОСЛІДОВНІСТЬ КРАТНИХ	107
7381. СВОЯ ГРА	108

ПЕРШІ КРОКИ З PYTHON

ЗАМІСТЬ ВСТУПУ

Як навчити школяра самостійно написати перший програмний код, ввести потрібні дані, отримати правильні результати? Причому, всі етапи такої діяльності пройти самостійно, продемонструвавши вчителю тільки фінальний успіх. Це схоже, коли навчаються плавати, їздити на ковзанах або на автомобілі. Абсолютно все залежить тільки від тебе, а наставник може вдалимими порадами скоротити шлях до успіху, додати майстерності, закріпити досвід. Отже, читачу, все в твоїх руках. Ми лише допоможемо тобі вільно почувати себе в лабіринтах сучасних алгоритмів і мов програмування.

Найперше і найважливіше - дуже сильно захотіти навчитися програмувати, вже потім необхідно мати в своєму комп'ютері середовище програмування однієї з сучасних мов, наприклад, Python, нік на сайті www.e-olymp.com, а ще бажання зрозуміти математичну модель, правильно скласти і виконати програмний код. Наша книга має стати надійним провідником на початку такого шляху.

У навчанні програмуванню ми не будемо дотримуватись класичного формального методу, коли спочатку вивчають конструкції і правила мови програмування, а лише потім переходять до розв'язання задач, використовуючи вивчені і зрозумілі елементи мови.

Наш підхід до викладу матеріалу можна назвати «програмування за зразком», діємо за таким планом: задача, ідея і пояснення алгоритму, запис програми мовою Python. А далі пропонується щось виправити в програмі, щоб розв'язати наступну задачу або написати аналогічну. Більшість необхідних команд, функцій і процедур пояснені в коротких довідниках Python. В деяких місцях для розуміння коду потрібно буде застосувати інтуїцію або глянути help по Python в Інтернеті.

Ми не ставимо перед собою завдання досконало вивчити мову програмування, а просто будемо розв'язок задачі записувати кодом програми. Отже, головним буде зміст, а не форма запису алгоритмів або опис всіх можливостей мови програмування. Python якраз дає змогу

таке зробити: мінімум формальностей, максимум математичного змісту і лаконічності коду.

Задачі, вміщені в збірнику, були в різний час запропоновані на районних(міських) і обласних олімпіадах Житомирської області, а також участвовали в змаганнях на сайті **e-olymp**. Авторство багатьох задач належить відомим вчителям, серед яких були Володимир Леонідович Дідковський та Анатолій Васильович Присяжнюк – тепер вже класики шкільної інформатики і олімпіадного руху.

Збірник ставить своєю метою допомогти вчителям, а також їх учням зробити перші кроки на складному і терновому шляху, який безумовно приведе до успіху. Закладені цеглинки розв'язаних задач і алгоритмів побудують в майбутньому монолітну споруду з досвіду, інтуїції та професіоналізму. Задачі розбиті по розділах, а у викладі матеріалу автор намагався дотримуватись наступності (наскільки це було можливим). Тому, іноді, деякі записи в кодах програм пояснені вже після їх викладу, що заставить читача думати і шукати самостійно відповіді на незрозумілі речі.

Збірник стане в пригоді на уроках інформатики, факультативах, гуртках, при підготовці до олімпіад і змагань.

ЛІНІЙНІ ПРОГРАМИ

Розпочнемо з вправ, які традиційно називають лінійними програмами – тобто не потребують особливих методів і прийомів програмування, а мають три чітко окреслених блоки, які послідовно виконуються в програмі – це ввід даних, обчислення та вивід результатів. Зазвичай, в таких програмах проблематичною є друга, тобто математична частина. Зрозуміло, що кожного разу будуть різні формули, їх можна придумати використовуючи свій математичний досвід або глянути в Інтернеті. Потрібно не забувати, що Ви розробник програми і невірно вибрана математична модель буде давати неправильні результати або не працювати взагалі. Короткі коментарі по інструкціях в лінійних програмах, тільки тези:

- для друку чисел, тексту і значень змінних в мові Python використовується оператор **print()**;
- щоб вивести текст, наприклад **Hello!**, його записують в круглих дужках і в лапках після **print**, ось так **print("Hello!")**;
- щоб вивести числове значення, в круглих дужках після **print()** записується константа, ім'я змінної або арифметичний вираз;
- щоб вивести декілька об'єктів, їх записують аргументами в інструкції **print()** через кому;
- імена змінних в Python відповідають загальноприйнятим правилам в мовах програмування, тобто позначаються буквами і цифрами (перша буква), але великі і маленькі букви розрізняються;
- типи змінних попередньо прописувати не потрібно;
- під час виконання програми одна та ж сама змінна може приймати значення різних типів в залежності від інструкцій;
- для перетворення типів значень змінних використовуються функції **int()**, **float()**, **bool()**, **str()**;
- ввід даних в Python виконує оператор **input()**, причому введене значення рахується текстовим рядком;
- для введення чисел використовуються інструкції **int(input())** для цілих або **float(input())** для дійсних;
- вирази в командах присвоювання і друку використовують загальноприйнятий порядок дій, арифметичні операції і математичні функції;
- схема лінійного алгоритму - перш за все потрібно ввести дані, потім обчислити і тільки на кінець вивести результати.

Якщо поки не все зрозуміло, не біда. Можете сміливо приступати до розгляду конкретних задач, а тези перечитаєте ще раз пізніше.

ДОВІДНИК PYTHON

Основні арифметичні операції	
додавання	+
віднімання	-
множення	*
ділення	/
піднесення до степеня	**
ділення націло	//
остача від ділення націло	%
Основні арифметичні функції	
модуль аргументу x	<code>abs(x)</code>
перетворення рядка x до цілого	<code>int(x)</code>
перетворення рядка x в системі числення n до цілого	<code>int(x, n)</code>
перетворення аргументу до дійсного	<code>float(x)</code>
округлення аргументу до n знаків після коми	<code>round(x, n)</code>
перетворення цілого числа в двійковий рядок	<code>bin(x)</code>
перетворення цілого числа в вісімковий рядок	<code>oct(x)</code>
перетворення цілого числа в шістнадцятьковий рядок	<code>hex(x)</code>
повертає частку і остачу від ділення a на b	<code>divmod(a, b)</code>
повертає a в степені b	<code>pow(a, b)</code>
Основні команди в лінійних програмах	
ввести текстовий рядок з консолі в x	<code>x = input()</code>
ввести ціле число в X	<code>x = int(input())</code>
ввести дійсне число в X	<code>x = float(input())</code>
ввести декілька цілих чисел в X, Y	<code>x, y = map(int, input().split())</code>
ввести декілька дійсних чисел в X, Y	<code>x, y = map(float, input().split())</code>
вивести значення X	<code>print(x)</code>
вивести значення X, Y	<code>print(x, y)</code>
виконує рядок S програмного коду	<code>eval(s)</code>

1024. HELLO WORLD!

Виведіть повідомлення "Hello World!"

Не будемо порушувати традицію і чемно привітаємося зі світом на Python, тим паче, що це так просто.

```
print("Hello World!")
```

2400. ТРИКУТНИКИ

Михайлик любив малювати трикутники, але він це робив у незвичний спосіб. Спочатку малював довільний трикутник, потім кожну сторону ділив на n рівних частин і проводив через точки поділу прямі, паралельні сторонам трикутника. У результаті виходить декілька рівних між собою трикутників. **Допоможіть** Михайлику знайти найбільшу кількість однакових трикутників у його фінальному рисунку.

Вхідні дані: Натуральне число n . $1 \leq n \leq 2 \cdot 10^9$.

Вихідні дані: Найбільша кількість рівних між собою трикутників.

Input 2 **Output** 4

Для початкових значень $n=1,2,3,4$ маємо результати **1,4,9,16**. Логічно припустити, що в загальному випадку відповідь буде n^2 . Дійсно, кожне збільшення n на одиницю збільшує загальну кількість маленьких трикутників ще на $2 \cdot n - 1$ штук. Тобто сумарно однакових трикутників буде

$1 + 3 + 5 + \dots + (2 \cdot n - 1)$

або за формулою суми арифметичної прогресії

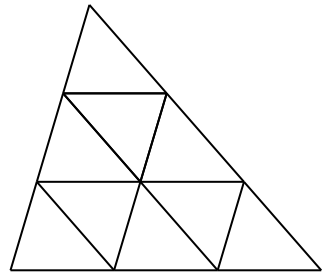
$(1 + 2 \cdot n - 1) \cdot n / 2 = 2 \cdot n \cdot n / 2 = n^2$. Отже, діємо за таким алгоритмом:

- в змінну n читаємо ціле число;
- обчислюємо результат $k = n^2$;
- виводимо значення k .

Ось два варіанта програми:

```
n=int(input())
k=n*n
print(k)
```

```
print(int(input())**2)
```



248. ЮНИЙ САДІВНИК

Для економії часу перефразуємо умову. Листочки на гілці ростуть за таким правилом: на верхівці знаходиться один листочок, а далі на першому ярусі два, на другому – чотири, на третьому – шість і тд. Скільки всього листочків на гілці, що має n ярусів?

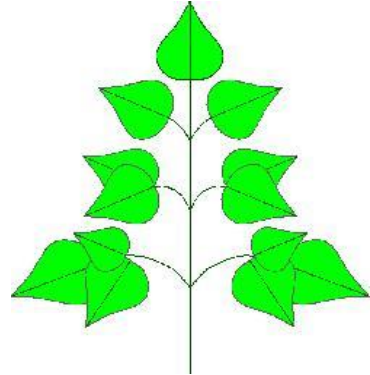
Вхідні дані: Кількість ярусів n ($1 \leq n \leq 1000$).

Вихідні дані: Сумарна кількість листочків.

Input 3 **Output** 13

Власне, дуже схожа на попередню задачу, відрізняється тільки обчислювальний блок, тобто потрібно скласти правильно формулу для обчислення результату. Наразі, є можливість ще раз перевірити математичні здобутки.

Кількість листочків на всіх ярусах $1..n$ – сума з n членів арифметичної прогресії, у якій перший член і різниця дорівнюють 2 , а останній член дорівнює $2*n$. Використовуємо формулу суми $(2+2*n)*n/2=(n+1)*n$ і не забуваємо додати один листочок на верхівці. Ось такий код програми.



```
n=int(input())
m=(n+1)*n+1
print(m)
```

Спробуйте розв'язати:

Сир для Анфіси - 2	67
Квадрат і точки	133
Сума чисел на проміжку	2860
Карткові будиночки	5716

6277. ПОКУПКА ВОДИ

Вартість пляшки води, враховуючи вартість порожньої пляшки, становить 1 грн 20 коп., а вартість порожньої пляшки 20 коп.

Скільки пляшок води можна випити на n грн, враховуючи, що порожні пляшки можна здавати, і на одержані гроші купувати нові пляшки води. Вода без пляшки не продається.

Вхідні дані: Натуральне число n ($1 \leq n \leq 1000$).

Вихідні дані: Кількість пляшок води, яку можна випити на n грн.

Input 2 **Output** 1

Задача не задумувалась для такого блискучого розв'язання, а передбачала циклічну модель покупки чергової партії пляшок з наступною задачею пустої посуду, але насправді все виявилось набагато простіше. Якби можна було порожню пляшку взяти в борг, то випитих пляшок було б n , але так як цього робити не можна, то у нас після операцій покупки і здачі завжди буде лишатись **1** грн., якої не вистачатиме на покупку ще однієї пляшки води. Отже, на n грн ми зможемо випити лише **$n-1$** пляшок.

Python дуже лаконічний і знову два коротких варіанта програми.

<pre>n=int(input()) print(n-1)</pre>	<pre>print(int(input())-1)</pre>
--------------------------------------	----------------------------------

Спробуйте розв'язати:

Друзі Степана	7401
Озеро з ліліями	2059
Додавання	3741
Сума квадратів	519

7817. ГАРНЕ ЧИСЛО

"Гарним" будемо вважати число, що складається лише з непарних цифр. Наприклад число **157953** гарне, а число **2452117** ні. Необхідно з'ясувати, скільки існує n - значних гарних чисел.

Вхідні дані: Одне ціле невід'ємне число n ($0 \leq n \leq 20$).

Вихідні дані: Кількість гарних чисел.

Input 4 **Output** 625

Розглянемо декілька комбінаторних задач і ось одна з них. На кожний розряд n -значного "гарного" числа претендує будь-яка одна з п'яти непарних цифр. Очевидно, що всього таких чисел буде **5^n** .

<pre>n=int(input()) print(5**n)</pre>	<pre>print(5**int(input()))</pre>
---------------------------------------	-----------------------------------

Два маленьких коди на Python.

1355. КІЛЬКІСТЬ N-ЗНАЧНИХ ЧИСЕЛ, ЩО МІСТИТЬ 7

Знайти **k** - кількість **n**-значних натуральних чисел, що мають у своєму запису хоча б одну цифру **7**.

Вхідні дані: Одне натуральне число **n** ($1 \leq n \leq 10$).

Вихідні дані: Шукане число **k**.

Input 2 **Output** 18

Відповідь до задачі можна знайти виходячи з таких міркувань:

- кількість всіх **n**-значних натуральних чисел дорівнює $9 \cdot 10^{n-1}$;
- кількість **n**-значних чисел, що не містять **7** дорівнює $8 \cdot 9^{n-1}$;
- якщо від першого результату відняти другий, то отримаємо кількість **n**-значних чисел, що мають у своєму запису хоча б одну цифру **7** (до речі, для будь-якої іншої ненульової цифри результат буде аналогічний).

Лише один варіант програми:

```
n=int(input())
k=9*10**(n-1)-8*9**(n-1)
print(k)
```

Спробуйте розв'язати:

Змагання з тенісу	1287
Ланч	1289
Номерний знак	1290
У хокей грають справжні...	1326
Садівник-художник	17

7336. ПИРІЖКИ

Пиріжок у шкільній їдальні коштує **a** гривень та **b** копійок. Знайдіть скільки гривень та копійок заплатить Петрик за **n** пиріжків.

Вхідні дані: Три натуральних числа **a**, **b**, **n** ($0 \leq a, b, n \leq 100$).

Вихідні дані: Через пропуск два числа: вартість покупки у гривнях та копійках.

Input 1 25 2 **Output** 2 50

Звичайна математична задачка. Знаходимо ціну одного, а потім **n** пиріжків у копійках. Розділивши отриманий результат націло на **100**, отримаємо цілі гривні, а остача при такому діленні – це копійки. Зверніть увагу, що вперше ми використовуємо команду для введення декількох цілих чисел, записаних в рядок через проміжок.

Може бути такий код:

```
a, b, n = map(int, input().split())
s = (a * 100 + b) * n
p = s//100
q = s%100
print (p,q)
```

Або ще ось так:

```
a, b, n = map(int, input().split())
p,q = divmod((a*100+b)*n,100)
print (p,q)
```

7460. ПОЇЗДКА НА ЕКСКУРСІЮ.

Учні **10-Б** класу, на осінні канікули, вирішили поїхати на екскурсію до столиці. Знаючи кількість хлопчиків **n** та дівчаток **m**, визначити скільки потрібно замовити кімнат в готелі, в якому є кімнати на **k** місць кожна, за умови, що хлопчиків та дівчаток поселяти разом заборонено.

Вхідні дані: Три цілих числа **n, m, k** ($n, m, k \leq 100$).

Вихідні дані: Одне число – необхідна кількість кімнат.

Input 6 12 3 **Output** 6

Все просто, знаходимо кількість кімнат для хлопчиків і дівчаток окремо і виводимо у відповідь їх суму. Цікаво, як знайти цю кількість? У випадку якщо, наприклад, **n** ділиться на **k** націло, потрібно розділити учнів на кількість місць – отримаємо кількість кімнат. А якщо націло не ділиться, то потрібно додати ще одну кімнату для тих учнів, які не потрапили до повністю заповнених.

Розглянемо як це можна записати в математичних формулах: $n//k$ – кількість повністю заповнених кімнат, $n\%k$ – кількість учнів, що лишилися. Далі, вираз $\text{int}(n\%k > 0)$ дорівнює **1**, якщо є остача або дорівнює **0** ($n\%k = 0$), тобто остачі немає. Для корегування кількості

кімнат логічно було б скористатися командою розгалуження, але ми її ще не розглянули.

```
n, m, k = map(int, input().split())
p = n//k+int(n%k>0)
q = m//k+int(m%k>0)
print (p+q)
```

Спробуйте розв'язати:

Поділ яблук - 1	4716
Поділ яблук - 2	4717
Вартість розмови	2391

Декілька наступних задач представляють серію про операції з цифрами в цілих числах фіксованої довжини. Спочатку, зауважимо основні правила.

В десятковій системі числення, наприклад, трицифрове число може бути записано в такий спосіб: $754=7*10^2+5*10^1+4*10^0$, тобто цифра сотень множиться на **100**, цифра десятків – на **10** і додається цифра одиниць.

Цифру наймолодшого розряду можна знайти, як остачу від ділення на **10**. Щоб виділити цифру десятків, потрібно видалити цифру одиниць, розділивши досліджуване число націло на **10**. Для наступних цифр процес можна повторити.

У випадку від'ємних значень, необхідно нейтралізувати знак мінус, застосувавши функцію модуль.

933. СУМА ЦИФР ДВОЦИФРОВОГО ЧИСЛА

Знайти суму цифр даного двоцифрового числа.

Вхідні дані: Задане двоцифрове ціле число.

Вихідні дані: Сума його цифр.

Input 23 **Output** 5

Насамперед відкинемо знак мінус, використавши функцію **abs()**. Далі розділимо задане число на **10** націло, тоді в результаті частка – буде цифра десятків, а остача – цифра одиниць. Можемо використати вже відомі операції цілочисельного ділення або ж функцію **divmod**. У відповідь виводимо суму отриманих .

Дві версії коду до цієї задачі.

<pre>n=abs(int(input())) a=n//10 b=n%10 print(a+b)</pre>	<pre>n=int(input()) a,b=divmod(abs(n),10) print(a+b)</pre>
----------------------------------------------------------	------------------------------------------------------------

943. ОБМІН КРАЙНІХ

У заданому трицифровому натуральному числі поміняти першу і останню цифри місцями.

Вхідні дані: Натуральне трицифрове число.

Вихідні дані: Число, отримане в результаті вказаного обміну.

Input 198 **Output** 891

Нехай **a** – цифра сотень, **c** – цифра одиниць заданого натурального числа **x**, тоді операція $x - 100*a + 100*c - c + a$ або $x + 99*(c - a)$ виконує потрібний обмін.

<pre>x=int(input()) a=x//100 c=x%10 print(x+99*(c-a))</pre>	<pre>print(input()[::-1])</pre>
-------------------------------------------------------------	---------------------------------

Другий варіант програми читає число, як текстовий рядок і виводить його символи у протилежному напрямку. Такий суперкороткий код проходить лише **90%** тестів, подумайте чому.

955. КВАДРАТ СУМИ

Знайти квадрат суми цифр чотирицифрового натурального числа.

Вхідні дані: Задане натуральне чотирицифрове число.

Вихідні дані: Квадрат суми цифр заданого числа.

Input 4765 **Output** 484

Застосовуємо вже відомі операції до заданого чотирицифрового числа, щоб виділити цифри тисяч, сотень, десятків і одиниць, а знайти квадрат їх суми, думаю, буде нескладно. Два варіанта програмного коду до цієї задачі відрізняються тільки способом знаходження цифр.

<pre>x=int(input()) a=x//1000 b=x//100%10 c=x//10%10 d=x%10 print((a+b+c+d)**2)</pre>	<pre>x=int(input()) a,x=divmod(x,1000) b,x=divmod(x,100) c,d=divmod(x,10) print((a+b+c+d)**2)</pre>
---------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------

7459. НЕПАРНІ РОЗРЯДИ

Добуток цифр п'ятицифрового числа, що стоять на непарних розрядах.

Вхідні дані: Ціле п'ятицифрове число **n**.

Вихідні дані: Вивести добуток цифр на непарних розрядах.

Input 12345 **Output** 15

```
x=abs(int(input()))
a=x//10000
b=x//100%10
c=x%10
print(a*b*c)
```

Спробуйте розв'язати:

Розклад трицифрового числа	935
Добуток цифр трицифрового числа	906
Зворотній порядок	947
Двозначне з чотиризначного	949

Розділ «Лінійні програми» завершує декілька нетипових задач, де ідея алгоритму використовується тільки для конкретної задачі і, як правило, більше не використовується.

1359. СТОРОНА КВАДРАТА

Знайти цілочисельну довжину сторони квадрата, який можна отримати з двох прямокутників **axb** та **cxд**, розрізавши їх паралельно сторонам на прямокутники, а потім склавши так, щоб утворився квадрат найбільш можливої площі.

Вхідні дані: У одному рядку знаходиться чотири числа **a, b, c, d**.

Вихідні дані: Одне число - сторона найбільшого квадрата .

Input 1 2 3 4 **Output** 3

Очевидно, що площа майбутнього квадрата не може перевищувати сумарну площу прямокутників, тобто це найбільший повний квадрат не більший $a*b+c*d$. Отже, довжина сторони квадрата дорівнює цілій частині від квадратного кореня з цієї площі. Квадратний корінь обчислимо, як степінь з показником **0.5**. Лише дві команди коду.

```
a, b, c, d = map(int,input().split())
print (int((a*b + c*d)**0.5))
```

2806. ЧИСЛА

Задано натуральне число **n**. Знайти кількість натуральних чисел, що не перевищують **n** і не діляться на жодне з чисел **2, 3, 5**.

Вхідні дані: Натуральне число **n** ($1 \leq n \leq 1000000000$).

Вихідні дані: Вивести знайдену кількість.

Input 10 **Output** 2

Скільки буде чисел на проміжку **1..n**, що діляться на **2** націло. Це просто перевірити простим підрахунком: виявляється $n//2$. Аналогічно і для других дільників. Порахуємо кількість чисел, що діляться хоча б на одне з чисел **2, 3** або **5**. Для цього потрібно додати ті, що діляться на **2**, на **3**, і на **5**, відняти ті, що діляться на **2*3=6**, **2*5=10**, **3*5=15** (бо ми їх порахували двічі) і додати ті, що діляться на **2*3*5=30** (бо ми їх зовсім не врахували). Відповідь отримаємо, якщо від усіх чисел проміжку **1..n**, тобто **n** відняти попередній результат.

```
n=int(input())
print (n-(n//2+n//3+n//5-n//6-n//10-n//15+n//30))
```

Спробуйте розв'язати:

Шкільна олімпіада	6272
Прямокутник	769
Білі кубики	478
Квадрат по двох вершинах	6344

РОЗГАЛУЖЕННЯ

Всі програми з попереднього розділу були лінійними, команди в яких виконувались безальтернативно – підряд, як рядки в пісні. Але зазвичай, розв'язуючи інші задачі, програма в залежності від вхідних даних повинна вміти зробити вибір: значення парне чи непарне; число менше, рівне або більше нуля; пора року весна, літо, осінь або зима, ось тоді в пригоді стане команда розгалуження. Така конструкція традиційно має засіб для перевірки логічних тверджень і перехід до виконання визначених команд у позитивному і протилежному випадках.

В кожній мові програмування розгалуження має свої особливості запису, але основний зміст від цього не змінюється. В мові Python команда розгалуження або вибору реалізується службовими словами **if** – **elif** – **else**. Спочатку записується частина **if** з умовним виразом, далі можуть йти декілька секцій **elif** і насамкінець необов'язкова секція **else**. Загальна форма розгалуження на Python:

```
if умова1 :      або   if умова1 :   блок1
    блок1
elif умова2 :    elif умова2 :   блок2
    блок2
else :           else           :   блок3
    блок3
```

Істинність умов перевіряється в порядку, як вони записані в команді. Відповідний блок команд виконується тільки у випадку, якщо його умова істина (**True**), а наступна перевірка умов припиняється. Блок команд секції **else** буде задіяний у випадку, якщо всі попередні умови виявилися хибними (**False**).

Для запису умов використовуються знаки відношення між величинами і службові слова **and**, **or** і **not**. Якщо потрібно, щоб декілька простих умов виконувались одночасно – з'єднуємо їх операцією **and**; **or** використовуємо, тоді коли достатньо виконання однієї з простих умов, а **not** – щоб записати, що умова не виконується.

Зауважимо також, що існує зв'язок між логічними значеннями умов і цілими числами, втім для такої емуляції достатньо лише двох чисел, отже **int(True)=1; int(False)=0; bool(1)= True; bool(0)= False**.

ДОВІДНИК PYTHON

Запис відношень між величинами	
дорівнює	<code>==</code>
не дорівнює	<code>!=</code>
більше	<code>></code>
менше	<code><</code>
більше рівне	<code>>=</code>
менше рівне	<code><=</code>
логічне «І»	<code>and</code>
логічне «АБО»	<code>or</code>
логічне заперечення	<code>not</code>
Функції, що пов'язані з розгалуженням	
повертає True, якщо всі елементи True	<code>all</code> (послідовність)
повертає True, якщо хоча б один елемент True	<code>any</code> (послідовність)
повертає найбільший елемент	<code>max</code> (послідовність)
повертає найменший елемент	<code>min</code> (послідовність)
перетворення аргументу до логічного типу	<code>bool(x)</code>
створення списку об'єктів	<code>list(x)</code>

6278. НОМЕРИ БУДИНКІВ

Вивести **1**, якщо будинки з номерами **n** і **m** знаходяться на одній стороні вулиці, і **0** у протилежному випадку.

Вхідні дані: Значення **n** і **m** ($1 \leq n, m \leq 100$).

Вихідні дані: Відповідь до задачі.

Input 1 2 **Output** 0

Розпочнемо з завдання, відповідь у якому може мати всього два можливих варіанти **1**, якщо **n** і **m** мають однакову парність і **0** у протилежному випадку. Отож, має бути дві версії коду: в першому детально опишемо можливі випадки парності значень **n** і **m**, у другому просто порівняємо остачі від ділення на **2** значень **n** і **m**. Для

лаконічності використаємо властивість логічних значень `int(True)=1` і `int(False)=0`.

```
n,m = map(int,input().split())
if n%2==0 and m%2==0 or n%2!=0 and m%2!=0 :
    print(1)
else :
    print(0)
```

І короткий код.

```
n,m = map(int,input().split())
print (int(n%2==m%2))
```

4736. Чи ділиться на 11?

Для введеного числа перевірити, чи ділиться воно на **11**.

Вхідні дані: Натуральне число.

Вихідні дані: Виведіть **Yes**, якщо число ділиться на **11**, інакше виведіть **No**.

Input 17 **Output** No

```
n = int(input())
if n%11==0 : print('Yes')
else : print('No')
```

Спробуйте розв'язати:

Чи ділиться на 2, 4 і 8?	4733
Торт	774
Чи високосний рік?	5054
Порівняння довгих чисел	5086
Сума непарних чисел	2861

6279. КІЛЬКІСТЬ ДНІВ У МІСЯЦІ

Вивести кількість днів в **n**-му місяці **m**-го року, враховуючи, що рік є високосним, якщо він кратний **4**, але не кратний **100**, або ж кратний **400**.

Вхідні дані: Значення **n** та **m** ($1 \leq n \leq 12$, $1 \leq m \leq 2100$).

Вихідні дані: Знайдена кількість днів.

Input 2 2016

Output 29

По ходу програми шукана кількість днів буде розміщена в змінній **d**. Так як кількість місяців, які мають **31** день – найбільша, то ініціалізуємо змінну **d** відповідним значенням. виправимо значення **d** на **30** у випадку **n=4, 6, 9** або **11**. Якщо ж **n=2**, то в залежності від року в змінній **d** отримаємо **28** або **29**. Відповідна умова високосності прописана аргументом функції **int()**, яка повертає **1** у випадку, якщо умова **True**, рік високосний і лютий буде мати **29** днів. В протилежному випадку, якщо умова буде **False**, то функція **int()** повертається **0**, а днів буде **28**.

```
n,m=map(int,input().split())
d=31
if n==4 or n==6 or n==9 or n==11 :
    d=30
elif n==2 :
    d=28+int(m%4==0 and m%100!=0 or m%400==0)
print(d)
```

2036. ВІК

Дано натуральне число **N** ($1 \leq N \leq 100$), яке визначає вік людини. Виведіть це число з відповідним доданком **rik**, **roki**, **rokiv**. Наприклад, **1 rik**, **12 rokiv**, **94 roki**.

Вхідні дані: Єдине число - вік людини **N** ($1 \leq N \leq 100$).

Вихідні дані: Вивести вік з відповідною назвою.

Input 1

Output 1 rik

Відповідь залежить від значень цифри десятків **a** і цифри одиниць **b** в заданому числі років **n**. Насамперед, обчислюємо значення цифр числа **n**, а у відповідь запишемо назву '**rokiv**', так як таких випадків буде найбільше. Далі залишилося описати два особливих випадки, якщо цифра **a** не дорівнює **1**, то у випадку **b=1**, маємо '**rik**', а у випадку **b=2,3** або **4** відповідь буде '**roki**'.

```
n=int(input())
a,b=divmod(n,10); s='rokiv'
if a!=1 and b==1 : s='rik'
if a!=1 and 1<b<5 : s='roki'
print(n,s)
```


Спробуйте розв'язати:

Який трикутник ?	905
Яка чверть ?	918
Квартал	2043
Місяць	2044
N-значні числа	1288

1955. КАЛЕНДАР

Толі потрібно порахувати скільки йому років, якщо відомо дату його народження та поточну дату.

Вхідні дані: У першому рядку три натуральних числа: день, місяць і рік, коли народився Толя. У другому рядку задано поточну дату.

Вихідні дані: Скільки років Толі.

Input	Output
13 4 2003	7
17 7 2010	

В загальному випадку відповідь рівна різниці між поточним і роком народження. Але, можливо, нинішнього року день народження ще не наступив? Це можливо, коли поточний місяць менший ніж місяць народження або поточний день менший ніж день народження, якщо місяці рівні. В цьому випадку результат потрібно зменшити на одиницю.

```
D,M,Y=map(int,input().split())
d,m,y=map(int,input().split())
a=y-Y
if m<M or m==M and d<D : a-=1
print(a)
```

Варто прокоментувати особливості запису команд присвоювання в Python. В нашому випадку запис **a-=1** означає – зменшити значення **a** на одиницю, тобто рівносильний команді **a=a-1**. Наведемо декілька прикладів еквівалентних команд присвоювання в мові Python:

a = a + b	a += b
a = a * b	a *= b
a = a // b	a //= b
a = a % b	a %= b

1357. НА СКІЛЬКИ НУЛІВ ЗАКІНЧУЄТЬСЯ ЧИСЛО

Деяке натуральне число n можна розкласти на прості множники, серед яких m дільників, що дорівнюють 2 і k дільників 5 . На скільки нулів закінчується натуральне число n ?

Вхідні дані: Два числа m і k ($0 \leq m, k \leq 32000$).

Вихідні дані: Кількість кінцевих нулів.

Input 1 2 **Output** 1

Кожен кінцевий нуль числа утворюється за рахунок пари дільників 2 і 5 , тому нулів буде рівно стільки, скільки можливо утворити таких пар, а це менше значення з чисел m і k . В першій версії програми мінімальне значення з двох знаходимо класичним способом з використанням одного розгалуження, а в другій застосовуємо стандартну функцію `min()`.

```
m,k=map(int,input().split())
n=m
if k<n : n=k
print(n)
```

```
m,k=map(int,input().split())
print(min(m,k))
```

108. СЕРЕДНЄ З ЧИСЕЛ

Дано три різні числа a , b , c . Вивести середнє з них.

Вхідні дані: Числа a , b , c цілі та за модулем не більші 1000 .

Вихідні дані: Вивести середнє з трьох чисел.

Input 11 3 7 **Output** 7

Існує маса способів вирішити таку задачу:

- знайти число, яке більше меншого і менше більшого;
- відсортувати числа і вивести середнє;
- від суми трьох відняти значення меншого і більшого з них.

Наступний код реалізує останній алгоритм.

```
a,b,c = map(int,input().split())
print (a+b+c-min(a,b,c)-max(a,b,c))
```

Зауважимо, що в програмах на Python дуже зручно користуватись функціями `min()` і `max()`, причому вони можуть застосовуватись, як до кількох змінних, так і до масивів з великою кількістю елементів.

1351. НАЙБІЛЬША ЦИФРА

Знайти найбільшу цифру цілого чотирицифрового числа.

Вхідні дані: Одне ціле чотирицифрове число.

Вихідні дані: Шукана найбільша цифра.

Input 2134 **Output** 4

В цій задачі все дуже просто: знаходимо цифри чотиризначного числа, а потім відомим способом вибираємо з них найбільшу. Більш цікавим буде друга версія коду, подумайте як він працює?

```
x=int(input())
a,x=divmod(x,1000)
b,x=divmod(x,100)
c,d=divmod(x,10)
print(max(a,b,c,d))
```

```
print(max(list(input())))
```

2392. ЦІКАВА СУМА

Задано тризначне натуральне число **n**. Визначити суму найбільшого та найменшого трицифрових чисел, які можуть бути утворені з числа **n** перестановкою цифр.

Вхідні дані: Натуральне число **n** ($100 \leq n \leq 999$).

Вихідні дані: Відповідь до задачі.

Input 521 **Output** 787

Будемо діяти за таким алгоритмом:

- виділити цифри **a**, **b**, **c** заданого числа;
- відсортувати їх значення у незростаючому порядку;
- число $100*a+10*b+c$ буде найбільшим;
- у зальному випадку найменшим буде число $100*c+10*b+a$;
- потрібно врахувати особливі випадки **b=c=0** і **c=0**;
- вивести шукану суму.

Перший код працює з цілими числами за описаним вище алгоритмом. Другий читає число як рядок і, відсортувавши в ньому цифри, складає їх у рядки з потрібних цифр по аналогії з першим алгоритмом. Тільки потім конвертує їх в цілі значення, використавши функцію **int()**, щоб обчислити і вивести потрібну суму.

<pre>n=int(input()) a,n=divmod(n,100) b,c=divmod(n,10) if a<b : a,b=b,a if b<c : b,c=c,b if a<b : a,b=b,a x=100*a+10*b+c y=100*c+10*b+a if b==0 : y=100*a elif c==0 : y=100*b+a print(x+y)</pre>	<pre>a = list(input()) a.sort() x = int(a[2]+a[1]+a[0]) if a[1]=='0': y=int(a[2]+a[1]+a[0]) elif a[0]=='0': y=int(a[1]+a[0]+a[2]) else : y=int(a[0]+a[1]+a[2]) print(x+y)</pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Наразі, ще декілька слів про команду присвоїти в Python. Звісно, що для надання значень змінним використовується знак дорівнює «=». Це може бути звична команда: ім'я змінної зліва, вираз або константа справа. Декілька послідовних команд присвоювання в Python можна записувати в одну стрічку, при цьому імена змінних перераховуються зліва від знаку дорівнює через кому, а їх майбутні значення справа. Після виконання такої команди перераховані змінні відповідно набувають своїх значень – так зване паралельне присвоювання.

Цікаво, що алгоритм обміну значень двох змінних в Python можна записати однією стрічкою, ось так **a,b = b,a**.

Також має місце групове присвоювання, це коли потрібно ініціалізувати нулем декілька змінних, то можна застосувати команду **a=b=c=0**.

Спробуйте розв'язати:

Лінивий Мишко	3867
Трицифрове число	5628
Більше серед чотирьох	7812

7337. ЗНИЖКИ

У супермаркеті електроніки, якщо вірити телерекламі, існує система знижок: з двох куплених товарів повністю сплачується тільки вартість дорожчого товару, а інший надається безкоштовно. Якої суми достатньо, щоб оплатити покупку трьох товарів, якщо відома ціна кожного.

Вхідні дані: Три натуральних числа **a, b, c** – ціни трьох товарів ($1 \leq a, b, c \leq 10000$).

Вихідні дані: Вартість покупки.

Input 8 11 6 **Output** 17

Зрозуміло, що в таких умовах, поступаючи найвигідніше для нас, під акцію пропонуємо два дорожчих товари, а вартість всієї покупки буде рівна сумі найдорожчого і найдешевшого. Ось такий код.

```
a,b,c = map(int,input().split())
print (max(a,b,c)+min(a,b,c))
```

107. КОМПАКТ-ДИСКИ

Чисті компакт-диски продаються в трьох різних видах упаковок. Упаковка зі **100** дисків коштує **100** грн., з **20** дисків - **30** грн., а один окремих диск коштує **2** грн. Якої мінімальної суми має вистачити для покупки **n** таких дисків?

Вхідні дані: Кількість **n** ($n \leq 1000$) дисків, які потрібно купити.

Вихідні дані: Мінімальна сума, потрібна для покупки.

Input 123 **Output** 136

Пригадайте, як Ви поступали кожен раз, коли потрібно було купити декілька чистих компакт-дисків – безумовно, найвигідніше купити відразу сто дисків, а ті диски, які виявляться зайвими стануть в пригоді і заощадять Ваші фінанси у майбутньому.

Проте, не будемо поспішати і насамперед обчислимо в змінну **x** суму грн., які потрібно заплатити за диски, якщо купувати їх точно за правилами без найменшого авантюризму. Для цього потрібно знайти скільки буде упаковок кожного виду, послідовно розділивши значення **n** на **100** і на **20**, а після тупо відрахувати відповідну суму грн.

Але стоп. Уявіть, що диски в упаковці по одному вже закінчились, тоді ми будемо змушені замість них взяти ще одну пачку з **20** дисків. Сумарно дисків буде точно не менше **n**, а грошей **y** грн. потрібно буде заплатити більше, а може й менше ніж в першому випадку.

І на кінець, якщо диски купувати тільки в коробках по **100** дисків, тобто просто взяти на одну пачку більше таких дисків всього буде **z** грн., а інших упаковок не брати взагалі, то ще невідомо яка сума **x**, **y** чи **z** виявиться меншою. Звичайно, в двох останніх випадках ми маємо

виграш не тільки в коштах – їх буде менше, а й в дисках – їх буде більше.

```
n=int(input())
a,n=divmod(n,100)
b,c=divmod(n,20)
x=100*a+30*b+2*c
y=100*a+30*(b+1)
z=100*(a+1)
print(min(x,y,z))
```

7670. СТЕПАН І ПОХІД В МАГАЗИН

Від будинку Степана до першого магазину веде доріжка довжиною **a**, а до другого магазину веде доріжка довжиною **b**. Також існує доріжка, яка безпосередньо сполучає два магазини один з одним, довжиною **c**.

Допоможіть Степану обчислити мінімальну відстань, яку йому буде потрібно пройти, щоб відвідати обидва магазини і повернутися додому.

Степан завжди стартує зі свого будинку. Він повинен відвідати обидва магазини, переміщуючись тільки за наявними трьома доріжками, і повернутися назад додому. При цьому його абсолютно не бентежить, якщо йому доведеться відвідати один і той же магазин або пройти по одній і тій же доріжці більше одного разу. Єдине його завдання - мінімізувати сумарну пройдену відстань.

Вхідні дані: 3 цілих числа **a, b, c** ($1 \leq a, b, c \leq 10^8$) - довжини доріжок.

Вихідні дані: Мінімальна сумарна відстань.

Input 10 20 30 **Output** 60

Розглянемо всі можливі варіанти переміщення Степана і виберемо з них найменший. Позначимо будинок Степана точкою **C**, перший магазин точкою **B**, а другий точкою **A**, тоді можливі такі маршрути:

- C – a - B – c - A – c - B – a - C
- C – b - A – c - B – c - A – b - C
- C – a - B – a - C – b - A – b - C
- C – a - B – c - A – b - C

Додаємо значення відрізків в таких маршрутах, виконуємо найпростіші математичні спрощення і залишається знайти найменший результат з чотирьох значень.

```
a,b,c = map(int,input().split())
print (min(2*(a+b),2*(b+c),2*(a+c),a+b+c))
```

7375. БОЧКА

Використавши дві посудини ємкістю **3л** і **5л** потрібно набрати в столітрову бочку **m** літрів води, причому сумарна кількість переливань в бочку і з бочки має бути мінімальною. Наприклад, щоб набрати **7** літрів води: два рази виливаємо в бочку по **5л**, потім відливаємо один раз **3л**, всього три переливання.

Вхідні дані: Ціле невід'ємне число **m**. $0 \leq m \leq 100$.

Вихідні дані: Відповідь до задачі.

Input 7 **Output** 3

Пам'ятаєте цікаву задачу з математики: як набрати з крану **4л** води, якщо маємо тільки дві посудини ємкістю **3л** і **5л**? Міркуємо приблизно так: два рази набрати по **3л** і вилити в **5л**, тільки акуратно, щоб після другого переливання в **5л** залишився **1** літр води в посудині **3л**. Тепер переливаємо його до **5л** і доливаємо туди ще раз **3л**. Порахували кількість переливань?

Математична модель задачі передбачає знаходження в цілих числах коренів рівняння $5a+3b=n$, такий що сума $|a|+|b|$ - мінімальна. Тому оптимальне розв'язання можна знайти масою способів: пошук коренів повним перебором або методом динамічного програмування, а також, виявляється, використавши тільки команду розгалуження.

Насправді, у випадку якщо **M** кратне **5**, то найменша кількість переливань буде $K=M/5$. Розглянемо ще декілька випадків, коли значення **M** не ділиться на **5** націло, тобто остача від ділення $L=M\%5$ приймає значення **1..4**.

L=1, набираємо **K-1** раз по **5л** і додаємо два рази по **3л**, тобто відповідь буде **K+1**. Але є один крайній особливий випадок **M=1**. Щоб отримати **1л** набираємо два рази по **3л** і відливаємо **5л**, тобто відповідь буде **3**.

L=2, набираємо **K+1** раз **5л** і відливаємо **3л**, тобто відповідь буде **K+2**.

L=3, набираємо **K** раз по **5л** і додаємо **3л**, отже, відповідь **K+1**.

L=4, набираємо **K-1** раз по **5л** і додаємо три рази по **3л**, тобто відповідь буде **K+2**. Тут теж маємо особливий випадок **M=4**. Мінімальна кількість переливань буде **4** - набираємо три рази по **3л** і відливаємо **5л**.

Наступний код реалізує описаний алгоритм. Використовується єдина команда розгалуження, в якій описані всі можливі загальні і особливі випадки. Всього їх буде п'ять, так як деякі мають однакові відповіді. Код вийшов трохи громіздкий, зате дуже швидкий O(1).

```
n = int(input())
k = n//5
if n%5==0 : print(k)
elif n==1 : print(3)
elif n==4 : print(4)
elif (n%5==1 or n%5==3) : print(k+1)
else: print(k+2)
```

7330. ПОДІЛЬНІСТЬ НА 3

Розглянемо послідовність

1 12 123 1234 12345 123456 1234567 12345678
123456789 12345678910 1234567891011 ...

Підрахуйте, скільки елементів цієї послідовності з перших **n** діляться на **3**.

Вхідні дані: Натуральне число **n** ($1 \leq n \leq 2^{31}-1$).

Вихідні дані: Відповідь до задачі.

Input 4 **Output** 2

Перебірне розв'язання цієї задачі передбачає використання оператора циклу, якого ми поки що не розглядали, тим більше, що такий алгоритм не буде ефективним і подолає тільки 80% тестів. Тому логічно шукати евристики на відомому полі.

Очевидно, що аналізуючи подільність на **3**, достатньо розглядати не самі числа, а лише суму їх цифр. Отже, завдання можна переформулювати так: розглядаючи суми послідовних натуральних чисел **1+2+3+...+n**, знайти скільки серед них кратні **3**. Проводимо деякі обчислення для пошуку закономірності.

N	1	2	3	4	5	6	7	8	9	10	11
$\Sigma=1+2+\dots+n$	1	3	6	10	15	21	28	36	45	55	66
$\Sigma\%3$	1	0	0	1	0	0	1	0	0	1	0
$n\%3$	1	2	0	1	2	0	1	2	0	1	2
$n//3*2$	0	0	2	2	2	4	4	4	6	6	6
Відповідь	0	1	2	2	3	4	4	5	6	6	7

Дві з трьох послідовних сум діляться на **3** у випадку, коли остача від ділення значення **n** на **3** дорівнює **0** або **2**. Отже, відповідь до задачі можна записати формулою **$n//3*2$** , яку потрібно буде збільшити на **1** у випадку, якщо остача від ділення значення **n** на **3** дорівнює **2**.

Два коди на вибір: зрозумілий і короткий.

<pre>n=int(input()) k=n//3*2 if n%3==2 : k+=1 print(k)</pre>	<pre>n=int(input()) print(n//3*2+int(n%3==2))</pre>
--------------------------------------------------------------	-----------------------------------------------------

Спробуйте розв'язати:

Магазин	1473
Маршрутне таксі	2206
Поїздки на метро	7818