<https://python-book.softuni.bg/>

**Основи на програмирането с Python**

Забелязани НЕТОЧНОСТИ и грешки В КНИГАТА

В *Глава 1. Първи стъпки в програмирането, Работа с Python интерпретатора* в дадения пример се казва: „Опитайте да изчислите **1234567890 *234567890*****34567890**.” Не се разбира какво се има предвид – може би има пропуснат знак за аритметично действие между цифрите?

\* \* \*

В *Глава 2.1. Прости пресмятания*, заглавието на *Примера: лице на правоъгълник в равнината* не е точно. Всички плоски фигури, изучавани в Евклидовата геометрия, са равнинни – правоъгълници, квадрати, успоредници и пр.; неравнинни са обемните (куб, сфера, паралелепипед). Конкретният случай, обаче, е частен – страните на правоъгълника са успоредни на координатните оси. По-точно заглавие на примера би било: „Лице на правоъгълник със страни, успоредни на координатните оси”.

Аналогично, заглавието на „*Задача: лице и периметър на правоъгълник в равнината*” може да се преименува: „Лице и периметър на правоъгълник със страни, успоредни на координатните оси”.

\* \* \*

В *Задача: учебна зала* към Глава *2.2 Прости пресмятания с числа – изпитни задачи*, по-късият размер на залата е наречен веднъж „височина” , втори път – „широчина” (в „Идея за решение”). Би следвало да се именуват по един и същ начин. Освен това, по-големият размер обикновено е дължина, така че за работно място с планови размери 70 х 120 см, 70 см следва да е ширината, а не дължината (както е казано в текста).

-----

В *Задача: Дневна печалба* към Глава *2.2 Прости пресмятания с числа – изпитни задачи*, е сбъркана валутата в данъците (25% от 22837.5 долара са 5709.375 долара, а не лева – вж. Обяснението към първите входни данни).

\* \* \*

В *Задача: бонус точки* към Глава 3.1. *Прости проверки*, условието „Ако числото е по-голямо от 100, бонус точките са 20% от числото” следва да се замени с „Ако числото е по-голямо от 100, НО ПО-МАЛКО ИЛИ РАВНО НА 1000, бонус точките са 20% от числото”.

-----

В *Задача:* *сумиране на секунди* към Глава 3.1. *Прости проверки*, в Насоки и подсказки, примерният сорс код не е точен – два „if цикъла” са идентични.

\* \* \*

В *Задача:* *поспаливата котка Том* към Глава 3.2. *Прости проверки – изпитни задачи*, в Примерен вход и изход, има несъответствие между посочения изход за игра над нормата: "{H} hours and {M} minutes **more** for play" и дадения в таблицата резултат за 113 дни: „3 hours and 47 minutes for play” (липсва думата „more”). От друга страна, не е ясна логиката на обяснението защо, ако котката ще избяга ("Tom will run away"), понеже си недоспива, ще остава още време за игра ("…more for play")? По-логично би било да следва надпис, указващ нещо като „Времето за игра е превишено с … часа и … минути” (напр. „Playing time is overpassed with … hours and … minutes”). Аналогично, когато котката спи добре, вторият надпис би звучал по-логично така: „… hours and … minutes left for play”.

-----

В *Задача:* *фирма* към Глава 3.2. *Прости проверки – изпитни задачи*, условието не е добре обяснено. Не е ясно дали дните, в които служителите работят по проекта (90% от наличните) трябва да се закръглят или не. Освен това, при първите входни данни: 90 часа необходими за проекта, неразбираемо е какво значи 99 часа остават (съобщението на изхода)? Как да остават повече часове (за работа), отколкото са нужни? По-логично е да се отпечата съобщение от типа на: „При необходими за работа 90 ч, служителите имат налични 189 ч, т.е. 99 ч в излишък”. Или някакво подобно – по-разбираемо е.

\* \* \*

В *Пример: точка върху страна на правоъгълник* към Глава *4.1. По-сложни проверки*, в Решението се казва „x съвпада с x1 или x2 …” и „y съвпада с y1 или y2…”. Тъй като задачата е от областта на геометрията, терминът „съвпада” означава две точки да лежат една върху друга. Понеже това не е разглеждания случай и точките имат само равни координати, но НЕ съвпадат една с друга, правилно е да се формулира така:

„абсциса x е ***равна на*** x1 или x2 …” и „ордината y е ***равна на*** y1 или y2…”.

\* \* \*

В *Задача: билети за мач* към *Глава 4.2. По-сложни проверки – изпитни задачи*, в Примерен вход и изход неправилно е указано 24499.51 < 18000 вместо знака **>**; да се допълни, че 24499.51 лв. е цената на 49 билета.

\* \* \*

В  *Примера: къщичка* към Глава 6.1. Вложени цикли, в Насоки и подсказки, неправилно е именувана височината на триъгълния покрив като дължина (*roof\_length*). Освен това, за втория цикъл неправилно се казва: „Започваме с цикъл от 0 до n”, но всъщност е до (n // 2), както се вижда на следващия ред.

-----

В  *Примера: диамант* към Глава 6.1. Вложени цикли, в Насоки и подсказки, фразата:

„Разликата е, че вместо да намаляме стойността на left\_right с 1 към края на цикъла, ще я увеличаваме с 1 в началото на цикъла.” по-правилно да се редактира на: „Разликата е, че вместо да намаляваме стойността на left\_right с 1 към края на цикъла, ще я увеличаваме с 1 в началото на цикъла, като я нулираме преди цикъла.” (иначе не се получава верен отговор).

\* \* \*

В  *Задача: знак "Стоп"* към Глава 6.2. Вложени цикли – изпитни задачи, в Насоки и подсказки, пояснението: „Средната част от фигурата има начало // + \_, среда STOP! и край \_ + \\. Броят на долните черти \_ е (underscores - 5) / 2” не е вярно – следва да завърши с целочислено деление: (underscores - 5) **//** 2, защото трябва да се получи цяло число при последващото форматиране.

-----

В *Задача: стрелка* към Глава 6.2. Вложени цикли – изпитни задачи, в Насоки и подсказки, пояснението: „Броят на # е равен на outer\_dots и за това можем да използваме отново същата променлива:” да се смени с: „Броят на # е равен на outer\_dots **+ 1** и затова можем да използваме отново същата променлива:”.

-----

В *Задача: брадва* към Глава 6.2. Вложени цикли – изпитни задачи, в Насоки и подсказки, изречението: „Тъй като в този случай ни трябва само цялата част (от условието на задачата виждаме, че при вход 5 височината на дръжката на брадвата е 2), можем да използваме метода math.trunc(…) …” би трябвало да се коригира на: „Тъй като в този случай ни трябва само цялата част (от условието на задачата виждаме, че при вход 5 височината на дръжката на брадвата е 2), можем да използваме метода math.trunc(…) или т.нар. „кастване”: int(n / 2) …”, понеже следващото пояснение включва именно „кастване”, а не функцията math.trunc(…).

\* \* \*

В *Пример: изчисляване на факториел* към Глава 7.1. По-сложни цикли, не е довършено решението – липсва операторът: print(fact).

---

В *Пример: проверка за просто число. Оператор break* към Глава 7.1. По-сложни цикли, в описанието е казано, че се „създава булева променлива **is\_prime** с начална стойност True”, а в сорс кода тя е именувана: **prime**.

---

В *Задача: числа на Фибоначи* към Глава 7.1. По-сложни цикли, в Примерен вход и изход, за първото въведено число n = 10, коментарът трябва да е: F(**10**) = F(9) + F(8), а не F(11) = F(9) + F(8).

\* \* \*

В *Задача: генератор за тъпи пароли* към Глава 7.2. По-сложни цикли – изпитни задачи, неудачно е избран символът ***l*** за цяло число, тъй като при избрания шрифт на текста не се различава малката латинската буква „***l***” от цифрата „1” в сорс кода. Освен това, в предпоследния ред на дадения сорс код не е нужно да се добавя празен низ ' ' в текстовата променлива „**result**” , понеже тя вече е въведена при началното ѝ дефиниране.

----

В *Задача: специални числа* към Глава 7.2. По-сложни цикли – изпитни задачи, зададеното в задачата условие „N да се дели на всяка една от неговите цифри без остатък” не отговаря на показаните примери, в които за решение са приети комбинациите на всеки един делител – цяло число от 1 до 9, на който се дели даденото число; за 3 това са цифрите на числата 1 и 3, а за 16: 1, 2, 4 и 8, като същите не са цифри на самото число.

\* \* \*

В *Задача: суми от 3 числа* към Глава 8.1. Подготовка за практически изпит – част I, зададеният в сорс кода формат за печат: print(‘%d + %d = %d’, b, c, a) и др.под. не се разпечатва правилно в моя вариант на IDE PyCharm, но се отчита за верен в системата Judge. Защо?

----

В *Задача: правоъгълник със звездички в центъра* към Глава 8.1. Подготовка за практически изпит – част I, в сорс кода неправилно са дадени форматите на печат (в края на програмата):

if i == floor(num\_rows / 2):

 print((' ' \* (n - 2)) + '\*\*' + (' ' \* (n - 2)))

else:

 print(' ' \* (n \* 2 - 2))

print('%',, end=' ')

Правилно би било:

if i == floor(num\_rows / 2):

 print((' ' \* (n - 2)) + '\*\*' + (' ' \* (n - 2)), end='')

else:

 print(' ' \* (n \* 2 - 2), end='')

print('%')

----

В *Задача: генериране на правоъгълници* към Глава 8.1. Подготовка за практически изпит – част I, би следвало да се уточни, че изложените пояснения важат за лява правоъгълна (декартова) координатна система, т.е. с ординатна (вертикална) ос, насочена надолу. В противен случай (по подразбиране) се приема дясна декартова система, в която ординатната ос е насочена нагоре (при дясно насочена абсциса), а тогава дадените към задачата обяснения не са верни за знаците на ординатите:

–n <= ***top*** < ***bottom*** <= n

и

for ***bottom*** in range(***top*** + 1, n + 1): (вж. четвъртия вложен цикъл).

Да се доуточни, че страните ***top*** и ***bottom*** са съответно горната и долната страни на правоъгълника (за наблюдателя), а показаните на изхода (печата) координати на „правоъгълниците се задават чрез горния си ляв и долния си десен ъгъл.”. Без такова пояснение за посоката на осите обикновено се подразбира дясна декартова координатна система.

\* \* \*

В *Задача: магазин за цветя* към Глава 8.2. Подготовка за практически изпит – част II, обяснението: „Всички отстъпки важат след оскъпяването за празничен ден!” е подвеждащо, защото така се разбира, че отстъпките важат САМО след наличието на празничен ден, докато от третия пример в Примерен вход и изход се вижда, че те са налице и когато няма празничен ден.

\* \* \*

В *Задача: магически дати* към Глава 9.1. Задачи за шампиони – част I, непонятно защо е даден противоречив пример още в началото: първо се казва: „Дадена е дата във формат "дд-мм-гггг", напр. **26-09-2018**…”, а веднага след това е взета друга дата: „В нашия случай имаме 8 цифри: **17032007**…”. Би следвало да се използва една и съща дата.

----

В *Задача: пет специални букви* към Глава 9.1. Задачи за шампиони – част I, от условието не става ясно дали еднаквите букви се премахват САМО при изчисляване на „теглото”, оставайки в крайното разпечатване или и при „теглото”, и в крайната разпечатка не трябва да се съдържат повтарящи се букви (сравни резултатите в Примерен вход и изход).

\* \* \*

В *Задача: бикове и крави* към Глава 9.2. Задачи за шампиони – част II, в Обработване на входа е казано: „За входа на нашата задача имаме 3 реда: Секретното число. Броят желани бикове. Броят желани крави.”. Не е ясно какво се разбира под „брой желани” – не е ли всъщност брой „познати” (бикове и крави)?

\* \* \*

В *Примера: знак на цяло число* към Глава 10. Функции, подраздел „Функции с параметри (по-сложни функции)” е казано: „Следващата ни стъпка е да прочетем входното число и да извикаме новата функция:

print\_sign(input(„Enter number: ”)

При тестване в *Judge* системата обаче текстът („Enter number: ”) се оказва излишен – може би да се зададе просто:

print\_sign(n)

при въведено n като n = int(input()) ?

----

Аналогично на предходния пример, в *Примера: рисуване на запълнен квадрат* към същата глава, *Judge* системата не възприема текста „Enter size: ” в оператора

side = int(input(“Enter size:”))

(дава 100% грешка). При отстраняване на текста във вътрешните скоби решението е 100% вярно.

----

Аналогично на предходните примери, в *Примера: пресмятане на лицето на триъгълник* към същата глава, *Judge* системата не възприема текста „Enter length / height: ” в операторите за променливите **a** и **h**. Освен това, въпреки че променливите са дробни числа (тип ***float***), *Judge* системата изисква резултат без дробна част (**6** вместо **6.0**), иначе отчита грешка.

----

В *Задача: число към бройна система* към Глава 10. Функции, в Насоки и подсказки, неправилно е указано: „Вмъкваме остатъка от числото в началото на низа, представящ резултата.”; всъщност, трябва да се вмъква в края на низа (или да се обърне обратно първоначално полученият низ).

----

В *Задача: известия* към Глава 10. Функции, в инструкциите е казано: „Когато *message\_type* е ***warning*** да се чете само *message*.”, а в приведените оператори след това са дадени две променливи: ***operation*** и ***message***:

**def** show\_warning\_message(*operation, message*)

Би следвало да е

**def** show\_warning\_message(*message*)

----

В *Задача: числа към думи* към Глава 10. Функции, в условието е казано: „Ако числото не е съставено от три цифри, не трябва да се принтира.”, а в Примерен вход и изход са разпечатани числата 15 („fifteen”) и 9 („nine”), което противоречи на условието на задачата.

Освен това, в Насоки и подсказки неправилно е указано: „Можем първо да отпечатаме стотиците като текст - (числото / 100) % 10, след тях десетиците - (числото / 10) % 10 …” докато вярното е да се вземе целочислено деление на 100 и на 10, т.е.

за стотиците - (числото **//** 100) % 10, … десетиците - (числото **//** 10) % 10 …

Допълнително, ако числото е отрицателно (трицифрено), то в горния израз трябва да се вземе неговата абсолютна стойност.

Също така, от условията: „Вторият специален случай е когато числото, формирано от последните две цифри на входното число, е по-малко от 10 …" и „Третият специален случай е когато числото, формирано от последните две цифри на входното число, е по-голямо от 10 и по-малко от 20…” не се разбира къде е включено числото 10 – в първия или във втория специален случай.

Не на последно място следва да се обърне внимание на терминологията в книгата. Използваният технически жаргон на места е грубоват, неточен или невинаги понятен:

Така например, използван е терминът „***кодене***”, който не е позната и употребявана дума. В българския език имаме „*кодиране*”, определяно от тълковния речник на БАН като: „Преобразуване на текст, картографско изображение и под. в код за осъществяване на тайно предаване, съхранение на информация или за машинна обработка.”. Другият познат и използван в тази област термин е „програмиране” – звучи по-добре (и разбираемо) от „*кодене*”.

На места е използвано понятието „***къдрави скоби***” – непознато в математиката, където са дефинирани големи **{** **}**, средни **[** **]** и малки **(** **)** скоби. Като граматично понятие може да се използват „фигурни скоби” – за **{ }**, но не и „*къдрави*”.

В книгата се среша и техническият жаргон ***„чупене”*** (на програма или цикъл) – вероятно неточен превод от английския глагол “break” (вж. Глава 7.1. По-сложни цикли, забележката към Пример: оператор *break* в безкраен цикъл). На български не може да се каже така – чупи се кристална чаша. Програма или цикъл може да спира (прекъсва) или зацикля, не да се „чупи”.

В текстакъм Глава 6.1. Вложени цикли, в Обработка на входните данни, в първия абзац на текста към „Вложени цикли” се казва: „За всяко завъртане на външния цикъл, вътрешният се извърта целият. Това ***се случва*** по следния начин…”. Глаголът „случвам се” е неприложим в програмирането, освен ако не става дума за генериране на случайни числа. Случват се само случайни събития, които, по своята същност, са непредсказуеми (напр.,генериране на случайно число или стохастичен процес) и, вследствие на това – непланируеми и неконтролируеми. Когато обаче имаме коректно написана програма (сорс код), в нея стъпките не се „случват” (освен ако са допуснати грешки) – програмата се изпълнява както е замислена и реализирана от програмиста. Модното езиково клише „случва се” е крайно неуместно в програмирането.