# 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

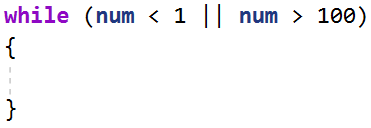
|  |
| --- |
| **Вход / Изход** |
| **35**  The number is: 35 |
| **105**  Invalid number!  **0**  Invalid number!  **-200**  Invalid number!  **77**  The number is: 77 |

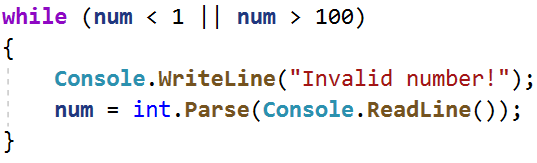
### Насоки

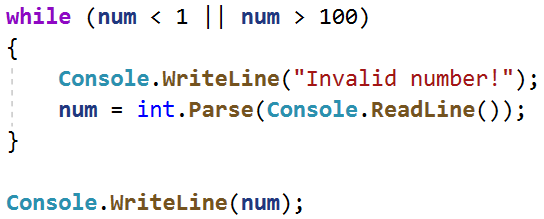
1. Създайте **нов проект** в съществуващото Visual Studio решение с име **"Number-In-Range"**.
2. **Прочетете входните данни от конзолата в искания вид – едно цяло число number.**



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 потребителят е въвел число, което не е в диапазона [1...100]. В тялото на цикъла първо печатайте "Invalid number!", след което **прочетете ново число** и **присвоете стойността му на променливата** number**.**
2. **След тялото на цикъла** отпечатайте променливата num в желания формат. Тя ще има стойност **между 1 и 100**, понеже програмата ще е **излязла от цикъла.**



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

## Парола

Напишете програма, която първоначално прочита име и парола на потребителски профил. След това чете парола за вход, при въвеждане на грешна парола, потребителя да се подкани да въведе нова парола.

### Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| Nakov  1234  pass  1324  1234 | Welcome Nakov! | Gosho  secret  secret | Welcome Gosho! |

### Насоки

1. Инициализирайте две променливи username и password, които ще съдържат потребителското име и паролата:



1. Инициализирайте променлива input, която ще държи въведената от потребителя парола за вход:



1. В while цикъл, до въвеждане на валидна парола, четете нова:



1. Когато се въведе **валидна парола** **принтирайте съобщението за успешен вход**:



## 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 |

### Насоки

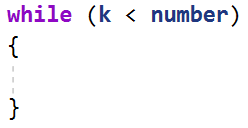
1. Прочетете от конзолата цяло число.



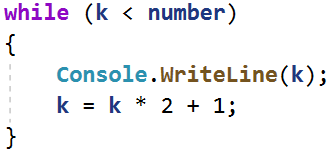
1. **Създайте** променлива от тип **цяло число**, която ще е брояч и има **първоначална стойност 1**.



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 стойността 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 |

## Най-голямо число

Напишете програма, която чете n-на брой **цели числа** (n > 0), въведени от потребителя, и намира **най-голямото** измежду тях. Първо се въвежда броят числа n, а след това самите n числа, по едно на ред.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2  100  99 | 100 | 3  -10  20  -30 | 20 | 4  45  -20  7  99 | 99 | 1  999 | 999 | 2  -1  -2 | -1 |

## Най-малко число

Напишете програма, която чете n-на брой **цели числа** (n > 0), въведени от потребителя и намира **най-малкото** измежду тях. Първо се въвежда броят числа n, а след това самите n числа, по едно на ред.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2  100  99 | 99 | 3  -10  20  -30 | -30 | 4  45  -20  7  99 | -20 | 1  999 | 999 | 2  -1  -2 | -2 |

## 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 |

## 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 |

# 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.** |

## 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/1014#4>

## Чаша с вода

Всеки офис си има диспенсър за вода. Напишете програма, която отчита дали служител от офиса е успял да напълни чашата си успешно.

Диспенсърът има три бутона :

* Easy **- лесен (50 милилитра)**
* Medium **- среден (100 милилитра)**
* Hard **- силен (200 милилитра)**

Ще получите **обемът на чашата** и на всеки следващ ред **кой бутон е бил натиснат**. **Ако чашата се напълни или прелее програмата приключва.**

Ако чашата прелее отпечатайте **колко вода е била излята**. При успешно напълване отпечатайте **броят на натисканията на бутона**.

### Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| 500  Hard  Medium  Easy  Easy  Medium | The dispenser has been tapped 5 times. | 600  Hard  Medium  Medium  Easy  Hard | 50ml has been spilled. |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1014#6>

## Монети

*Изпитна задача от практически приемен изпит на* ***7 Януари 2018****. Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/911#10)*.*

Производителите на вендинг машини искали да направят машините си да връщат възможно **най-малко монети ресто**. Напишете програма, която приема **сума** – **рестото**, което трябва да се върне и изчислява **с колко най-малко монети може да стане това**.

### Примерен вход и изход:

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 1.23 | 4 | Рестото ни е 1 лев и 23 стотинки. Машината ни го връща с 4 монети: монета от 1 лев, монета от 20 стотинки, монета от 2 стотинки и монета от 1 стотинка. |
| 2 | 1 | Рестото ни е 2 лева. Машината ни го връща с 1 монета от 2 лева. |
| 0.56 | 3 | Рестото ни е 56 стотинки. Машината ни го връща с 3 монети: монета от 50 стотинки, монета от 5 стотинки и монета от 1 стотинка. |
| 2.73 | 5 | Рестото ни е 2 лева и 73 стотинки. Машината ни го връща с 5 монети: монета от 2 лева, монета от 50 стотинки, монета от 20 стотинки, монета от 2 стотинки и монета от 1 стотинка. |

## Торта

*Изпитна задача от практически приемен изпит на* ***7 Септември 2017****. Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/750#3)*.*

Поканени сте на 30-ти рожден ден, на който рожденикът черпи с огромна торта. Той обаче не знае **колко парчета могат да си вземат гостите от нея**. Вашата задача е да напишете програма, която изчислява **броя на парчетата**, които гостите са взели, преди тя да свърши. Ще получите **размерите на тортата** (широчина и дължина – цели числа в интервала [1...1000]) и след това на всеки ред, до получаване на командата "STOP" или **докато не свърши тортата**, броят на парчетата, които гостите вземат от нея.

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

* "{брой парчета} pieces are left." - ако стигнете до STOP и не са свършили парчетата торта
* **"No more cake left! You need {брой недостигащи парчета} pieces more."**

### Примерен вход и изход

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 10  10  20  20  20  20  21 | No more cake left! You need 1 pieces more. |
| 10  2  2  4  6  STOP | 8 pieces are left. |