# Лаб: Повторения с цикли – while-цикъл

Задачи за упражнение и домашно към курса ["Основи на програмирането" в СофтУни](https://softuni.bg/courses/programming-basics).

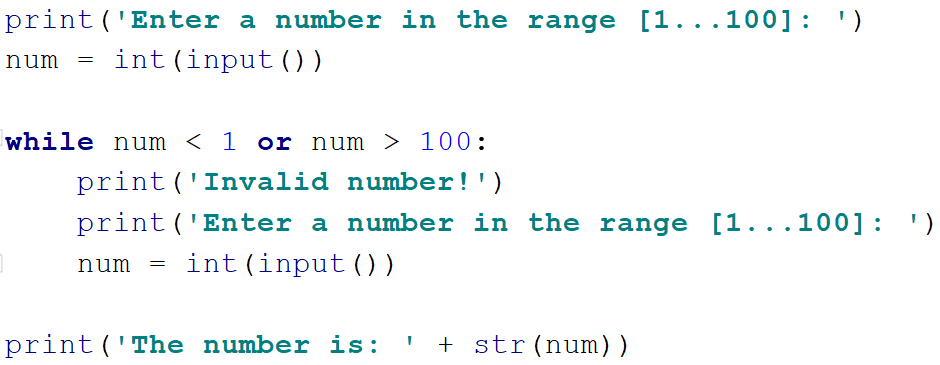
## Число в диапазона [1…100]

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 |

1. Създайте **нов проект** в съществуващото Visual Studio решение с име **"number\_in\_range"**.
2. Напишете решението на задачата. Можете да си помогнете с **кода от картинката по-долу**:



1. **Стартирайте** програмата и я **тествайте** с различни входни стойности.
2. **Тествайте** решението си в **judge системата**: [https://judge.softuni.bg/Contests/Compete/Index/1258#0](https://judge.softuni.bg/Contests/Compete/Index/1258)

## Завършване

Напишете програма, която изчислява **средната оценка** на ученик от цялото му обучение. На първия ред ще получите **името на ученика**, а на всеки следващ ред неговите годишни оценки. Ученикът преминава в следващия клас, **ако годишната му оценка е** **по-голяма или равна на 4.00**. **Ако оценката му е под 4.00**, той ще повтори класа.

При успешно завършване на **12-ти** клас да се отпечата:

"{**име на ученика**} graduated. Average grade: {**средната оценка от цялото обучение**}"

**Стойността трябва да бъде форматирана до втория знак** след десетичната запетая.

### Примерен вход и изход

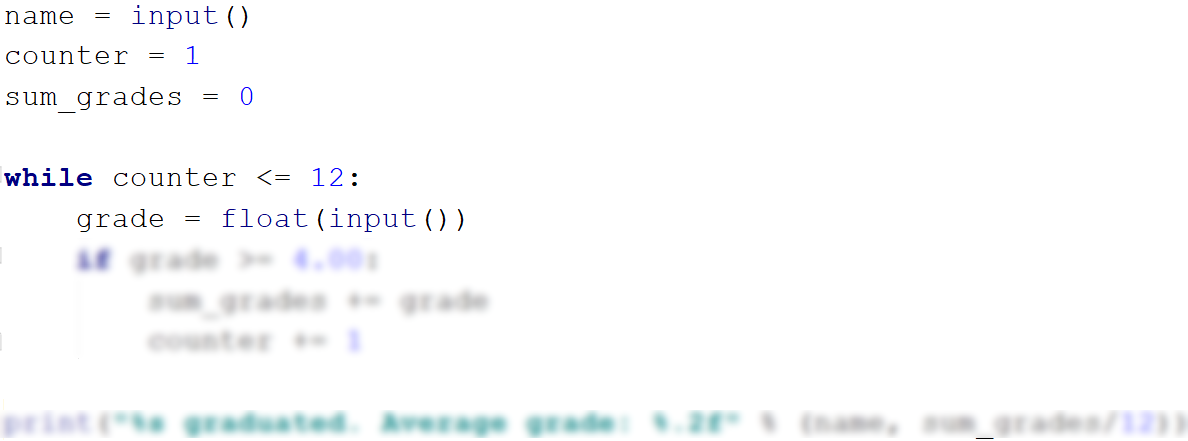
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| Pesho  4  5.5  6  5.43  4.5  6  5.55  5  6  6  5.43  5 | Pesho graduated. Average grade: 5.37 | Ani  5  5.32  6  5.43  5  6  5.5  4.55  5  6  5.56  6 | Ani graduated. Average grade: 5.45 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1258#1>

### Насоки

1. Създайте конструкция за while цикъл с условие на повтаряне докато ученикът не е завършил 12 клас. Създайте броят, с който да отчитате класовете на ученика.
2. На всяка повторение четете оценка
3. В зависимост от прочетената стойност за оценката, извършвайте действия, които определят дали ученикът минава в по-горен клас или не
4. Когато ученикът достигне 12 клас, прекъснете цикъла и сметнете средната му оценка, като разделите сумата на оценките, с които е преминал към по-горен клас на 12, за да получите средната оценка за обучението му.

Можете да си помогнете с кода от картинката, който е **нарочно замъглен**, за да помислите **как да решите задачата сами**.



## Завършване - част 2

Напишете програма, която изчислява **средната оценка** на ученик от цялото му обучение. На първия ред ще получите **името на ученика**, а на всеки следващ ред неговите годишни оценки. Ученикът преминава в следващия клас, **ако годишната му оценка е по-голяма или равна на 4.00**. **Ако ученикът бъде скъсан повече от един път, то той бива изключен и програмата приключва,** като се отпечатва **името на ученика и в кой клас бива изключен**.

При успешно завършване на **12-ти** клас да се отпечата :

"{**име на ученика**} graduated. Average grade: {**средната оценка от цялото обучение**}"

**В случай, че ученикът е изключен от училище, да се отпечата:**

"{**име на ученика**} has been excluded at {**класа, в който е бил изключен**} grade"

**Стойността трябва да бъде форматирана до втория знак** след десетичната запетая.

### Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| Gosho  5  5.5  6  5.43  5.5  6  5.55  5  6  6  5.43  5 | Gosho graduated. Average grade: 5.53 | Mimi  5  6  5  6  5  6  6  2  3 | Mimi has been excluded at 8 grade |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1258#2>

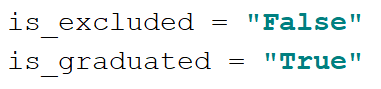
### Насоки:

Използвайте решението от предната задача и добавете нужната функционалност

1. Създайте помощен брояч за ниските оценки на ученика и променлива от [булев](https://docs.python.org/2.0/ref/lambda.html) тип, в която да пазим информация дали ученикът е изключен или не. Присвоете й първоначална стойност “False”.
2. Създайте конструкция на while цикъл, който да повтаря действия докато ученикът **не е 12 клас** или **не е изключен**. Комбинирайте условия, като използвате броячът за броя на класовете и създадената **булевата** **променлива** с логическия оператор “not” преди нея. По дефиниция, когато използваме булеви променливи в условна конструкция или цикъл, интепретаторът проверява дали стойността им не е “True”. В нашия случай, чрез операторът not ние отричаме тази стойност, т.е. цикъла ще повтаря действията си докато тази стойност е “False”. Вътре в конструкцията трябва да присвоим стойност “True”, когато лошите оценки на ученика станат 2 на брой и това ще прекъсне повтарянето на цикъла.

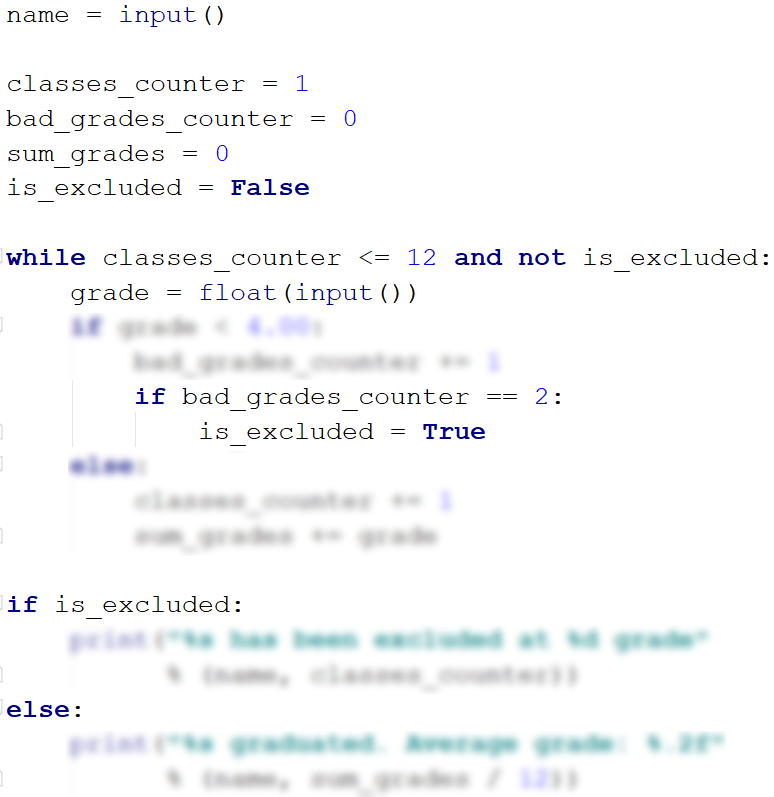
**Важно нещо, което трябва да запомним е, че булевите стойнсти “**True**” и “**False**” не са текст!**

**Ако присвоим следните стойности (в кавички) на дадена променлива, те ще бъдат считани за текст:**



1. В зависимост от направените изчисления и взетите решения в тялото на цикъла, принтирайте подходящия изход спрямо условието

Помогнете си използвайки картинката по-долу:



## Редица цели числа

Напишете програма, която чете **цели числа**, **докато** не получи командата **"**END**"**. Принтирайте **най-голямото** и **най-малкото** число сред въведените.

### Примерен вход и изход

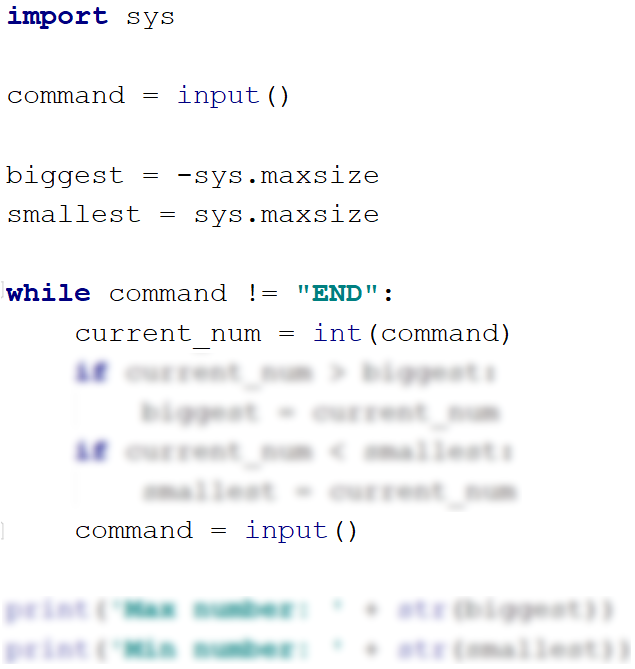
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| 10  20  304  0  50  END | Max number: 304  Min number: 0 | 250  5  2  0  100  1000  END | Max number: 1000  Min number: 0 |

**Тествайте** решението си в **judge системата**: [https://judge.softuni.bg/Contests/Compete/Index/1258#3](https://judge.softuni.bg/Contests/Compete/Index/1258%233)

### Насоки:

1. Прочетете команда от потребителя
2. Създайте контролни променливи за най-голямото и най-малкото число, което ще бъде прочетено от конзолата. Вмъкнете за използване библиотеката sys и от нея присвоете стойностите от резултата на sys.maxsize. Присвоявайки най-голямата и най-малката възможна стойност на тези променливи, можем да гарантираме, че всяка прочетена стойност ще е валидна и съответно по-голяма или по-малка от възможните в комютъра.
3. Създайте конструкция на while цикъл, в който да повтаряте действия докато прочетената команда е различна от текста “END”. В тялото на цикъла превърнете стойността на прочетената команда в числова
4. **Сравнете** **стойностите на текущото число с контролните** **променливи** и им присвоете стойността, ако е по-голямо или по-малко от текущите им стойности
5. Прочетете отново команда на края на блока от код в цикъла, за да продължите операциите със следващото число

Помогнете си използвайки кода от картинката по-долу:



## Стъпки

Габи иска да започне здравословен начин на живот и си е поставила за цел да върви **10 000 стъпки всеки ден**. Някои дни обаче е много уморена от работа и ще иска да се прибере преди да постигне целта си. Напишете програма, която **чете от конзолата по колко стъпки изминава** тя всеки път като излиза през деня и **когато постигне целта си да се изписва** "**Goal reached! Good job!**"

Ако иска да се **прибере преди това**, тя ще въведе **командата** "Going home" и **ще въведе стъпките, които е извървяла докато се прибира**. След което, ако не е успяла да постигне целта си, на конзолата трябва да се изпише: "{разликата между стъпките} more steps to reach goal."

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 1000  1500  2000  6500 | Goal reached! Good job! | 1500  300  2500  3000  Going home  200 | 2500 more steps to reach goal. |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 1500  3000  250  1548  2000  Going home  2000 | Goal reached! Good job! | 125  250  4000  30  2678  4682 | Goal reached! Good job! |

**Тествайте** решението си в **judge системата**: [https://judge.softuni.bg/Contests/Compete/Index/1258#4](https://judge.softuni.bg/Contests/Compete/Index/1258%234)

## Редица числа 2k+1

Напишете програма, която чете число n, въведено от потребителя,и отпечатва **всички числа ≤ n от редицата**: 1, 3, 7, 15, 31, …. Всяко следващо число се изчислява като умножим **предишното** с **2** и добавим **1**.

### Примерен вход и изход

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

**Тествайте** решението си в **judge системата**: [https://judge.softuni.bg/Contests/Compete/Index/1258#5](https://judge.softuni.bg/Contests/Compete/Index/1258%235)

**Подсказки**:

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

## Преместване

На осемнадесетия си рожден ден на Хосе взел решение, че ще се изнесе да живее на квартира. Опаковал багажа си в **кашони** и намерил подходяща обява за апартамент под наем. Той започва да пренася своя багаж **на части**, защото не може да пренесе целия наведнъж. Има ограничено **свободно пространство** в новото си жилище, където може да разположи вещите, така че мястото да бъде подходящо за живеене.

Напишете **програма, която изчислява свободния обем от жилището на Хосе, който остава след като пренесе багажа си.**

**Бележка: Един кашон е с точни размери: 1m. x 1m. x 1m.**

**Вход**

Потребителят въвежда следните данни на отделни редове:

1. **Широчина на свободното пространство - цяло число в интервала [1...1000]**
2. **Дължина на свободното пространство - цяло число в интервала [1...1000]**
3. **Височина на свободното пространство - цяло число в интервала [1...1000]**
4. На следващите редове **(до получаване на команда "Done") -** **брой кашони, които се пренасят в квартирата - цели числа в интервала [1...10000];**

**Програмата трябва да приключи прочитането на данни при команда "Done"** **или ако свободното място свърши.**

**Изход**

Да се **отпечата** на конзолата **един** от следните редове:

* Ако стигнете до командата **"Done"** и има още свободно място:

**"{брой свободни куб. метри} Cubic meters left."**

* Ако свободното място свърши преди да е дошла команда **"**Done**":**

**"No more free space! You need {брой недостигащи куб. метри} Cubic meters more."**

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснение** |
| 10  10  2  20  20  20  20  122 | No more free space! You need 2 Cubic meters more. | 10 \* 10 \* 2 = **200 кубични метра.** 20 + 20 + 20 + 20 + 122 = **202** **кубични метра.** 200 - 202 = **2 недостигащи** кубични метра |
| 10  1  2  4  6  **Done** | 10 Cubic meters left. | 10 \* 1 \* 2 = **20 кубични метра.** 4 + 6 = **10** **кубични метра.** 20 - 10 = **10 кубични метра.** |

**Тествайте** решението си в **judge системата**: [https://judge.softuni.bg/Contests/Compete/Index/1258#5](https://judge.softuni.bg/Contests/Compete/Index/1258#6)