# Лаб: Повторения с цикли – while-цикъл

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

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1014>.

## Число в диапазона [1…100]

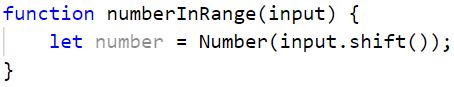
Напишете програма, която въвежда цяло положително **число** n **в диапазона [1…100]**. При въвеждане на число извън посочения диапазон, да се отпечата съобщение за грешка и **да се въведе ново число**.

### Примерен вход и изход

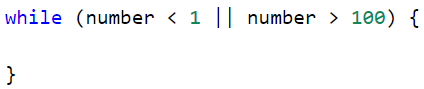
|  |
| --- |
| **Вход / Изход** |
| **35**  The number is: 35 |
| **105**  Invalid number!  **0**  Invalid number!  **-200**  Invalid number!  **77**  The number is: 77 |

**Насоки**

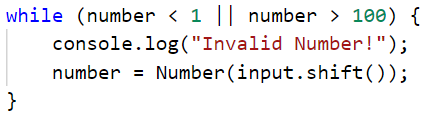
1. Създайте **нов js файл** в съществуващата папка с подходящо име например: **"Number-In-Range"**.
2. **Извадете входните данни от аргумента input в искания вид – число number.**



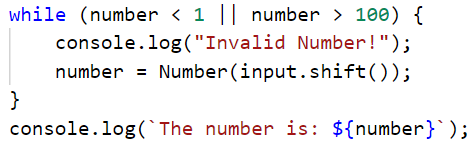
1. Направете един while цикъл, който да се повтаря **докато** съответното число number **е по-малко от 1 или по-голямо от 100.**



1. Ако програмата **влезе в тялото на цикъла,** то това ще означава, че сте получили число, което не е в диапазона [1...100]. В тялото на цикъла първо печатайте "Invalid number!", след което **извадете ново число** и **присвоете стойността му на променливата** number**.**



1. **След тялото на цикъла** отпечатайте промеливата number в желания формат. Тя ще има стойност **между 1 и 100**, понеже програмата ще е **излезнала от цикъла.**



1. **Стартирайте** програмата с [Ctrl+F5] и я **тествайте** с различни стойности.

## Редица числа 2k+1

Напишете програма, която чете число n, въведено от потребителяи 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 |

**Насоки**

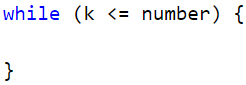
1. Вземете от **input** число.



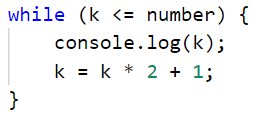
1. **Създайте** променлива, която ще е брояч и има **първоначална стойност 1**.



1. **Създайте while** цикъл, който се повтаря докато **брояча е по-малък** от числото, което сте прочели от конзолата.



1. При всяко повторение на цикъла **принтирайте стойноста на брояча** и му **прибавяйте дадената стойност**.



## Баланс по сметка

Напишете програма, която пресмята колко общо пари има в сметката, след като направите определен брой вноски. На първия ред ще получите **колко вноски трябва да се направят**. На всеки следващ ред ще получавате сумата, която трябва да внесете в сметката, **докато не се достигне броя вноски**. При всяка получена сума на конзолата трябва да се извежда **"Increase: "** + сумата и тя да се **прибавя в сметката**. Ако получите число **по-малко от 0** на конзолата трябва да се изведе **"Invalid operation!"** и **програмата да приключи**. Когато програмата приключи трябва да се принтира **"Total: "** + общата сума в сметката закръглена до втория знак след десетичната запетая.

### Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| 3  5.51  69.42  100 | Increase: 5.51  Increase: 69.42  Increase: 100  Total: 174.93 | 5  120  45.55  -150 | Increase: 120  Increase: 45.55  Invalid operation!  Total: 165.55 |

**Насоки**

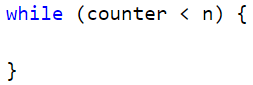
1. Извадете число от аргумента на функцията, което определя, колко вноски трябва да се направят.



1. Създайте **две променливи**, една променлива, за да пазите сумата, която е внесена и една променлива, която ще е брояч.



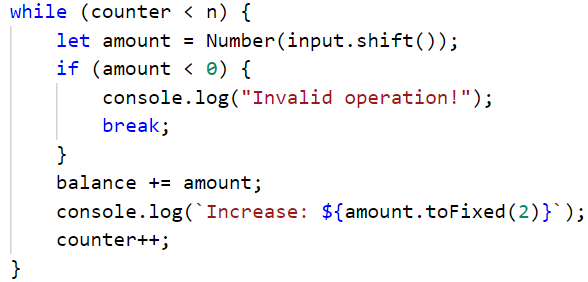
1. Създайте **while** цикъл, който ще продължава, докато брояча е по-малък от броя вноски.



1. При всяко повтаряне на цикъла ще вземаме сума от аргумента на функцията **(число)**.



1. Създайтепроверка, **която проверява, дали полученото число е отрицателна сума** и ако е така принтирайте на конзолата нужното съобщение и прекратявате цикъла, ако сумата е положителна, **увеличавайте баланса по сметката**, принтирайте на конзолата с колко сте увеличили баланса и **увеличавайте брояча с едно.**



1. След цикъла принтирайте тоталния баланс.



## Редица цели числа

Напишете програма, която изважда **числа от input**, **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 |

**Насоки**

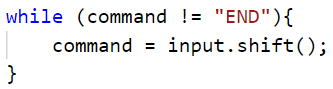
1. Създайте **две променливи, като променливата, в която ще пазите най-голямото число започва от възможно най-малкото число, а променливата, в която ще пазите най-малкото число започва от възможно най-голямото число.**



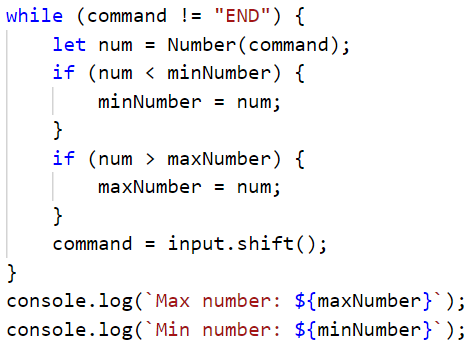
1. Създайте **променлива, която да взема команда от input.**



1. Създайте **while, който продължава, докато не получи вход END и на всяка итерация прочита нов вход след логиката в цикъла.**



1. **Ако входа не е END го преобразувайте в число и проверявайте, дали стойността е по-голяма и по-малка от променливите, които създадохте и ако е ги презаписвайте.** След като получите вход **END**, отпечатайте **след цикъла** нужния изход.



## 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!**"

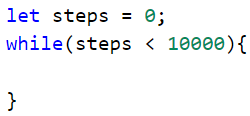
Ако иска да се **прибере преди това**, тя ще въведе **командата** "Going home" и **ще въведе стъпките, които е извървяла докато се прибира**. След което, ако не е успяла да постигне целта си, на конзолата трябва да се изпише: "{разликата между стъпките} more steps to reach goal."

### Примерен вход и изход

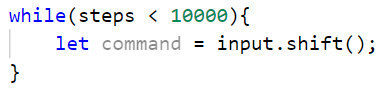
|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 1000  1500  2000  6500 | Goal reached! Good job! | 1500  300  2500  3000  Going home  200 | 2500 more steps to reach goal. |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 1500  3000  250  1548  2000  Going home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

**Насоки**

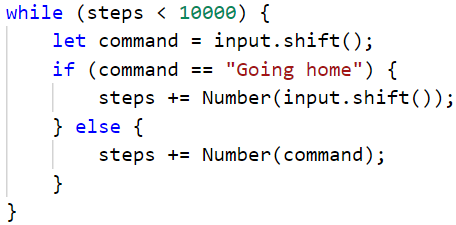
1. Създайтепроменлива, чрез която **ще следите, колко стъпки е извървяла Габи** и while, който ще продължава, **докато Габи не достигне целта си**.



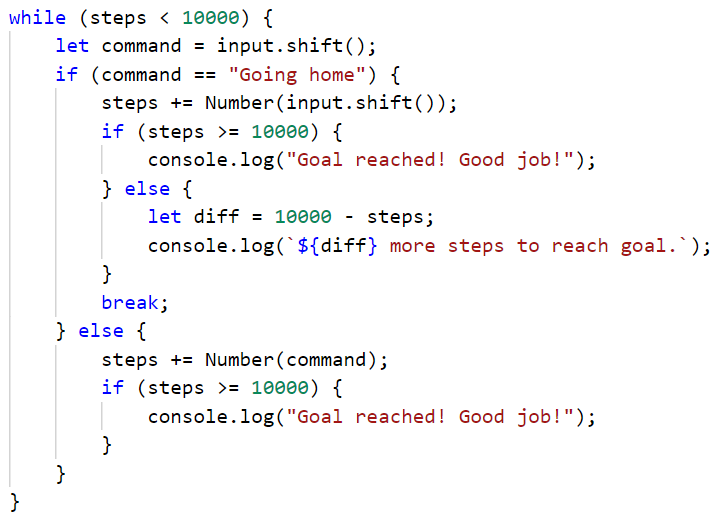
1. При всяко повторение на цикъла взимайте **по един елемент от аргумента на функцията**.



1. Проверете, дали полученият елемент е Going home и ако е, **прибавяйте стъпките, които е извървяла до тях**, в противен случай, **прибавяйте стъпките, които е извървяла**.



1. В случая, в който Габи се прибира, проверете, дали е достигнала целта си, ако не е успяла **намерете разликата между нейната цел и стъпките, които е успяла да извърви**, отпечатайте нужното съобщение и след това прекъснете цикъла чрез командата break.



## 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**}"

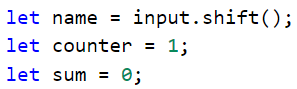
**Стойността трябва да бъде форматирана до втория знак** след десетичната запетая.

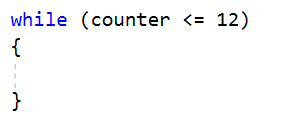
### Примерен вход и изход

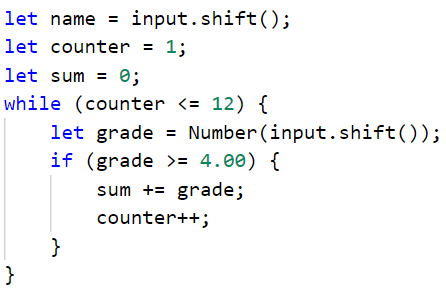
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| Pesho  4  5.5  6  5.43  4.5  6  5.55  5  6  6  5.43  5 | Pesho graduated. Average grade: 5.37 | Ani  5  5.32  6  5.43  5  6  5.5  4.55  5  6  5.56  6 | Ani graduated. Average grade: 5.45 |

**Насоки**

1. Извадете от аргумента на функцията **името**. Създайте **две помощни променливи** – една променлива **брояч** **с начална стойност 1** и една променлива с начална стойност 0, в която ще държим **сумата от всички оценки** на дадения ученик.



1. Направете **един** while **цикъл**, който да се повтаря, **докато брояча е по-малък или равен на 12**. 
2. **В тялото на цикъла четете оценката на ученика за съответния клас.**
3. След това направете **една проверка, дали въведената оценка е по-голяма от 4.00** и **ако е**, **я добавете към помощната променлива за сума**, която направихте в началото. **Увеличете** **стойността на брояча с 1**, понеже **щом оценката е 4 или повече**, **то ученикът е преминал в следващия клас**.



1. **След тялото на** while **цикъла**, изчислете **средната оценка** **от цялото обучение**, като разделите **сумата от всички оценка на 12.**



1. **Принтирайте** резултата в желания формат.



## 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

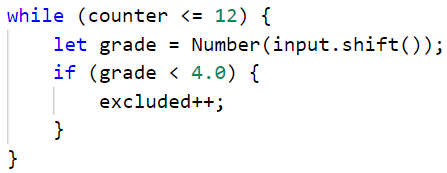
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| Gosho  5  5.5  6  5.43  5.5  6  5.55  5  6  6  5.43  5 | Gosho graduated. Average grade: 5.53 | Mimi  5  6  5  6  5  6  6  2  3 | Mimi has been excluded at 8 grade |

**Насоки**

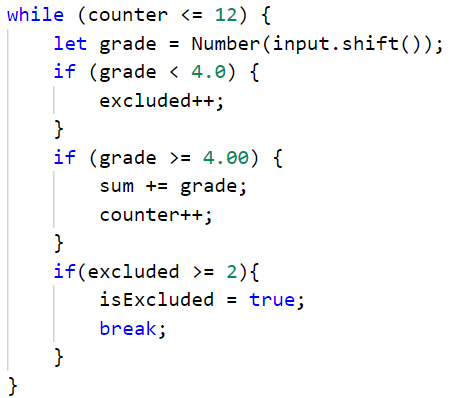
1. Използвайте решението от предната задача и добавете нужната функционалност:
2. Направете **още две помощни променливи в началото**, които да следят, **дали ученика трябва да бъде изключен**. Едната променлива ще е **брояч** и трябва да е с **първоначална стойност нула**. С нея ще следим, **колко пъти ученикът е трябвало да повтаря**. Другата променлива трябва да е **с началната стойност false**.



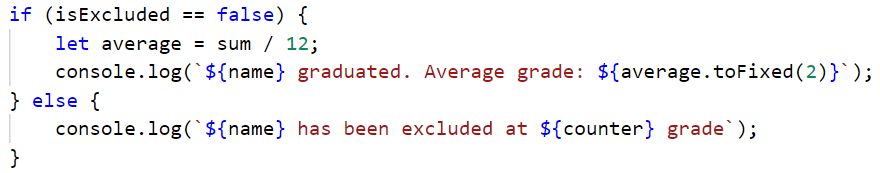
1. **В тялото на while цикъла** проверявайте, **дали съответната оценка е по-ниска от 4.00** и ако е, **увеличавайте брояча с 1**.



 4. Направете **още една проверка, дали ученикът е бил изключван 2 пъти** (като използвате съответния брояч) и ако е, **променете стойността на булевата променлива на true и излезте от цикъла**.



1. След тялото на цикъла, **печатайте изхода в зависимост от това, дали ученикът е бил изключен или не**.



# Примерна изпитна задача

## \*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**":**

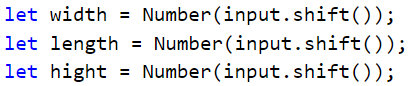
**"No more free space! You need {брой недостигащи куб. метри} Cubic meters more."**

**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.** |

**Насоки**

1. Извадете от аргумента на функцията три цели числа, които са **широчина, дължина и височина**.



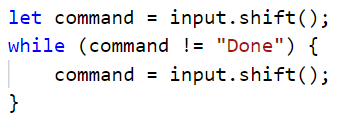
1. Пресметнете, **колко кубични метра е апартамента**.



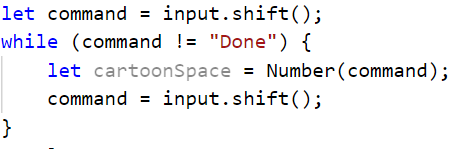
1. Създайтедве променливи, една за **пространството, което заемат кашоните** и една, в която ще запазите **разликата между пространството, което заемат кашоните и пространството на апартамента**.



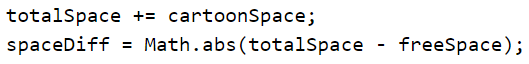
1. Създайте **while** цикъл, който се повтаря до получаване на команда **DONE** и при всяко повторение се взема по един елемент от аргумента.



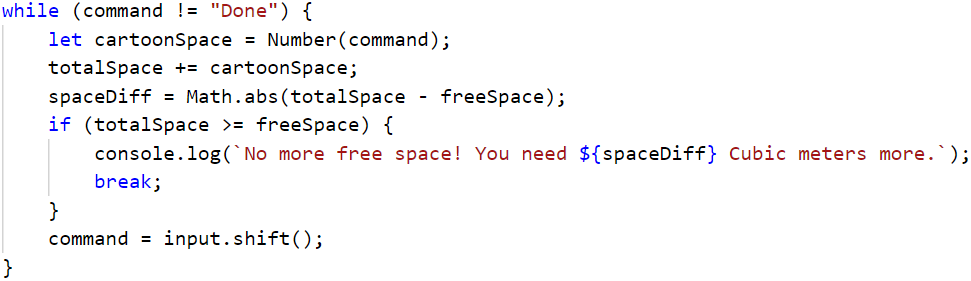
1. Ако командата не е **DONE** променете типа на командата в **цяло число**.



1. Добавете пространството, което заемат внесените кашони към по-рано създадената променлива и пресметнете разликата.



1. Ако **напълните или пренапълните апартамента**, принтирайте на конзолата съобщение и **спрете цикъла.**



1. Ако цикъла приключи с команда **DONE** принтирайте на конзолалата съобщението.

