# Упражнение: Повторения с цикли – while-цикъл

## Най-голям общ делител

Напишете програма, която чете две цели положителни числа a и b, въведени от потребителя, и изчислява и отпечатва **най-големият им общ делител (НОД)**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2416 | 8 | 6718 | 1 | 159 | 3 | 10088 | 4 | 1010 | 10 |

**Подсказка**: имплементирайте **алгоритъма на Евклид**: [https://bg.wikipedia.org/wiki/алгоритъм-на-Евклид](https://bg.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D1%8A%D0%BC_%D0%BD%D0%B0_%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4).

Можете да си помогнете с кода от картинката по-долу:



## Старата Библиотека

Ани отива до родния си град след много дълъг период извън страната. Прибирайки се вкъщи тя вижда старата библиотека на баба си и си спомня за любимата си книга. Помогнете на Ани, като напишете програма в която тя въвежда търсената от нея **книга**(**текст**) и **капацитета** **на** **библиотеката**(**цяло число**). **Докато Ани не намери любимата си книга или не провери всички в библиотеката, програмата трябва да чете всеки път на нов ред името на всяка следваща книга (текст).**

* **Ако не открие книгата да се отпечата на два реда:**
* **"The book you search is not here!"**
* **"You checked {брой} books.**
* **Ако открие книгата си се отпечатва един ред:**
	+ **"You checked {брой} books and found it."**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| Troy8StrongerLife StyleTroy | You checked 2 books and found it. | Книгата която Ани търси, в случая е Troy, а библиотеката съдържа 8 книги. Първата е Stronger, втората е Life Style, третата книга е търсената – Troy и програмата приключва. |
| The Spot4Hunger GamesHarry PotterTorrontoSpotify | The book you search is not here!You checked 4 books. | Книгата, която търси Ани е "The Spot".Библиотеката съдържа 4 книги.Първата е Hunger Games, втората Harry Potter, третата Torronto, а четвъртата Spotify. Понеже няма повече книги в библиотеката четенето на имена приключва. Ани не намери книгата, която търсеше.  |
| Bourne32True StoryForeverMore SpaceThe GirlSpaceshipStrongestProfitTrippleStellaThe MatrixBourne | You checked 10 books and found it. |  |

1. Прочетете входните данни от конзолата.
2. Направете **while цикъл**, в които **четете** **по една книга всеки път**, докато книгата не е **намерена** или докато не **свърши капацитета на библиотеката**. Можете да си помогнете със следния код, които нарочно е замъглен за да помислите какво трябва да се допише:



## Подготовка за изпит

Напишете програма, в която Марин решава задачи от изпити **докато** **не** **получи** съобщение **"Enough"** отлектораси. При всяка решена задача той получава оценка. **Програмата трябва да приключи прочитането на данни при команда "Enough"** **или ако Марин получи определеният брой незадоволителни оценки.**
**Незадоволителна е всяка оценка, която е по-малка или равна на 4.**

### Вход

* **На първи ред - брой незадоволителни оценки - цяло число в интервала [1…5]**
* **След това многократно се четат по два реда:**
	+ **Име на задача - текст (низ)**
* **Оценка - цяло число в интервала [2…6]**

# Изход

* Ако Марин стигне до командата **"Enough", отпечатайте на 3 реда:**
* **"Average score: {средна оценка}"**
* **"Number of problems: {броя на всички задачи}"**
	+ **"Last problem: {името на последната задача}"**
* **Ако получи определеният брой незадоволителни оценки:**
* **"You need a break, {брой незадоволителни оценки} poor grades."**

**Средната оценка да бъде форматирана до втория знак след десетичната запетая.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3Money6Story4Spring Time5Bus6Enough | Average score: 5.25Number of problems: 4Last problem: Bus | Броя на позволени незадоволителни оценки е 3. Първата задача се казва Money, оценката на Марин е 6.Втората задача е Story, оценката на Марин е 4.Третата задача е Spring Time, оценката на Марин е 5.Четвъртата задача е Bus, оценката на Марин е 6.Следващата команда е Enough, програмата приключва.Средна оценка: 21 / 4 = 5.25Брой решени задачи: 4Последна задача: Bus |
| **Вход** | **Изход** | **Обяснения** |
| 2Income3Game Info6Best Player4 | You need a break, 2 poor grades. | Броят незадоволителни оценки е 2.Първата задача е Income, оценката на Марин е 3.Втората задача е Game Info, оценката на Марин е 6.Третата задача е Best Player, оценката на Марин е 4.Марин достигна допустимия брой незадоволителни оценки, време е за почивка. |

## Почивка

Джеси е решила да събира пари за екскурзия и иска от вас да ѝ помогнете да разбере **дали ще успее да събере необходимата сума**. **Тя спестява** или **харчи част от** **парите** си **всеки ден**. Ако иска да **похарчи повече от наличните си пари**, то тя ще похарчи **всичките и ще ѝ останат 0 лева**.

### Вход

От конзолата се четат:

* **Пари нужни за екскурзията** - **реално** **число в интервала [1.00.. .25000.00]**
* **Налични пари** - **реално** **число в интервала [0.00... 25000.00]**

**След това многократно се четат по два реда:**

* **Вид действие – текст с възможности "spend" и "save".**
	+ **Сумата, която ще спести/похарчи - реално число в интервала [0.01… 25000.00]**

# Изход

Програмата трябва да приключи при следните случаи:

* Ако **5 последователни дни** Джеси **само харчи, на конзолата да се изпише:**
* **"You can't save the money."**
* **"{Общ брой изминали дни}"**
* **Ако** **Джеси събере парите за почивката** на **конзолата се изписва:**
* **"You saved the money for {общ брой изминали дни} days."**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 20001000spend1200save2000 | You saved the money for 2 days. | Пари нужни за почивката: 2000Наличните пари: 1000spend – изваждаме от парите следващото число1200 - ние разполагаме с 1000, но се опитваме да похарчим 1200, тъй като не разполагаме с толкова, харчим наличните си 1000 и оставаме 0 лева.save – прибавяме към парите следващото число2000 – разполагаме с 0, добавяме 2000 и събираме парите успешно за 2 дни. |
| 11060spend10spend10spend10spend10spend10 | You can't save the money.5 | 250150spend50spend50save100save100 | You saved the money for 4 days. |

# Примерни изпитни задачи

## Монети

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

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** |
| 10102020202021 | No more cake left! You need 1 pieces more. |
| 102246STOP | 1. pieces are left.
 |

## 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** |
| **35**The number is: 35 |
| **105**Invalid number!**0**Invalid number!**-200**Invalid number!**77**The number is: 77 |

1. Създайте **нов проект** в съществуващото Visual Studio решение. В Solution Explorer кликнете с десен бутон на мишката върху **Solution** реда и изберете [Add] 🡪 [New Project…]:



1. Ще се отвори диалогов прозорец за избор на тип проект за създаване. Изберете **C#** **конзолно приложение** и задайте подходящо име, например “Number-In-Range”:



Вече имате solution с едно конзолно приложение в него. Остава да напишете кода за решаване на задачата.

1. Отидете в тялото на метода Main(string[] args) и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:



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





5. **Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/1014#0>.

## 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** |
| Pesho45.565.434.565.555665.435 | Pesho graduated. Average grade: 5.37 | Ani55.3265.43565.54.55565.566 | Ani graduated. Average grade: 5.45 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/1014#1>.

**Насоки**:

* Прочетете името от конзолата и направете брояч, с който да бройте класовете, която да е равна на нула. Направете още една променлива, която също да има стойност нула.
* Направете един while цикъл, който да се повтаря докато броячът е по-малък или равен на 12.
* В тялото на цикъла всеки път четете едно реално число от конзолата, след което проверявайте дали оценката е по-голяма или равна на 4. Ако е, прибавяйте я към сумата, която създадохме и увеличавайте стойността на брояча с 1.
* Извън тялото на цикъла направете една променлива, с която да изчислите средното аритметично от оценките и я принтирайте.



## 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** |
| Gosho55.565.435.565.555665.435 | Gosho graduated. Average grade: 5.53 | Mimi565656623 | Mimi has been excluded at 8 grade |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/1014#2>.

### Насоки:

* Използвайте решението от предната задача и добавете нужната функционалност

## Редица цели числа

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

### Примерен вход и изход:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| 1020304050END | Max number: 304Min number: 0 | 2505201001000END | Max number: 1000Min number: 0 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/1014#3>.

## 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** |
| 1000150020006500 | Goal reached! Good job! | 150030025003000Going home200 | 2500 more steps to reach goal. |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 1500300025015482000Going home2000 | Goal reached! Good job! | 12525040003026784682 | Goal reached! Good job! |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/1014#4>.

## 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 **"Your account balance was increased by: "** + 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 balance: "** + общата сума в сметката закръглена до втория знак след десетичната запетая.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| 35.5169.42100 | Your account balance was increased by: 5.51Your account balance was increased by: 69.42Your account balance was increased by: 100Total balance: 174.93 | 512045.55-150 | Your account balance was increased by: 120Your account balance was increased by: 45.55Invalid operation!Total balance: 165.55 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/1014#5>.

## 7. 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** |
| 500HardMediumEasyEasyMedium | The dispenser has been tapped 5 times. | 600HardMediumMediumEasyHard | 50ml has been spilled. |

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

## 8. 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 | 13 | 8 | 137 | 17 | 13715 | 31 | 1371531 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/1014#7>.

**Подсказки**:

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

## 9. Най-голям общ делител\*

Напишете програма, която чете две цели положителни числа a и b, въведени от потребителя, и изчислява и отпечатва **най-големият им общ делител (НОД)**.

### Примерен вход и изход:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2416 | 8 | 6718 | 1 | 159 | 3 | 10088 | 4 | 1010 | 10 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/1014#8>.

**Подсказка**: имплементирайте **алгоритъма на Евклид**: [https://bg.wikipedia.org/wiki/алгоритъм-на-Евклид](https://bg.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D1%8A%D0%BC_%D0%BD%D0%B0_%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4).

# Изпитни задачи от изминали издания на курса

## Преместване

*Изпитна задача от онлайн практически приемен изпит на* ***14 и 15 Април 2018****. Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/978/Programming-Basics-Online-Exam-14-and-15-April-2018)*.*

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.**

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.**

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** |
| 1010220202020122 | 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 |
| 101246**Done** | "10 Cubic meters left." | 10 \* 1 \* 2 = **20 кубични метра.**4 + 6 = **10** **кубични метра.**20 - 10 = **10 кубични метра.** |