# Лаб: Вложени цикли – допълнително упражнение

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/1381>

## 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

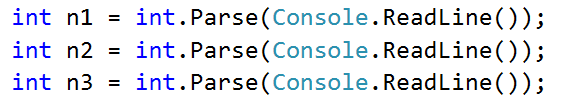
Да се отпечатат на конзолата всички валидни **трицифрени PIN кодове**, чиито цифри **отговарят** на съответните интервали.

### Примерен вход и изход

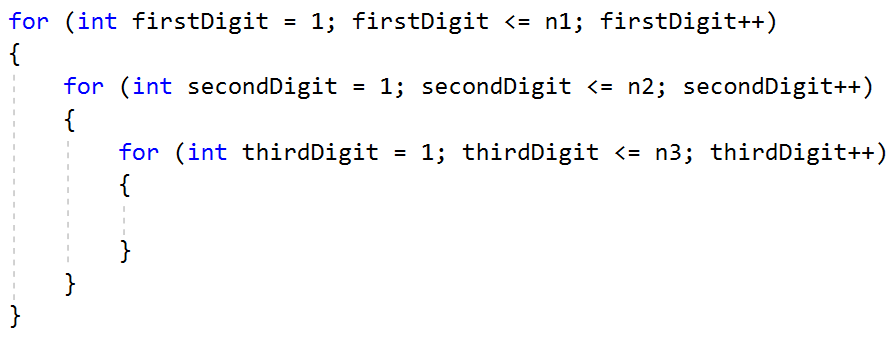
|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3  5  5 | 2 2 2  2 2 4  2 3 2  2 3 4  2 5 2  2 5 4 | Първото въведено число е **3**, отговарящо за максималната стойност на **първата цифра**.  Второто въведено число е **5**, отговарящо за максималната стойност на **втората цифра**.  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 |  |

## Насоки

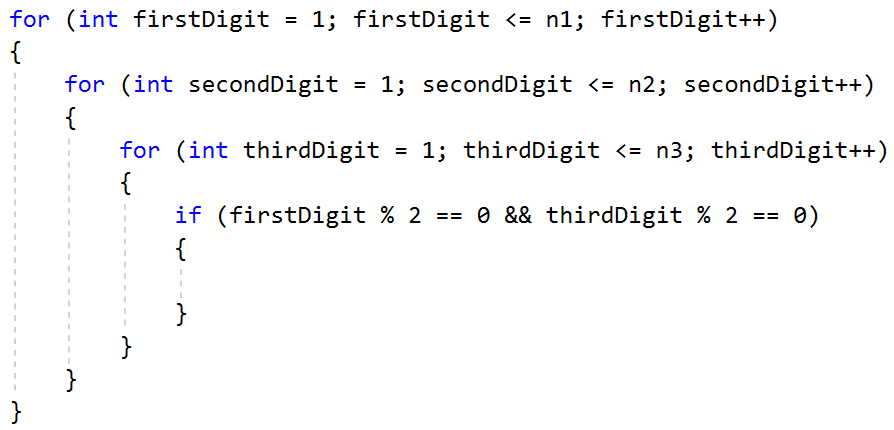
1. Прочетете входните данни от конзолата:



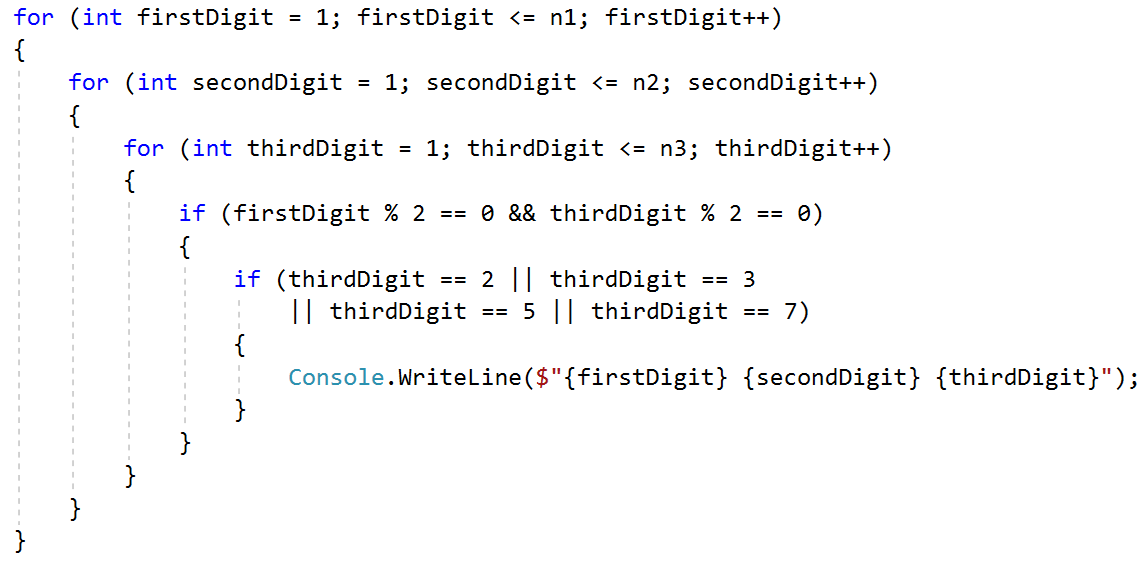
1. Направете **три вложени for цикъла**, всеки от които ще отговаря за една цифра от пин кода. Всеки цикъл трябва е от 1 до числото, което сме прочели :



1. В най-вътрешния цикъл направете **проверка дали първата и третата цифра са четни**:



1. Ако тази проверка е вярна, трябва да проверим и **дали втората цифра е просто число в интервала [2... 7].** Това може да направим с **още една if конструкция**, в която да проверяваме дали второто число е равно на едно от четирите прости числа между 2 и 7:



## 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

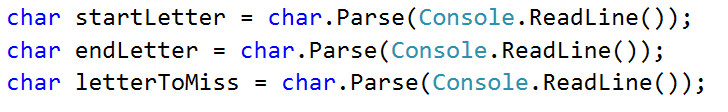
Да се отпечатат на **един ред** **всички комбинации отговарящи на условието** **плюс броят им** **разделени с интервал.**

### Примерен вход и изход

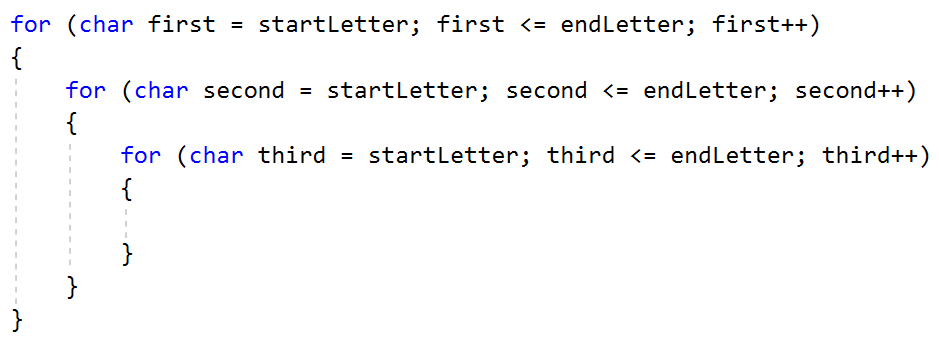
|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| a  c  b | aaa aac aca acc caa cac cca ccc 8 | Всички възможни комбинации с буквите a, b, и c са:  aaa aab aac aba abb abc aca acb acc baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb cbc cca ccb ccc  Комбинациите **съдържащи 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 | |

## Насоки

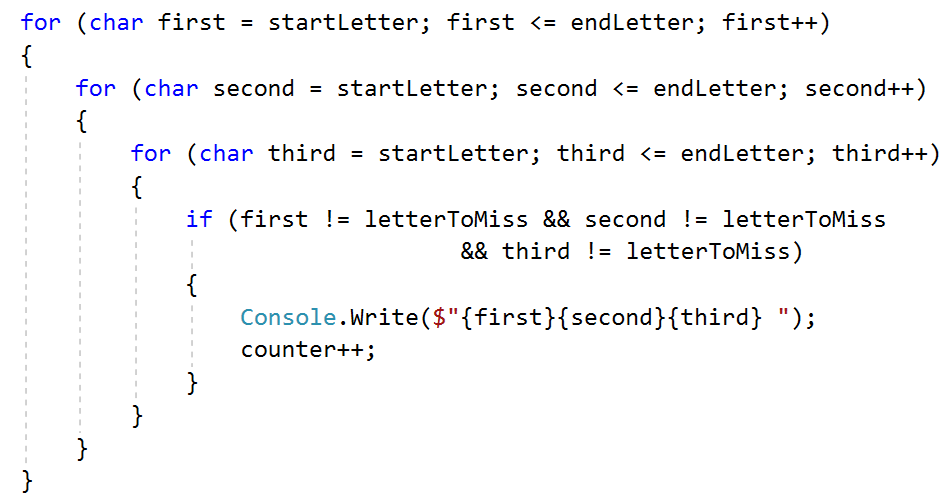
* 1. Прочетете **входните данни** от конзолата:



* 1. Направете **една променлива от тип int**, в която да държите **броя на валидните комбинации**:  
     
  2. Направете **три вложени for цикъла** – по един **за всяка буква от комбинациите**. Всеки цикъл трябва да е дефиниран **от началната буква до крайната** (които сме въвели на конзолата) и **променливата му трябва да е от тип char**:



* 1. В **най-вътрешния for цикъл** направете проверка дали **всички букви са различни от тази**, която не трябва да присъства. **Ако е така**, **отпечатайте комбинацията** в желания вид и **увеличете брояча с едно**:



* 1. След като сте отпечатали всички комбинации трябва да изпишете и стойността на брояча. Това принтиране трябва да е **извън вложените цикли**:



* 1. Тествайте програмата с различни входни стойности.

## 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**

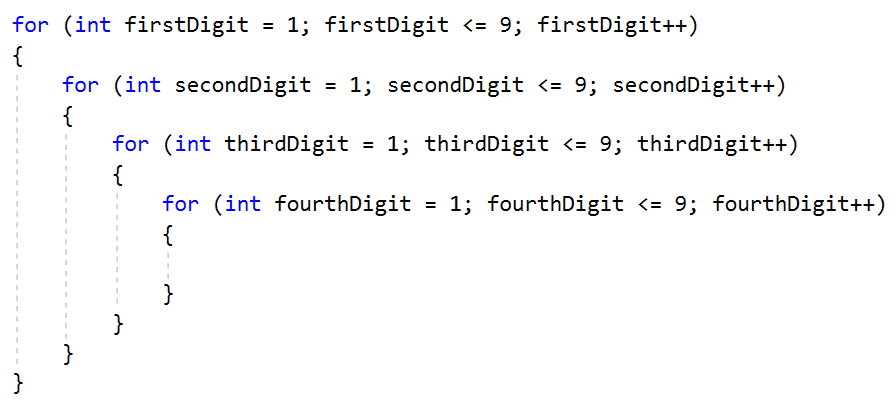
### Примерен вход и изход

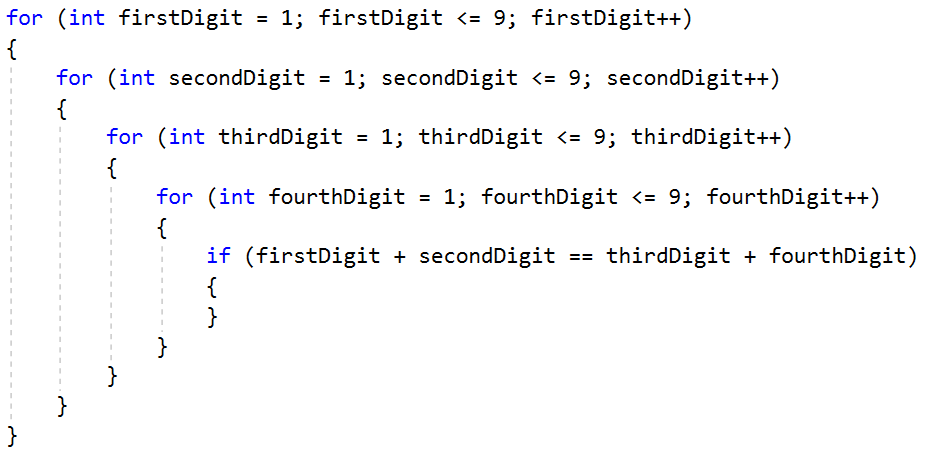
|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3 | 1212 1221 2112 2121 | И четирите числа са "щастливи"  3 / (1+2) = 1 – остатък 0 |
| 7 | 1616 1625 1634 1643 1652 1661 2516 2525 2534 2543 2552 2561 3416 3425 3434 3443 3452 3461 4316 4325 4334 4343 4352 4361 5216 5225 5234 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 |  |

## Насоки

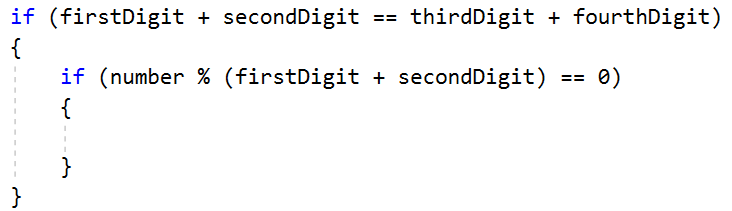
* 1. Прочетете входните данни от конзолата – едно цяло число:



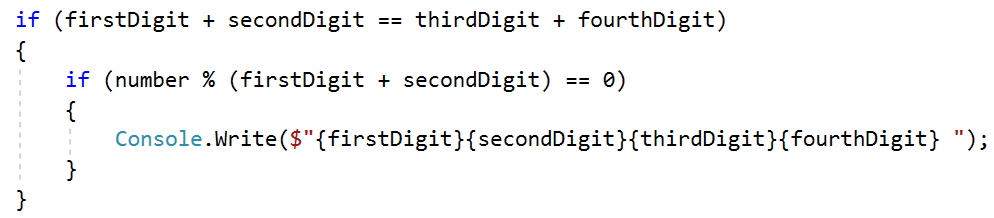
* 1. Напревете четири вложени for цикъла – всеки от 1 до 9. Всеки цикъл ще отговаря за генерирането на една цифра от четирицифреното число:  
     
  2. В най-вътрешния цикъл направете проверка дали сборът на първите две цифри е равен на сборът на вторите две:



* 1. Ако условието е изпълнено трябва да проверите дали и числото, което сте въвели се дели без остатък на сбора от първите две цифри. За целта направете вложена if проверка:



* 1. Ако това условие също е изпълнено, принтирайте "щастливото число" в желания формат:



* 1. Тествайте решението си с разични входни стойности.

## 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) | |

## Сватбени места

Младоженците искат да направят списък кой на кое място ще седи на сватбената церемония. Местата са разделени на различни сектори. **Секторите са главните латински букви като започват от A**. Във всеки сектор има определен брой редове. От конзолата се чете броят на редовете в **първия сектор** (A), като във всеки **следващ сектор редовете се увеличават с 1.** На всеки ред има **определен брой места** - тяхната номерация е представена с **малките латински букви.** Броя на местата **на нечетните редове се прочита от конзолата**, а на **четните** редове местата са **с 2 повече.**

### Вход

От конзолата се четaт **3** реда:

* **Последния сектор от секторите** - символ (**B-Z**)
* **Броят на редовете в първия сектор** - цяло число (**1-100**)
* **Броят на местата на нечетен ред** - цяло число (**1-24**)

### Изход

Да се отпечата на конзолата всяко **място на отделен ред** по следния формат:  
{сектор}{ред}{място}

**Накрая трябва да отпечата броя на всички места.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| B  3  2 | A1a  A1b  A2a  A2b  A2c  A2d  A3a  A3b  B1a  B1b  B2a  B2b  B2c  B2d  B3a  B3b  B4a  B4b  B4c  B4d  20 | Първият въведен символ е ‘В’, който представлява означението на последният **сектор, който ще има в залата**.  На вторият ред получаваме **броя на редовете в първия сектор (A) - 3**.  Накрая получаваме **броя на местата в нечетните редове - 2**.  **Първият** принтиран символ е сектора, в случая **A** или **B**;  Вторият символ представлява реда. В сектор ‘А’ има общо 3 реда. Местата на **нечетен ред са 2** и ще бъдат представени с буквите a и b, а на четен са с 2 повече = 4 - a, b, c, d. Генерираните места за сектор А са:  A1a - **нечетен** ред - имаме две места **първо** място - а  A1b - **второ** място - b  A2a - **четен** ред - имаме общо 4 места, **първо** място - а  A2b - **второ** място - b  A2c - **трето** място - c  A2d - **четвърто** място - d  A3a - **нечетен** ред - имаме две места, **първо** място - а  A3b - **второ** място - b  Местата за сектор B се генерират по същия начин. По условие във всеки следващ сектор имаме с 1 ред повече. Тоест в сектор ‘B’ ще имаме **4 реда, вместо 3.**  Печатаме всички места за сектор B.  Накрая печатаме броя на местата - в случая **20.** |
| C  4  2 | A1a  A1b  A2a  A2b  A2c  A2d  A3a  A3b  A4a  A4b  A4c  A4d  B1a  B1b  B2a  B2b  B2c  B2d  B3a  B3b  B4a  B4b  B4c  B4d  B5a  B5b  C1a  C1b  C2a  C2b  C2c  C2d  C3a  C3b  C4a  C4b  C4c  C4d  C5a  C5b  C6a  C6b  C6c  C6d  44 |  |

## Генератор за сигурни пароли

Ани се страхува от това, да не й бъде хакнат някой от профилите в социалните мрежи, затова решава да направи генератор за пароли, които да бъдат достатъчно сигурни. Вашата задача е да й помогнете да напише програма, която ще генерира тези пароли, разделени една от друга от знака "**|**".

Да се напише програма, която генерира серия от символи като в **шаблона**:

ABxyBA

като при всяко генериране на нов код, стойностите на символите се увеличават с **1**. Ако A **надхвърли 55**, се връща на 35**.** Ако B **надхвърли 96**, се връща на 64.

### Вход

От конзолата се чете 1 ред:

* На първия ред **a – цяло число** в интервала[1 … 1000]
* На втория ред **b – цяло число** в интервала[1 … 1000]
* На третия ред **максимален брой генерирани пароли** – **цяло число** в интервала [1 … 1000000]

### Ограничения:

* A **е** **символ с ASCII стойност в диапазона [35… 55]**
* B **е** **символ с ASCII стойност в диапазона [64 … 96]**
* x **e цяло число в диапазона [1… a]**
* y **e цяло число в диапазона [1… b]**

### Изход:

Да се отпечата на конзолата**:**

* Генерираният код. Ако броят на комбинациите е по-голям от максималния на кода, да се отпечата до подадената стойност, в противен случай да се отпечата до текущия брой на комбинациите.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 2  3  10 | #@11@#|$A12A$|%B13B%|&C21C&|'D22D'|(E23E(| | Понеже се достига стойността на числата **a** и **b** по-рано от постигането на максималния брой на комбинациите, програмата приключва. |
| 20  50  10 | #@11@#|$A12A$|%B13B%|&C14C&|'D15D'|(E16E(|)F17F)|\*G18G\*|+H19H+|,I110I,| | Понеже се достига максималния брой на комбинациите по-рано от стойностите на числата **a** и **b**, програмата приключва. |