# Допълнителни упражнения: Повторения

Практически упражнения към курса ["Увод в програмирането" за ученици](https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Basics).

Тествайте решенията си от тази тема в Judge:

<https://judge.softuni.bg/Contests/2691/Допълнителни-задачи-за-повторения>

## 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** |

## 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**. |

## 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 "**. |

## Завръщане в миналото

Иванчо е на **18 години** и получава наследство, което се състои от **X сума пари** и **машина на времето**. Той решава **да се върне до 1800 година**, но не знае **дали парите** ще **са достатъчни**, за да живее без да работи. Напишете **програма, която пресмята**, дали Иванчо **ще има достатъчно пари,** за да не се налага да работи **до дадена година включително**. Като приемем, че **за всяка четна** (1800, 1802 и т.н.) **година** ще **харчи 12 000 лева**. За **всяка нечетна** (1801,1803 и т.н.) ще **харчи 12 000 + 50 \* [годините, които е навършил през дадената година]**.

### Вход

Входът се чете от конзолата и **съдържа точно 2 реда**:

* **Наследените пари** – **реално число в интервала [1.00 ... 1 000 000.00]**
* **Годината,** **до която трябва да живее (включително) – цяло число в интервала [1801 ... 1900]**

### Изход

Да се **отпечата** на конзолата **1 ред. Сумата** трябва да е **форматирана** до **два знака след десетичната запетая**:

* Ако **парите са** **достатъчно**:
  + "Yes! He will live a carefree life and will have {N} dollars left." – където **N** са парите, които ще му останат.
* Ако **парите НЕ са достатъчно**:
  + "**He will need {М} dollars to survive."** – където **M е сумата**, която **НЕ достига**.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 50000  1802 | Yes! He will live a carefree life and will have 13050.00 dollars left. | 1800 → **четна**  → **Харчи** **12000** лева  → **Остават** 50000 – 12000 = **38000**  1801 → **нечетна**  → **Харчи** 12000 + **19\*50** = 12000 + 950 = **12950**  → **Остават** 38000 – 12950 = **25050**  1802 → **четна**  → **Харчи** 12000 лева  → **Остават** 25050 – 12000 = **13050** |
| **Вход** | **Изход** | **Обяснения** |
| 100000.15  1808 | He will need 12399.85 dollars to survive. | 1800 → **четна**  → **Остават** 100000.15 – 12000 = **88000.15**  1801 → **нечетна**  → **Остават** 100000.15 – 12950 = **87050.15**  ...  1808 → **четна** → -399.85 – 12000 = -12399.85  **12399.85 не достигат** |

## Болница

**За даден период от време,** **всеки ден** в болницата пристигат пациенти за преглед. Тя разполага **първоначално** със **7 лекари**.Всеки лекар може да преглежда **само по един пациент на ден**, но понякога има недостиг на лекари, затова **останалите пациенти се изпращат в други болници**. **Всеки трети ден**, болницата прави изчисления и **ако броят на непрегледаните пациенти е по-голям от броя на прегледаните**, **се назначава още един лекар**. Като **назначаването става преди да започне приемът на пациенти за деня.**

Напишете програма, която изчислява **за дадения период броя на прегледаните и непрегледаните пациенти**.

### Вход

Входът се чете от **конзолата** и съдържа:

* На **първия ред** – **периода**, за който трябва да направите изчисления. **Цяло число** в интервала **[1 ... 1000]**
* На следващите **редове(равни на броят на дните)** – **броя пациенти**, които пристигат за преглед за **текущия ден.** **Цяло число** в интервала [**0**…**10 000**]

### Изход

Да се **отпечатат** на конзолата **2 реда** :

* На **първия ред**: "Treated patients: {брой прегледани пациенти}."
* На **втория ред**: "Untreated patients: {брой непрегледани пациенти}."

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| 4  7  27  9  1 | Treated patients: 23.  Untreated patients: 21. | **1 ден:** 7 прегледани и 0 непрегледани пациента за деня  **2 ден:** 7 прегледани и 20 непрегледани пациента за деня  **3 ден:** До момента прегледаните пациенти са общо 14, а непрегледаните – 20 –> Назначава се нов лекар –>  8 прегледани и 1 непрегледан пациент за деня  **4 ден:** 1 прегледан и 0 непрегледани пациента за деня  **Общо**: **23 прегледани** и **21 непрегледани** **пациенти**. | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 6  25  25  25  25  25  2 | Treated patients: 40.  Untreated patients: 87. | 3  7  7  7 | Treated patients: 21.  Untreated patients: 0. |

## Логистика

Отговаряте за логистиката на различни товари. **В зависимост от теглото** на товара е **нужно различно превозно средство**. **Цената на тон**, за която се превозва товара **е различна за всяко превозно средство**:

* До **3 тона – микробус (200 лева на тон)**
* От **4 до 11 тона – камион (175 лева на тон)**
* **12 и повече тона – влак (120 лева на тон)**

Вашата задача е да изчислите **средната цена на тон превозен товар**, както и **процента на тоновете превозвани с всяко превозно средство, спрямо общото тегло(в тонове) на всички товари.**

### Вход

От конзолата се четат **поредица от числа, всяко на отделен ред**:

* На **първия ред** – **броя на товарите** за превоз – **цяло число в интервала** **[1...1000]**
* За **всеки един товар** на отделен ред – **тонажа на товара** – **цяло число в интервала [1...1000]**

### Изход

Да се отпечатат на конзолата **4 реда**, както следва:

* **Първи ред** – **средната цена на тон превозен товар (закръглена до втория знак** **след дес. запетая)**;
* **Втори ред** – **процентът** тона превозвани с **микробус (процент между** 0.00% **и** 100.00%**)**;
* **Трети ред** – **процентът**  тона превозвани с **камион (процент между** 0.00% **и** 100.00%**)**;
* **Четвърти ред** – **процентът** тона превозвани с **влак (процент между** 0.00% **и** 100.00%**)**.

### Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | **Вход** | **Изход** |
| 4  1  5  16  3 | 143.80  16.00%  20.00%  64.00% | Всички товари 1 + 5 + 16 + 3 = 25 тона  (4 \* 200 + 5 \* 175 + 16 \* 120)/25 = **143.80** (средно на тон)  **С микробус** – 4/25\*100 = **16.00%**  **С камион** – 5/25\*100 = **20.00%**  **С влак** – 16/25\*100 = **64.00%** | 5  2  10  20  1  7 | 149.38  7.50%  42.50%  50.00% |

## Оценки

Напишете програма, която да **пресмята статистика на оценки** от изпит. В началото програмата получава **броят на студентите** явили се на изпита и за **всеки студент неговата оценка**. На края програмата трябва да **изпечата процента на студенти** с оценка между **2.00 и 2.99**, между **3.00 и 3.99**, между **4.00 и 4.99**, **5.00 или повече**. Също така и **средният успех** на изпита.

### Вход

От конзолата се четат **поредица от числа, всяко на отделен ред**:

* На **първия ред** – **броя на студентите явили се на изпит** – **цяло число в интервала [1...1000]**
* За **всеки един студент** на отделен ред – **оценката от изпита** – **реално число в интервала [2.00...6.00]**

### Изход

Да се отпечатат на конзолата **5 реда**, които съдържат следната информация:

1. "Top students: {процент студенти с успех 5.00 или повече}%"
2. "Between 4.00 and 4.99: {между 4.00 и 4.99 включително}%"
3. "Between 3.00 and 3.99: {между 3.00 и 3.99 включително}%"
4. "Fail: {по-малко от 3.00}%"
5. "Average: {среден успех}"

Всички числа трябва да са **форматирани до вторият знак** след десетичната запетая.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 10  3.00  2.99  5.68  3.01  4  4  6.00  4.50  2.44  5 | Top students: 30.00%  Between 4.00 and 4.99: 30.00%  Between 3.00 and 3.99: 20.00%  Fail: 20.00%  Average: 4.06 | 5 и повече – трима = 30% от 10  Между 4 и 4.99 – трима = 30% от 10  Между 3 и 3.99 – двама = 20% от 10  Под 3 – двама = 20% от 10  Средният успех: 3 + 2.99 + 5.68 + 3.01 + 4 + 4 + 6 + 4.50 + 2.44 + 5 = 40.62 / 10 = 4.062 |
| **Вход** | **Изход** | |
| 6  2  3  4  5  6  2.2 | Top students: 33.33%  Between 4.00 and 4.99: 16.67%  Between 3.00 and 3.99: 16.67%  Fail: 33.33%  Average: 3.70 | |

## Игра на интервали

Напишете програма, която да **пресмята резултата от игра**. Първо получавате **число**, което **показва колко хода ще продължи играта**. После **за всеки ход на играта** ще получавате **по едно ново число**. **Според интервала** в който попада числото **се прибавят точки**. Ако числото е **отрицателно или по-голямо 50**, тогава то **е невалидно**. **В началото на играта резултата е 0** и **на всеки ход се прибавят точки** по следният начин:

* **От 0 до 9** 🡪 **20 % от числото**
* **От 10 до 19** 🡪 **30 % от числото**
* **От 20 до 29** 🡪 **40 % от числото**
* **От 30 до 39** 🡪 **50 точки**
* **От 40 до 50** 🡪 **100 точки**
* **Невалидно** **число** 🡪 **резултата се дели на 2**

Освен резултата програмата **трябва да изкарва статистика за проценти числа в дадените интервали**.

### Вход

Входът се чете от конзолата:

* **Първи** **ред** - колко хода ще има по време на играта – **цяло число в** **интервала [1...100]**
* **За всеки ход** – числата, които се проверяват в кой интервал са – **цели числа в интервала [-100...100]**

### Изход

Да се отпечата на конзолата **7 реда:**

* **1ви ред:** "{Краен резултат}"
* 2ри ред: "From 0 to 9: {процент в интервала}%"
* 3ти ред: "From 10 to 19: {процент в интервала}%"
* 4ти ред: "From 20 to 29: {процент в интервала}%"
* 5ти ред: "From 30 to 39: {процент в интервала}%"
* 6ти ред: "From 40 to 50: {процент в интервала}%"
* 7ми ред: "Invalid numbers: {процент в интервала}%"

**Всички числа** трябва да са **форматирана до вторият знак след запетаята.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 10  43  57  -12  23  12  0  50  40  30  20 | 295.80  From 0 to 9: 10.00%  From 10 to 19: 10.00%  From 20 to 29: 20.00%  From 30 to 39: 10.00%  From 40 to 50: 30.00%  Invalid numbers: 20.00% | **10 хода**; **Начален резултат** = **0 точки**  **1ви ход**: 40 <= 43 <= 50 🡪 **към резултата се добавят 100 точки**  **2ри ход**: 57 > 50 🡪**невалидно число** 🡪 **100/2 = 50 точки**  **3ти ход**:-12 < 0 🡪 **невалидно число** 🡪 **50/2 = 25 точки**  **4ти ход**: 20 <= 23 <= 29 🡪 **към резултата се добавят 40% от 23** = 25 + 9.2 = 34.2 и така до 10тият ход......  **От 0 до 9**: 1 число (0) = **10%**  **От 10 до 19**: 1 число (10) = **10%**  **От 20 до 29**: 2 числа (20 и 23) = **20%**  **От 30 до 39**: 1 число (30) = **10%**  **От 40 до 50**: 3 числа (40, 43 и 50) = **30%**  **Невалидни**: 2 числа (57 и -12) = **20%** |

## Месечни разходи

Напишете програма която да **пресмята средният разход за месец** на семейство **за даден период време**. За **всеки месец разходите са следните**:

* За **ток** – **всеки месец е различен**, ще се чете от конзолата
* за **вода** – **20 лв.**
* за **интернет** – **15 лв.**
* за **други** – **събират се тока, водата и интернета** **за месеца** и **към сумата се прибавят 20%.**

**За всеки разход** трябва да се пресметне **колко общо е платено за всички месеци**.

### Вход

Входът се чете от конзолата:

* **Първи** **ред** – **месеците** за които се търси средният разход – **цяло число** в интервала **[1...100]**
* **За всеки месец** – **сметката за ток** – **реално число** в интервала **[1.00...1000.00]**

### Изход

Да се отпечата на конзолата **5 реда:**

* 1ви ред:"Electricity: {ток за всички месеци} lv"
* 2ри ред: "Water: {вода за всички месеци} lv"
* 3ти ред: "Internet: {интернет за всички месеци} lv"
* 4ти ред: "Other: {други за всички месеци} lv"
* 5ти ред: "Average: {средно всички разходи за месец} lv"

**Всички числа** трябва да са **форматирана до вторият знак след запетаята.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 5  68.63  89.25  132.53  93.53  63.22 | Electricity: 447.16 lv  Water: 100.00 lv  Internet: 75.00 lv  Other: 746.59 lv  Average: 273.75 lv | **За 5 месеца**:  **Ток** 🡪 68.63 + 89.25 + 132.53 + 93.53 + 63.22 = **447.16 лв.**  **Вода** 🡪 5 месеца по 20 лв. = **100 лв.**  **Интернет** 🡪 5 месеца по15 = **75 лв.**  **Други** 🡪 *(68.63+20+15) + 20% = 124.356* 🡺 **общо 746.592 лв.**  *(89.25+20+15) + 20% = 149.1*  *(132.53+20+15) + 20% = 201.036*  *(93.53+20+15) + 20% = 154.236*  *(63.22+20+15) + 20% = 117.864*  **Средно на месец** 🡺 *(447.16+100+75+746.592)/5* = **273.7504 лв.** |
| **Вход** | **Изход** | |
| 8  123.54  231.54  140.23  100  122.4  430  178.52  64.2 | Electricity: 1390.43 lv  Water: 160.00 lv  Internet: 120.00 lv  Other: 2004.52 lv  Average: 459.37 lv | |

## Футболен турнир

Екипът на СофтУни си организира футболен турнир. Първоначално прочитаме от **конзолата капацитета на стадиона и броят на всички фенове**. След това **за всеки фен се чете секторът, в който се намира**. Феновете на **първия отбор** са в **сектор А и Б**, а на **втория – В и Г**. Да се напише програма, която **изчислява процентите на феновете във всеки сектор, спрямо общия брой фенове на стадиона, както и общият процент на феновете за двата отбора, спрямо капацитета на стадиона. Общият брой на феновете НЕ надвишава капацитета на стадиона.**

### Вход

От конзолата се четат **поредица от числа**, всяко **на отделен ред**:

1. **Капацитетът на стадиона** – **цяло число в интервала [1 … 10000]**;
2. **Броят на всички фенове** – **цяло число в интервала [1 … 10000]**.

**За всеки един фен**, **на отделен ред** се прочита:

1. **Секторът, на който се намира** – **текст** – **"A", "B", "V" и "G"**.

### Изход

Да се отпечатат на конзолата **5 реда**, всеки от които съдържа **процент между 0.00% и 100.00%**, **форматирани до втората цифра след десетичната запетая**:

1. **Процентът на феновете в сектор А**
2. **Процентът на феновете в сектор Б**
3. **Процентът на феновете в сектор В**
4. **Процентът на феновете в сектор Г**
5. **Процентът на всички фенове, спрямо капацитета на стадиона**.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| 76  10  A  V  V  V  G  B  A  V  B  B | 20.00%  30.00%  40.00%  10.00%  13.16% | Феновете в сектор А са **2**. Изчисляваме какъв процент са от **общия брой фенове – 10** на стадиона.  2 / 10 \* 100 = **20.00%**  Феновете в сектор B са **3**. По същия начин изчисляваме **техния процент** спрямо общия брой на феновете.  3 / 10 \* 100 = **30.00%**  Феновете в сектор V са **4**. По същия начин изчисляваме **техния процент** спрямо общия брой на феновете.  4 / 10 \* 100 = **40.00%**  В сектор G има само **1** фен. По същия начин изчисляваме **какъв процент е спрямо общия брой на феновете.**  1/10 \* 100 = **10.00%**  Изчисляваме процентът на всички фенове, спрямо капацитетът на стадиона:  10/ 76 \* 100 = **13.16%** | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 93  16  A  V  G  G  B  B  G  B  A  B  B  B  A  B  B  A | 25.00%  50.00%  6.25%  18.75%  17.20% | 1000  12  A  A  V  V  A  G  A  A  V  G  V  A | 50.00%  0.00%  33.33%  16.67%  1.20% |

## Уникални PIN кодове

Да се напише програма, която генерира трицифрени PIN кодове, като цифрите на всеки PIN код са в определен интервал. За да бъде валиден един PIN код той трябва да отговаря на следните условия:

* **Първата** и **третата** цифра трябва да бъдат **четни**.
* **Втората** цифра трябва да бъде **просто число** в диапазона **[2...7]**.

### Вход

От конзолата се четат **3** реда:

* **Горната граница на първото число** - цялочисло в диапазона **[1...9]**
* **Горната граница на второто число** - цяло число в диапазона **[1...9]**
* **Горната граница на третото число** - цяло число в диапазона **[1...9]**

### Изход

Да се отпечатат на конзолата всички валидни **трицифрени PIN кодове**, чиито цифри **отговарят** на съответните интервали.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3  5  5 | 2 2 2  2 2 4  2 3 2  2 3 4  2 5 2  2 5 4 | Първото въведено число е **3**, отговарящо за максималната стойност на **първата цифра**.  Второто въведено число е **5**, отговарящо за максималната стойност на **втората цифра**.  Третото въведено число е **5**, отговарящо за максималната стойността на **третата цифра**.  Във всички трицифрени PIN кодове, които сме получили **първата цифра** ни е **2**, защото това е **единственото възможно** **четно** **число**.  При **втората цифра** важи **друго** правило. Там трябва да подберем **всички възможни прости числа** в диапазона **от 2 до 7**. В нашия случай тези числа са както следва 2, 2, 3, 3, 5, 5.  При **третата цифра** важи правилото за **четните** **числа** и ако го спазваме, получаваме, че възможните числа са: **2**, **4**, **2**, **4**, **2**, **4.** |
| 8  2  8 | 2 2 2  2 2 4  2 2 6  2 2 8  4 2 2  4 2 4  4 2 6  4 2 8  6 2 2  6 2 4  6 2 6  6 2 8  8 2 2  8 2 4  8 2 6  8 2 8 |  |

## Комбинации от букви

Напишете програма, която да **принтира** на конзолата **всички комбинации от 3 букви** в **зададен интервал**, **като се пропускат** комбинациите **съдържащи зададена от конзолата буква**. **Накрая трябва да се изпринтира броят** на **отпечатаните комбинации**.

### Вход

Входът се чете от **конзолата** и съдържа **точно 3 реда:**

1. **Малка буква от английската азбука за начало на интервала – от ‘a’ до ‚z’.**
2. **Малка буква от английската азбука за край на интервала – от първата буква до ‚z’.**
3. **Малка буква от английската азбука – от ‘a’ до ‚z’ – като комбинациите съдържащи тази буквата се пропускат.**

### Изход

Да се отпечатат на **един ред** **всички комбинации отговарящи на условието** **плюс броят им** **разделени с интервал.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| a  c  b | aaa aac aca acc caa cac cca ccc 8 | Всички възможни комбинации с буквите a, b, и c са:  aaa aab aac aba abb abc aca acb acc baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb cbc cca ccb ccc  Комбинациите **съдържащи b** **не са валидни**.  Остават 8 валидни комбинации |
| **Вход** | **Изход** | |
| f  k  h | fff ffg ffi ffj ffk fgf fgg fgi fgj fgk fif fig fii fij fik fjf fjg fji fjj fjk fkf fkg fki fkj fkk gff gfg gfi gfj gfk ggf ggg ggi ggj ggk gif gig gii gij gik gjf gjg gji gjj gjk gkf gkg gki gkj gkk iff ifg ifi ifj ifk igf igg igi igj igk iif iig iii iij iik ijf ijg iji ijj ijk ikf ikg iki ikj ikk jff jfg jfi jfj jfk jgf jgg jgi jgj jgk jif jig jii jij jik jjf jjg jji jjj jjk jkf jkg jki jkj jkk kff kfg kfi kfj kfk kgf kgg kgi kgj kgk kif kig kii kij kik kjf kjg kji kjj kjk kkf kkg kki kkj kkk 125 | |
| **Вход** | **Изход** | |
| a  c  z | aaa aab aac aba abb abc aca acb acc baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb cbc cca ccb ccc 27 | |

## Щастливи числа

Да се напише програма, която **прочита едно цяло число** N и генерира всички възможни **"щастливи" и различни 4-цифрени числа(всяка цифра от числото е в интервала [1...9])**.

Числото трябва да отговаря на **следните условия**:

Щастливо число е 4-цифрено число, на което **сбора от първите две цифри** **е равен на сбора от последните две**. Числото N трябва да се дели **без остатък** от **сбора на първите две цифри** на "**щастливото**" **число**.

### Вход

Входът се чете от конзолата и се състои от **едно** **цяло число** в интервала [**2**...**10000**]

### Изход

На конзолата трябва да се отпечатат **всички "щастливи" и различни 4-цифрени числа**, разделени с **интервал**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3 | 1212 1221 2112 2121 | И четирите числа са "щастливи"  3 / (1+2) = 1 – остатък 0 |
| 7 | 1616 1625 1634 1643 1652 1661 2516 2525 2534 2543 2552 2561 3416 3425 3434 3443 3452 3461 4316 4325 4334 4343 4352 4361 5216 5225 5234 5243 5252 5261 6116 6125 6134 6143 6152 6161 | Всички числа са "щастливи"  7 / (1+6) = 1 – остатък 0  7 / (2+5) = 1 – остатък 0  7 / (3+4) = 1 – остатък 0 |
| 24 | 1111 1212 1221 1313 1322 1331 1515 1524 1533 1542 1551 1717 1726 1735 1744 1753 1762 1771 2112 2121 2213 2222 2231 2415 2424 2433 2442 2451 2617 2626 2635 2644 2653 2662 2671 3113 3122 3131 3315 3324 3333 3342 3351 3517 3526 3535 3544 3553 3562 3571 3939 3948 3957 3966 3975 3984 3993 4215 4224 4233 4242 4251 4417 4426 4435 4444 4453 4462 4471 4839 4848 4857 4866 4875 4884 4893 5115 5124 5133 5142 5151 5317 5326 5335 5344 5353 5362 5371 5739 5748 5757 5766 5775 5784 5793 6217 6226 6235 6244 6253 6262 6271 6639 6648 6657 6666 6675 6684 6693 7117 7126 7135 7144 7153 7162 7171 7539 7548 7557 7566 7575 7584 7593 8439 8448 8457 8466 8475 8484 8493 9339 9348 9357 9366 9375 9384 9393 |  |

## Номер

Поздравления, поради вашите задълбочени знания в сферата на програмирането МВР реши да наеме точно вас за създаването на новата им система за **генериране на специални автомобилни номера**. Всеки един специален автомобилен номер се състой от **четири числа. Условията, които разграничават специалните от обикновените номера са следните:**

* **Ако номерът започва с четна цифра, то той трябва да завършва на нечетна цифра и обратното – ако започва с нечетна, завършва на четна**
* **Първата цифра от номера е по-голяма от последната**
* **Сумата от втората и третата цифра трябва да е четно число**

Входа се състой от две числа - начало и край на интервал, между които трябва да се генерира всяко едно число от номера.

### Вход

1. **Първи ред** - едноцифрено число - **началото** на интервала – **цяло число в интервала [1…9]**
2. **Втори ред** - едноцифрено число - **края** на интервала – **цяло число в интервала [1…9]**

### Изход

На конзолата трябва да се отпечатат **всички специални номера**, **разделени с** **интервал**.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 2  3 | 3222 3332 | Всяка цифра от номера е в интервала **[2…3]**. За всеки номер проверяваме дали изпълнява съответните условия и ако ги изпълнява го печатаме на конзолата. В противен случай го игнорираме.  Понеже 3 е нечетно, а 2 е четно, 3 >2 и резултата от 2+2 е четно число, то 3222 отговаря на условията и го принтираме.  По същия начин при 3332 - 3 е нечетно, а 2 е четно , 3 >2 и 3+3 = 6 , което е четно число. |
| 3  5 | 4333 4353 4443 4533 4553 5334 5354 5444 5534 5554 | |
| 5  8 | 6555 6575 6665 6685 6755 6775 6865 6885 7556 7576 7666 7686 7756 7776 7866 7886 8555 8557 8575 8577 8665 8667 8685 8687 8755 8757 8775 8777 8865 8867 8885 8887 | |

## Предизвикай сватбата

Провокирани от сватбата си, Михаела и Иван решават да предоставят нова услуга на клиенти на ресторанта си, а именно вечеря за запознанства - "Предизвикай Сватбата". **Напишете** **програма**, която **отпечатва всички възможни срещи на клиентите на ресторанта**. При настаняване всеки мъж и всяка жена получават талончета с поредни номера **стартирайки от 1.** Ако бъдат **заети всички маси, програмата трябва да приключи**. Всяка маса има две места.

### Вход

От конзолата се **четат точно 3 числа**, всяко на **отделен ред**:

* **Броя** клиенти **мъже** - **цяло** **число в интервала [1...100]**
* **Броя** клиенти **жени** - **цяло** **число в интервала [1...100]**
* **Максималният брой маси - цяло число в интервала [1...100]**

### Изход

На конзолата се **принтират на един ред, разделени с интервал** всички срещи в следният формат:

* ({№ клиент} <-> {№ клиент}) ({№ клиент} <-> {№ клиент}) ...

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 2  2  6 | (1 <-> 1) (1 <-> 2) (2 <-> 1) (2 <-> 2) | Мъж 1 се среща с Жена 1, Мъж 1 се среща с Жена 2, Мъж 2 се среща с Жена 1, Мъж 2 се среща с Жена 2. |
| 2  2  3 | (1 <-> 1) (1 <-> 2) (2 <-> 1) | Мъж 1 се среща с Жена 1, Мъж 1 се среща с Жена 2, Мъж 2 се среща с Жена 1. И трите маси за заети и програмата приключва. |
| 5  8  40 | (1 <-> 1) (1 <-> 2) (1 <-> 3) (1 <-> 4) (1 <-> 5) (1 <-> 6) (1 <-> 7) (1 <-> 8) (2 <-> 1) (2 <-> 2) (2 <-> 3) (2 <-> 4) (2 <-> 5) (2 <-> 6) (2 <-> 7) (2 <-> 8) (3 <-> 1) (3 <-> 2) (3 <-> 3) (3 <-> 4) (3 <-> 5) (3 <-> 6) (3 <-> 7) (3 <-> 8) (4 <-> 1) (4 <-> 2) (4 <-> 3) (4 <-> 4) (4 <-> 5) (4 <-> 6) (4 <-> 7) (4 <-> 8) (5 <-> 1) (5 <-> 2) (5 <-> 3) (5 <-> 4) (5 <-> 5) (5 <-> 6) (5 <-> 7) (5 <-> 8) | |

## Четни двойки

Напишете програма, която генерира и принтира на конзолата **четирицифрени** числа, в които **първата** и **втората** двойка цифри образуват **двуцифрени прости числа** (пример за такова число **1723**). Крайната стойност до която трябва да се генерират двойките се определя от други 2 цифри, подадени като вход, които определят **с колко крайната стойност е по-голяма от началната**.

### Вход

От конзолата се четат **четири** реда:

* **На първия ред** – началната стойност на първите **първата** **двойка** числа – **цяло** положително число в диапазона **[10… 90]**
* **На втория ред** – началната стойност на **втората** **двойка** числа – **цяло** положително число в диапазона **[10… 90]**
* **На третия ред** – разликата между началната и крайната стойност на **първата** двойка числа – **цяло** положително число в диапазона **[1… 9]**
* **На четвъртия ред** – разликата между началната и крайната стойност на **втората** двойка числа – **цяло** положително число в диапазона **[1… 9]**

### Изход:

Да се отпечатат на конзолата четирицифрените числа, в които **първите** **две** и **вторите** **две** **цифри** са **прости** двуцифрени числа.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 10  20  5  5 | 1123  1323 | Началната стойност на първата двойка цифри е 10, а на втората 20. Крайните стойности са съответно: **10 + 5 = 15**  **20 + 5 = 25**  Получават се следните комбинации: **1020 1021 1022 1023 1024 1025 1120 1121 1122 1123 1124 1125 ...** **1320 1321 1322** **1323 1324 1325 1420 1421 1422 1423 1424 1425 1520 1521 1522 1523 1524 1525**  но от тях само **1123** и **1323** са четирицифрени числа, в които първата част и втората са **едновременно** **прости** **числа**. |
| 10  30  9  6 | 1131  1331  1731  1931 | Началната стойност на първата двойка цифри е 10, а на втората 30. Крайните стойности са съответно: **10 + 9 = 19**  **30 + 6 = 36**  Получават се следните комбинации: **1030 1031 1032 1033 1034 1035 1036 1130 1131 1131 1132 ...** **1330** **1331 1332 1333 1334 ... 1930 1931 1932 1933 1934 1935 1936**  но от тях само **1123** **1331 1731 1931** са четирицифрени числа, в които първата част и втората са **едновременно** **прости** **числа**. |

## Министерство на образованието и науката (МОН)

* Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".



* Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен** **лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

