# While - цикъл – допълнително упражнение

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

**Тествайте** решенията си в **judge системата**:

<https://alpha.judge.softuni.org/contests/while-loop-more-exercises/1684>

## Съдомиялна

Гошо работи в ресторант и отговаря за зареждането на съдомиялната накрая на деня.

Вашата задача е да напишете програма, която изчислява, **дали дадено закупено количество бутилки от препарат за съдомиялна е достатъчно, за да измие определено количество съдове.** Знае се, че всяка бутилка съдържа **750 мл. препарат**, **за 1 чиния са нужни 5 мл., а за тенджера 15 мл.**  Приемете, че **на всяко трето зареждане със съдове, съдомиялната се пълни само с тенджери, а останалите пъти с чинии.** Докато не получите команда "End" ще продължите да получавате бройка съдове, които трябва да бъдат измити.

### Вход

От конзолата се четат:

* **Брой бутилки от препарат, който ще бъде използван за миенето на чинии - цяло число в интервала [1…10]**

На всеки следващ ред, до получаване на командата "**End**" или докато **количеството препарат не се изчерпи**, **брой съдове, които трябва да бъдат измити - цяло число в интервала [1…100]**

### Изход

В случай, че количеството препарат **е било достатъчно** за измиването на съдовете:

**"Detergent was enough!"**

**"{брой чисти чинии} dishes and {брой чисти тенджери} pots were washed."**

**"Leftover detergent {количество останал препарат} ml."**

В случай, че количеството препарат **не е било достатъчно** за измиването на съдовете:

**"Not enough detergent, {количество не достигнал препарат} ml. more necessary!"**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 2  53  65  55  End | Detergent was enough!  118 dishes and 55 pots were washed.  Leftover detergent 85 ml. | Количество препарат = 2 \* 750 = **1500** мл.  53 **чинии** са заредени => 53 \* 5 = 265 мл. 1500 **—** 265 = 1235 мл. (**остатък**)  65 **чинии** => 65 \* 5 = 325 мл. 1235 **—** 325 = 910 мл. **(остатък)**  55 **тенджери** => 55 \* 15 = 825 мл. 910 **—** 825 = **85** мл. (**остатък**)  Получаваме команда "**End**", следователно **количеството е стигнало** и се печата съответното съобщение: Брой чинии = 53 + 65 = **118.** Брой тенджери = **55** |
| 1  10  15  10  12  13  30 | Not enough detergent, 100 ml. more necessary! | Количество препарат = 1 \* 750 = 750 мл.  10 **чинии** => 10 \* 5 = 50 мл. 750 **—** 50 => 700 **(остатък)**  15 **чинии =>** 15 \* 5 = 75 мл. 700 **—** 75 = 625 мл. **(остатък)**  10 **тенджери** => 10 \* 15 = 150 мл. 625 **—** 150 = 475 мл. **(остатък)**  12 **чинии** => 12 \* 5 = 60 мл. 475 **—** 60 = 415 мл. (**остатък)**  13 **чинии** => 13 \* 5 = 65 мл. 415 **—** 65 = 350 мл. **(остатък)**  30 **тенджери** => 30 \* 15 = 450 мл. **350 <= 450** **(100 недостиг)**,**следователно** **печатаме съобщение за недостиг на препарата** |

### JS - Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| (["2 ",  "53",  "65",  "55",  "End"]) | Detergent was enough!  118 dishes and 55 pots were washed.  Leftover detergent 85 ml. | Количество препарат = 2 \* 750 = **1500** мл.  53 **чинии** са заредени => 53 \* 5 = 265 мл. 1500 **—** 265 = 1235 мл. (**остатък**)  65 **чинии** => 65 \* 5 = 325 мл. 1235 **—** 325 = 910 мл. **(остатък)**  55 **тенджери** => 55 \* 15 = 825 мл. 910 **—** 825 = **85** мл. (**остатък**)  Получаваме команда "**End**", следователно **количеството е стигнало** и се печата съответното съобщение: Брой чинии = 53 + 65 = **118.** Брой тенджери = **55** |
| (["1",  "10",  "15",  "10",  "12",  "13",  "30"]) | Not enough detergent, 100 ml. more necessary! | Количество препарат = 1 \* 750 = 750 мл.  10 **чинии** => 10 \* 5 = 50 мл. 750 **—** 50 => 700 **(остатък)**  15 **чинии =>** 15 \* 5 = 75 мл. 700 **—** 75 = 625 мл. **(остатък)**  10 **тенджери** => 10 \* 15 = 150 мл. 625 **—** 150 = 475 мл. **(остатък)**  12 **чинии** => 12 \* 5 = 60 мл. 475 **—** 60 = 415 мл. (**остатък)**  13 **чинии** => 13 \* 5 = 65 мл. 415 **—** 65 = 350 мл. **(остатък)**  30 **тенджери** => 30 \* 15 = 450 мл. **350 <= 450** **(100 недостиг)**,**следователно** **печатаме съобщение за недостиг на препарата** |

## Система за отчет

На благотворително събитие плащанията за закупените продукти **винаги се редуват: плащане в брой и плащане с карта.** Установени са следните правила за заплащане:

* Ако продуктът **надвишава 100лв**., за него **не може да се плати в брой**
* Ако продуктът е на цена **под 10лв**., за него **не може да се плати с кредитна карта**

Програмата приключва **или след като получим команда "**End**" или след като средствата бъдат събрани.**

### Вход

От конзолата се четат:

* **Сумата, която се очаква да бъде събрана от продажбите -** **цяло** **число в интервала [1 ... 10000]**

На всеки следващ ред, до получаване на командата "End" или докато **не се съберат нужните средства**: **цените на предметите**, които ще бъдат закупени **-** **цяло** **число в интервала [1 ... 500]**

### Изход

На конзолата да се отпечата:

* При **успешна транзакция**: "**Product sold!**"
* При **неуспешна транзакция:** "**Error in transaction!**"
* Ако сумата **на всички закупени продукти надвиши или достигне очакваната сума**, програмата трябва **да приключи** и на конзолата **да се изпишат два реда**:
* **"Average CS:** {средно плащане в кеш на човек}**"**
* **"Average CC:** {средно плащане с карта на човек}**"**

**Плащанията трябва да бъдат форматирани до втората цифра след десетичния знак.**

* При получаване на **команда "**End**", да се изпише един ред:**
* **"Failed to collect required money for charity."**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 500  120  8  63  256  78  317 | Error in transaction!  Error in transaction!  Product sold!  Product sold!  Product sold!  Product sold!  Average CS: 70.50  Average CC: 286.50 | По условие се редуват **първо в брой плащане, след това чрез кредитна карта**  120 >= 100 **транзакцията** **се отхвърля** 8 <= 10 **транзакцията се отхвърля** 63 <= 100 => **транзакцията е успешна.** 256 >= 10 => **транзакцията е успешна.** 78 <= 100 => **транзакцията е успешна.** 317 >= 10 => **транзакцията е успешна.** Обща събрана сума => 63 + 256 + 78 + 317 = **714**;  **714** >= 500 => Обща сума в брой => 63 + 78 = 141Средно в брой => 141 / 2 = **70.50** Общо кредитни карти => 256 + 317 = 573Средно кредитни карти => 573 / 2 = **286.50** |
| 600  86  150  98  227  End | Product sold!  Product sold!  Product sold!  Product sold!  Failed to collect required money for charity. | 86 <= 100 => **транзакцията е успешна.** 150 >= 10 => **транзакцията е успешна.** 98 <= 100 => **транзакцията е успешна.** 227 >= 10 => **транзакцията е успешна.** Четем от конзолата команда "End". Проверка, дали е достатъчна сумата => 86 + 150 + 98 + 227 = 561; **561 <= 600 =>** печатаме, че сумата **не е била събрана**. |

### JS - Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| (["500",  "120",  "8",  "63",  "256",  "78",  "317"]) | Error in transaction!  Error in transaction!  Product sold!  Product sold!  Product sold!  Product sold!  Average CS: 70.50  Average CC: 286.50 | По условие се редуват **първо в брой плащане, след това чрез кредитна карта**  120 >= 100 **транзакцията** **се отхвърля** 8 <= 10 **транзакцията се отхвърля** 63 <= 100 => **транзакцията е успешна.** 256 >= 10 => **транзакцията е успешна.** 78 <= 100 => **транзакцията е успешна.** 317 >= 10 => **транзакцията е успешна.** Обща събрана сума => 63 + 256 + 78 + 317 = **714**;  **714** >= 500 => Обща сума в брой => 63 + 78 = 141Средно в брой => 141 / 2 = **70.50** Общо кредитни карти => 256 + 317 = 573Средно кредитни карти => 573 / 2 = **286.50** |
| (["600",  "86",  "150",  "98",  "227",  "End"]) | Product sold!  Product sold!  Product sold!  Product sold!  Failed to collect required money for charity. | 86 <= 100 => **транзакцията е успешна.** 150 >= 10 => **транзакцията е успешна.** 98 <= 100 => **транзакцията е успешна.** 227 >= 10 => **транзакцията е успешна.** Четем от конзолата команда "End". Проверка, дали е достатъчна сумата => 86 + 150 + 98 + 227 = 561; **561 <= 600 =>** печатаме, че сумата **не е била събрана**. |

## Поток от букви

Напишете програма, която прочита скрито съобщение в поредица от символи. Те се получават по един на ред до получаване на командата "**End**". Думите се образуват от буквите в **реда на четенето** им. Символите, които не са **латински букви трябва да бъдат игнорирани.** Думите скрити в потока са разделени от **тайна команда от три букви** – **"c", "o" и "n".** При **първото получаване** на една от тези букви, тя се маркира като срещната, **но не се запазва в думата**. При всяко **следващо нейно срещане се записва** нормално в думата. След като са налични **и трите символа от командата**, се печата думата и интервал " ". Започва се нова дума, която по същия начин чака тайната команда, за да бъде отпечатана.

### Вход

От конзолата се чете поредица от редове с един символ на всеки до получаване на командата **"End"**.

### Изход

На конзолата се печата на един ред **всяка дума след тайната команда,** следвана от интервал.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| H  n  e  l  l  o  o  c  t  c  h  o  e  r  e  n  e  End | Hello there | **"H", "n", "e", "l", "l", "o", "o", "c", "t", "c", "h", "o", "e", "r", "e", "n", "e"** са всички букви прочетени от конзолата. Първо се чете **"H"** и се прибавя към думата. След това се чете **"n"**. То е част от командата и **не се прибавя към думата тъй като се среща за първи път**. След това се четат **"e"**, **"l"**, **"l"** от конзолата и се добавят към думата. Прочита се **"o"** и се отбелязва като срещната, но не се добавя към думата. След това се прочита **"o"** отновои се записва. Следва **"c"** и вече са налични символите за тайната команда. **Отпечатва се** **"Hello "** и се следи за нова тайна команда. Прочита се **"t"** и се запазва в новата дума. След това **"c"** e част от новата команда и не се добавя. Прочита се **"h"** и се добавя към думата. Следва **"o"**,което e част от новата команда. Прочитат се **"e", "r", "e"** и се добавят към думата. Приема се **"n"** и тайната команда е завършена. Печата се **"there "**. Прочита се **"e"** и се добавя към нова дума. Получава се командата **"End"** и програмата приключва.  Крайният резултат е **"Hello there " .** |
| %  !  c  ^  B  `  o  %  o  o  M  )  {  n  \  A  D  End | BooM | **"%", "!", "c", "^", "B", "`", "o", "%", "o", "o", "M", ")", "{", "n", "\", "A", "D"** са прочетените символи. Пропускат се **"%"** и **"!"**,защото **не са част от латинската азбука**.Прочита се **"c"** и се запаметява за тайната команда, **без да се добавя към думата**. Пропуска се **"^"**. Следва **"B"** и се записва в думата. Пропуска се **"`"**. След това **"o"** е част от тайната команда и **не се добавя в думата**. Пропуска се **"%"**. Прочитат се **"o", "o", "M"** и се записват към думата една след друга. Пропускат се **")"** и **"{".** Следва **"n"** и тайната команда е завършена. Печата се **"BooM "**. Пропуска се **"\"**. След това се четат **"A", "D"** и се запаметяват в нова дума. Чете се командата **"End"** и програмата приключва.  Окончателният резултат е **"BooM "**. |
| o  S  %  o  l  ^  v  e  c  n  &  m  e  c  o  n  End | Solve me | **"o", "S", "%", "o", "l"^", "v", "e", "c", "n", "&", "m", "e", "c", "o"** и **"n"** са прочетените символи. Прочита се **"o"** и се запаметява за тайната команда, **без да се добавя към думата**. Следва **"S"** и се записва в думата. Пропуска се **"%"**. Прочитат се **"o"** и **"l"** и се добавят към думата. Пропуска се **"^"**. Прочитат се **"v"** и **"e"** и се добавят. Прочитат се **"c"** и **"n"** и командата се завършва. Отпечатва се **"Solve "**. Пропуска се **"&"**. Прочитат се **"m"** и **"e"** и се записват в новата дума. Прочитат се **"c", "o"** и **"n"** и командата е изпълнена. Отпечатва се **"me "**. Прочита се **"End"** и програмата приключва.  Крайния резултат е **"Solve me "**. |

### JS - Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| (["H ",  "n",  "e",  "l",  "l",  "o",  "o",  "c",  "t",  "c",  "h",  "o",  "e",  "r",  "e",  "n",  "e",  "End"]) | Hello there | **"H", "n", "e", "l", "l", "o", "o", "c", "t", "c", "h", "o", "e", "r", "e", "n", "e"** са всички букви прочетени от конзолата. Първо се чете **"H"** и се прибавя към думата. След това се чете **"n"**. То е част от командата и **не се прибавя към думата тъй като се среща за първи път**. След това се четат **"e"**, **"l"**, **"l"** от конзолата и се добавят към думата. Прочита се **"o"** и се отбелязва като срещната, но не се добавя към думата. След това се прочита **"o"** отновои се записва. Следва **"c"** и вече са налични символите за тайната команда. **Отпечатва се** **"Hello "** и се следи за нова тайна команда. Прочита се **"t"** и се запазва в новата дума. След това **"c"** e част от новата команда и не се добавя. Прочита се **"h"** и се добавя към думата. Следва **"o"**,което e част от новата команда. Прочитат се **"e", "r", "e"** и се добавят към думата. Приема се **"n"** и тайната команда е завършена. Печата се **"there "**. Прочита се **"e"** и се добавя към нова дума. Получава се командата **"End"** и програмата приключва.  Крайният резултат е **"Hello there " .** |
| (["%",  "!",  "c",  "^",  "B",  "`",  "o",  "%",  "o",  "o",  "M",  ")",  "{",  "n",  "\",  "A",  "D",  "End"]) | BooM | **"%", "!", "c", "^", "B", "`", "o", "%", "o", "o", "M", ")", "{", "n", "\", "A", "D"** са прочетените символи. Пропускат се **"%"** и **"!"**,защото **не са част от латинската азбука**.Прочита се **"c"** и се запаметява за тайната команда, **без да се добавя към думата**. Пропуска се **"^"**. Следва **"B"** и се записва в думата. Пропуска се **"`"**. След това **"o"** е част от тайната команда и **не се добавя в думата**. Пропуска се **"%"**. Прочитат се **"o", "o", "M"** и се записват към думата една след друга. Пропускат се **")"** и **"{".** Следва **"n"** и тайната команда е завършена. Печата се **"BooM "**. Пропуска се **"\"**. След това се четат **"A", "D"** и се запаметяват в нова дума. Чете се командата **"End"** и програмата приключва.  Окончателният резултат е **"BooM "**. |
| (["o",  "S",  "%",  "o",  "l",  "^",  "v",  "e",  "c",  "n",  "&",  "m",  "e",  "c",  "o",  "n",  "End"]) | Solve me | **"o", "S", "%", "o", "l"^", "v", "e", "c", "n", "&", "m", "e", "c", "o"** и **"n"** са прочетените символи. Прочита се **"o"** и се запаметява за тайната команда, **без да се добавя към думата**. Следва **"S"** и се записва в думата. Пропуска се **"%"**. Прочитат се **"o"** и **"l"** и се добавят към думата. Пропуска се **"^"**. Прочитат се **"v"** и **"e"** и се добавят. Прочитат се **"c"** и **"n"** и командата се завършва. Отпечатва се **"Solve "**. Пропуска се **"&"**. Прочитат се **"m"** и **"e"** и се записват в новата дума. Прочитат се **"c", "o"** и **"n"** и командата е изпълнена. Отпечатва се **"me "**. Прочита се **"End"** и програмата приключва.  Крайния резултат е **"Solve me "**. |

# Задачи за шампиони

## Числа от 1 до 100, които се делят на 3 без остатък

Напишете програма, която печата всички числа в интервала от 1 до 100, който се делят на 3 без остатък, по едно на ред.

|  |  |
| --- | --- |
| **вход** | **изход** |
| (няма) | 3  6  9  …  99 |

**Насоки:** потърсете информация за **for** цикъл.

## Средно аритметично

Напишете програма, която прочита едно число **n**, след това прочита **n** на брой **цели числа** и принтира средно аритметичното на тяхната сума число, форматирано до втората цифра след десетични знак.

### JS - Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 4  3  2  4  2 | 2.75 | 2  6  4 | 5.00 | 3  82  43  22 | 49.00 | 4  95  23  76  23 | 54.25 |

### JS - Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| (["4",  "3",  "2",  "4",  "2"]) | 2.75 | (["2",  "6",  "4"]) | 5.00 | (["3",  "82",  "43",  "22"]) | 49.00 | (["4",  "95",  "23",  "76",  "23"]) | 54.25 |