# Допълнитено упражнение: Вложени цикли

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

Тествайте решенията си в judge системата: <judge.softuni.bg/Contests/1382>

## Отключване на тайната врата

В града има тайна врата, за която всички знаят, но никой не е успявал да я отключи и да види какво има зад нея. За да бъде отключена трябва да се въведе трицифрен код.

Напишете програма, която генерира комбинации спрямо въведени числа – предположения от потребителя. От конзолата се въвеждат **три цифри**. Тези цифри ще бъдат **горната граница**, до която ние искаме да получим всички **трицифрени** числа, на които всяка една цифра отговаря на следните условия:

* Цифрата на **единиците** и цифрата на **стотиците** трябва да бъде четна
* Цифрата на **десетиците** да бъде **просто число** в диапазона (**2**...**7**).

Това ще са възможните комбинации според въведените предположения от потребителя, с които ще може евентуално да се отключи вратата.

### Вход

От конзолата се четът **3** реда:

* **Горната граница на стотиците** - цялочисло в диапазона (1-9)
* **Горната граница на десетиците** - цяло число в диапазона (1-9)
* **Горната граница на едениците** - цяло число в диапазона (1-9)

### Изход

Да се отпечатат на конзолата всички **трицифрени** **числа**, за които всяка една част отговаря на **условията** по-горе.

### 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 **стотиците**.  Второто въведено число е **5**, отговарящо за максималната стойност на **десетиците**.  Третото въведено число е **5**, отговарящо за максималната стойността на **едениците**.  Във всички кобинации, които сме получили стотицата ни е **2**, защото това е **единственото** **четно** **число**.  При десетиците важи **друго** правило. Там трябва да подберем **всички прости числа** в диапазона **от 2 до 7**. В нашия случай тези числа са както следва 2, 2, 3, 3, 5, 5.  При **единиците** важи правилото за **четните** **числа** и ако го следваме, получаваме, че резултатът ни е: **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 |  |

## Сума от две числа

Напишете програма която проверява **всички възможни комбинации от двойка числа** **в интервала от две дадени числа**. На изхода се отпечатва, **коя поред е комбинацията** чиито **сбор от числата** **е равен** на дадено **магическо число**. Ако няма **нито една комбинация** отговаряща на условието се изпечатва **съобщение, че не е намерено**.

### Вход

Входът се чете от конзолата и се състои от **три реда**:

* **Първи ред** – **начало на интервала** – **цяло число** в интервала **[1...999]**
* **Втори ред** – **край на интервала** – **цяло число** в интервала **[по-голямо от първото число...1000]**
* **Трети ред – магическото число – цяло число** в интервала **[1...10000]**

### Изход

На конзолата трябва да се отпечата **един ред,** според резултата:

* Ако **е намерена** **комбинация** чиито **сбор на числата е равен на магическото число**
  + "Combination N:{пореден номер} ({първото число} + {второ число} = {магическото число})"
* Ако **не е намерена** комбинация отговаряща на условието
  + "{броят на всички комбинации} combinations - neither equals {магическото число}"

### Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | **Вход** | **Изход** |
| 1  10  5 | Combination N:4 (1 + 4 = 5) | Всичски комбинации от две числа между 1 и 10 са:  1 1, 1 2, 1 3, **1 4**, 1 5, ... 2 1, 2 2, ... 4 9, 4 10, 5 1 ... 10 9, 10 10  Първата комбинация, чиито сбор на числата е равен на магическото число 5 е **четвъртата (1 и 4)** | 88  888  1000 | Combination N:20025 (112 + 888 = 1000) |
| **Вход** | **Изход** | **Обяснения** | **Вход** | **Изход** |
| 23  24  20 | 4 combinations - neither equals 20 | Всичски комбинации от две числа между 23 и 24 са: 23 23, 23 24, 24 23, 24 24 (общо 4)  Няма двойки числа, чиито сбор е равен на магическото 20 | 88  888  2000 | 641601 combinations - neither equals 2000 |

## Банкноти и монети

Имаме банкноти и монети по **1лв.**, по **2лв.** и по **5лв.** Да се напише програма, която прочита въведените от потребителя брой банкноти и монети и **сума**, и извежда на екран всички възможни начини по които сумата може да се изплати с наличните банкноти.

### Вход

Входът се чете от **конзолата** и съдържа **точно 4 реда**:

1. **Брой** монети по **1лв.** - **цяло положително число;**
2. **Брой** монети по **2лв.** - **цяло положително число;**
3. **Брой** банкноти по **5лв.** - **цяло положително число;**
4. **Сума** - **цяло положително число** в интервала [**1…1000**];

### Изход

Да се отпечатат на конзолата **всички комбинации от дадените номинали, образуващи сумата**, форматирани по следния начин:

* **"{бр. 1лв.} \* 1 lv. + {бр. 2лв.} \* 2 lv. + {бр. 5лв.} \* 5 lv. = {сума} lv."**

### Примерен вход и изход

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3  2  3  10 | 0 \* 1 lv. + 0 \* 2 lv. + 2 \* 5 lv. = 10 lv.  1 \* 1 lv. + 2 \* 2 lv. + 1 \* 5 lv. = 10 lv.  3 \* 1 lv. + 1 \* 2 lv. + 1 \* 5 lv. = 10 lv. |
| **Вход** | **Изход** |
| 5  3  1  7 | 0 \* 1 lv. + 1 \* 2 lv. + 1 \* 5 lv. = 7 lv.  1 \* 1 lv. + 3 \* 2 lv. + 0 \* 5 lv. = 7 lv.  2 \* 1 lv. + 0 \* 2 lv. + 1 \* 5 lv. = 7 lv.  3 \* 1 lv. + 2 \* 2 lv. + 0 \* 5 lv. = 7 lv.  5 \* 1 lv. + 1 \* 2 lv. + 0 \* 5 lv. = 7 lv. |

## Паркинг "Happy Cat"

Деси трябва да заведе котката си на ветеринар в клиниката **"**Happy Cat**"**, но паркингът се заплаща. Напишете програма, която пресмята колко общо трябва да се плати за престоя на колата на Деси на паркинга, за да заведе котката си на ветеринар. Паркингът е различен от останалите и има разнообразен ценоразпис. За всеки **четен ден** и **нечетен час**, паркингът таксува **2.50 лева**. Във всеки **нечетен ден** и **четен час** таксата е **1.25 лева**, във **всички останали случаи** се заплаща **1 лев**. Таксуването става на **всеки изминал час от деня**. Всеки един от изходите трябва да бъде закръглен до **втория знак** след десетичната запетая.

### Вход

От конзолата се четaт два реда:

* **Брой дни – цяло число** в интервала **[1 … 5]**
* **Брой часове за всеки един от дните - цяло число** в интервала **[1 … 24]**

### Изход:

Да се отпечата на конзолата**:**

* За всеки изминал ден, общата сума, която трябва да се плати – **"**Day: {индексът на деня} – {общата сума за деня} leva**"**
* Когато програмата приключи - **"Total: {общата сума за всички дни} leva"**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 2  5 | Day: 1 - 5.50 leva  Day: 2 - 9.50 leva  Total: 15.00 leva | 2 дни по 5 часа за всеки =>  Ден 1 - нечетен, 1вият час също => таксата е равна на 1 лев  Ден 1, 2рият час е четен => таксата е равна на 1.25 лева  Ден 1, 3ти час => 1 лев  Ден 1, 4ти час => 1.25 лева  Ден 1, 5ти час => 1 лев  Ден 1 => обща сума – 5.50 лева  Ден 2, 1ви час => таксата е равна на 2.50  …..  Ден 2 => обща сума – 9.50 лева  Обща сума за всички дни => 5.50 + 9.50 = 15.00 лева |
| 5  2 | Day: 1 - 2.25 leva  Day: 2 - 3.50 leva  Day: 3 - 2.25 leva  Day: 4 - 3.50 leva  Day: 5 - 2.25 leva  Total: 13.75 leva |  |

## Песента на колелетата

Прапраправнукът на Сали Яшар е получил наследство - сейф с парола - четири цифри. В него е заключена тайната на пеещите каруци. Той има автосервиз и се нуждае от реклама, затова е решил да направи такава каруца. Проблема е, че паролата е скрита в следната задача:

***„Парола ще получиш, ако знаеш едно цяло число, контролна стойност се нарича то,***

***почива в интервала от 4 до 144 включително, но да го откриеш може би ще е мъчително. ”***

Паролата има формат: "**abcd"** и контролната стойност трябва да е равна н**а a\*b + c\*d ,** но трябва да бъдат спазени следните условия:

* **при намирането на а и b: a < b**
* **при намиране на c и d: c > d .**

**Каруцата има четири колела, затова паролата ще е четвърто число, което трябва да се отпечата.**

**В случай, че НЕ се намери такова число, отпечатва се** „**No!“**.

### Вход

От конзолата се четe **едно цяло число (контролната стойност)**: **M** – **цяло число в интервала [4 … 144]**;

### Изход

Отпечатването на конзолата зависи от резултата:

* **Ако са намерени четворки числа (аbcd), отговарящи на условието, ги отпечатваме всичките с разделител интервал**: **“{а}{b}{c}{d} ”**
* **Отпечатва се един от двата реда на нов ред:**
  + - * **Ако съществува четвърта четворка се отпечатва: „Password: {а}{b}{c}{d}“**
      * **Ако НЕ са намерени такива числа или няма четвърта четворка отпечатваме**: **“No!”**

### Примерен вход и изход

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | | | |
| 11 | 1291 1342 1381 1471 1532 1561 1651 1741 1831 1921 2351 2431  Password: 1471 | Започваме да проверяваме числата последователно:  {1} {1} {1} {1}; {1} {1} {1} {2}; {1} {1} {1} {3}; …; {1} {2} {9} {1}; …  четворката 1291 **спазва условието и отпечатваме и т.н.** | | | |
| **Вход** | **Изход** | **Вход** | **Изход** | **Вход** | **Изход** |
| 139 | No! | 110 | 6987 7896  No! | 55 | 1786 2595 3585 3974 4575 4793 5754 5853 5952 6871  Password: 3974 |

## 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**. |