

# Вложени цикли – допълнително упражнение

Задачи за допълнително упражнение към курса "[Основи на програмирането](#)" @ СофтУни.

Тествайте решенията си в judge системата: <https://judge.softuni.bg/Contests/Practice/Index/1381#0>

## 1. 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** - цяло число 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 отпечата на конзолата всички валидни **трицифрени PIN кодове**, чиито цифри **отговарят** на съответните интервали.

### Примерен вход и изход

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

## 2. 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**:

**Ред 1.** Малка буква от английската азбука за начало на интервала – от 'а' до 'z'.

**Ред 2.** Малка буква от английската азбука за край на интервала – от първата буква до 'z'.

**Ред 3.** Малка буква от английската азбука – от 'а' 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 saa sac csa 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 <b>съдържащи b</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 gi 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	

## 3. 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 отпечата **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	

## 4. 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 състои 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 състои 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 отпечата **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	

## 5. 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 отпечата 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:

- ( $\{\text{№ клиент}\} <-> \{\text{№ клиент}\}$ ) ( $\{\text{№ клиент}\} <-> \{\text{№ клиент}\}$ ) ...

## 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	(1 <-> 1)	(1 <-> 2)	(1 <-> 3)	(1 <-> 4)	(1 <-> 5)	(1 <-> 6)	(1 <-> 7)	(1 <-> 8)
8	(2 <-> 1)	(2 <-> 2)	(2 <-> 3)	(2 <-> 4)	(2 <-> 5)	(2 <-> 6)	(2 <-> 7)	(2 <-> 8)
40	(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)

## 6. 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 А.** Във всеки сектор има определен брой редове. От конзолата се чете броят на редовете в **първия сектор (А)**, 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 четат **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
В	A1a	Първият въведен символ е 'В', който представлява означението на последният <b>сектор, който ще има в залата</b> .
3	A1b	
2	A2a	На вторият ред получаваме <b>броя на редовете в първия сектор (А) - 3</b> .
	A2b	Накрая получаваме <b>броя на местата в нечетните редове - 2</b> .
	A2c	<b>Първият</b> принтиран символ е сектора, в случая <b>А</b> или <b>В</b> ;
	A2d	Вторият символ представлява реда. В сектор 'А' има общо 3 реда.
	A3a	Местата на <b>нечетен ред са 2</b> и ще бъдат представени с буквите а и b, а на четен са с 2
	A3b	повече = 4 - а, b, c, d. Генерираните места за сектор А са:
	V1a	A1a - <b>нечетен</b> ред - имаме две места <b>първо</b> място - а
	V1b	A1b - <b>второ</b> място - b
	V2a	A2a - <b>четен</b> ред - имаме общо 4 места, <b>първо</b> място - а
	V2b	A2b - <b>второ</b> място - b
	V2c	A2c - <b>трето</b> място - c
	V2d	A2d - <b>четвърто</b> място - d
	V3a	A3a - <b>нечетен</b> ред - имаме две места, <b>първо</b> място - а
	V3b	A3b - <b>второ</b> място - b
	V4a	Местата за сектор В се генерират по същия начин. По условие във всеки следващ сектор
	V4b	имаме с 1 ред повече. Тоест в сектор 'В' ще имаме <b>4 реда, вместо 3</b> .
	V4c	Печатаме всички места за сектор В.
	V4d	Накрая печатаме броя на местата - в случая <b>20</b> .

	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	

## 7. Генератор за сигурни пароли

Ани се страхува от това, да не ѝ бъде хакнат някой от профилите в социалните мрежи, затова решава да направи генератор за пароли, които да бъдат достатъчно сигурни. Вашата задача е да ѝ 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** е цяло число в диапазона [1... a]
- **y** е 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 <b>a</b> и <b>b</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

## 8. 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 четат **3** реда:

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

### Изход

Да се отпечата 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	<p>Първото въведено число е <b>3</b>, отговарящо за максималната стойност на <b>стотиците</b>.</p> <p>Второто въведено число е <b>5</b>, отговарящо за максималната стойност на <b>десетиците</b>.</p> <p>Третото въведено число е <b>5</b>, отговарящо за максималната стойността на <b>единиците</b>.</p> <p>Във всички комбинации, които сме получили стотицата ни е <b>2</b>, защото това е <b>единственото четно число</b>.</p> <p>При десетиците важи <b>друго</b> правило. Там трябва да подберем <b>всички прости числа</b> в диапазона <b>от 2 до 7</b>. В нашия случай тези числа са както следва <b>2, 2, 3, 3, 5, 5</b>.</p> <p>При <b>единиците</b> важи правилото за <b>четните числа</b> и ако го следваме, получаваме, че резултатът ни е: <b>2, 4, 2, 4, 2, 4</b>.</p>
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	

## 9. 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 печата **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)	Всички комбинации от две числа между 1 и 10 са: 1 1, 1 2, 1 3, <b>1 4</b> , 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 <b>четвъртата (1 и 4)</b>	88 888 1000	Combination N:20025 (112 + 888 = 1000)
Вход	Изход	Обяснения	Вход	Изход
23 24 20	4 combinations - neither equals 20	Всички 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

## 10. 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 отпечата на конзолата **всички комбинации от дадените номинали, образуващи сумата**, форматирани по следния начин:

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

### Примерен вход и изход

Вход	Изход
3	0 * 1 лв. + 0 * 2 лв. + 2 * 5 лв. = 10 лв.
2	1 * 1 лв. + 2 * 2 лв. + 1 * 5 лв. = 10 лв.
3	3 * 1 лв. + 1 * 2 лв. + 1 * 5 лв. = 10 лв.
10	
Вход	Изход
5	0 * 1 лв. + 1 * 2 лв. + 1 * 5 лв. = 7 лв.
3	1 * 1 лв. + 3 * 2 лв. + 0 * 5 лв. = 7 лв.
1	2 * 1 лв. + 0 * 2 лв. + 1 * 5 лв. = 7 лв.
7	3 * 1 лв. + 2 * 2 лв. + 0 * 5 лв. = 7 лв.
	5 * 1 лв. + 1 * 2 лв. + 0 * 5 лв. = 7 лв.

## 11. 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 четат 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	

## 12. 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. Проблемата 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 на  $a*b + c*d$ , но трябва да бъдат спазени следните условия:

- при намирането на a и b:  $a < b$
- при намиране на c и d:  $c > d$
- a, b, c и d са числа в интервала [1 - 9]

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 чете едно цяло число (контролната стойност): M – цяло число в интервала [4 ... 144];

## Изход

Отпечатването на конзолата зависи от резултата:

- Ако са намерени четворки числа (abcd), отговарящи на условието, ги отпечатваме всичките с разделител интервал: "{a}{b}{c}{d}"
- Отпечатва се един от двата реда на нов ред:
  - Ако съществува четвърта четворка се отпечатва: „Password: {a}{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

## 13. Четни двойки

Напишете програма, която генерира и принтира на конзолата **четирицифрени** числа, в които **първата** и **втората** двойка цифри образуват **двучифрени 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	1123 1323	Началната стойност на първата двойка цифри е 10, а на втората 20. Крайните стойности са съответно:

5		$10 + 5 = 15$
5		$20 + 5 = 25$
		Получават се следните комбинации: <b>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</b> но от тях само <b>1123</b> и <b>1323</b> са четирицифрени числа, в които <b>първата</b> част и <b>втората</b> са <b>едновременно прости числа</b> .
10	<b>1131</b>	Началната стойност на първата двойка цифри е 10, а на втората 30. Крайните стойности са съответно:
30	<b>1331</b>	
9	<b>1731</b>	$10 + 9 = 19$
6	<b>1931</b>	$30 + 6 = 36$
		Получават се следните комбинации: <b>1030 1031 1032 1033 1034 1035 1036 1130 1131 1132 ... 1330 1331 1332 1333 1334 ... 1930 1931 1932 1933 1934 1935 1936</b> но от тях само <b>1123 1331 1731 1931</b> са четирицифрени числа, в които <b>първата</b> част и <b>втората</b> са <b>едновременно прости числа</b> .

## 14. Генератор за пароли

Да се напише програма, която чете две цели числа  $n$  и  $L$ , въведени от потребителя, и генерира по азбучен ред всички възможни **пароли**, които се състоят от следните **5 символа**:

- Символ 1: цифра от **1** до  $n$ .
- Символ 2: цифра от **1** до  $n$ .
- Символ 3: малка буква измежду първите  $L$  букви на латинската азбука.
- Символ 4: малка буква измежду първите  $L$  букви на латинската азбука.
- Символ 5: цифра от 1 до  $n$ , по-голяма от първите 2 цифри.

### Вход

Входът се чете от конзолата и се състои от две **цели числа**  $n$  и  $L$  в интервала  $[1...9]$ , по едно на ред.

### Изход

На конзолата трябва да се отпечата **всички пароли** по **азбучен ред**, разделени с **интервал**.

### Примерен вход и изход

вход	изход
2 4	11aa2 11ab2 11ac2 11ad2 11ba2 11bb2 11bc2 11bd2 11ca2 11cb2 11cc2 11cd2 11da2 11db2 11dc2 11dd2
3 1	11aa2 11aa3 12aa3 21aa3 22aa3
3 2	11aa2 11aa3 11ab2 11ab3 11ba2 11ba3 11bb2 11bb3 12aa3 12ab3 12ba3 12bb3 21aa3 21ab3 21ba3 21bb3 22aa3 22ab3 22ba3 22bb3
4 2	11aa2 11aa3 11aa4 11ab2 11ab3 11ab4 11ba2 11ba3 11ba4 11bb2 11bb3 11bb4 12aa3 12aa4 12ab3 12ab4 12ba3 12ba4 12bb3 12bb4 13aa4 13ab4 13ba4 13bb4 21aa3 21aa4 21ab3 21ab4 21ba3 21ba4 21bb3 21bb4 22aa3 22aa4 22ab3 22ab4 22ba3 22ba4 22bb3 22bb4 23aa4 23ab4 23ba4 23bb4 31aa4 31ab4 31ba4 31bb4 32aa4 32ab4 32ba4 32bb4 33aa4 33ab4 33ba4 33bb4

