# Подготовка за изпит

Задачи за подготовка за **онлайн** практически приемен изпит на **14 и 15 април 2018 г.** към курса [„Основи на програмирането“ @ СофтУни](https://softuni.bg/courses/programming-basics)

В този документ са поместени избрани задачи от проведения **онлайн** изпит на **10 и 11 Март** **2018**. Дадените насоки за решение на задачите **няма** да бъдат включени в условията, които ще бъдат предоставени на приемния изпит от курса.

# Прости пресмятания

## Задача 1. PC Store

*Първа задача от изпит " Основи на програмирането" –* ***10 и 11 Март 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/968#1)

Винету искал да си купи няколко части за неговия компютър и отишъл в магазин. Понеже там нямало частите в наличност, те трябвали да ги поръчат онлайн, но от чуждестранен сайт. Той трябвало да си купи **процесор**, **видео** **карта** и **рам** **памет**, като за **процесора и видео картата** сайтът му правел няколко процента **отстъпка**. Всичко трябвало да плати в **долари**, затова трябва цените **да се обърнат в лева**, като приемем, че 1 долар = **1.57** лева.

Да се напише програма, която пресмята **колко общо пари ще му трябват в лева,** за да може да си закупи частите.

### Вход

От конзолата се прочитат 5 числа:

• На първи ред **цена в долари за процесора** – **реално** число в интервала [200.00 … 3000.00]

• На втори ред **цена в долари за видео карта** – **реално** число в интервала [100.00 … 1500.00]

* На трети ред **цена в долари за една платка рам памет** – **реално** число в интервала [80.00 ... 500.00]

• На четвърти ред **брой платки рам памет** – **цяло** число в интервала [1 ... 4]

• На пети ред **отстъпка – процент** в интервала [0.01 … 0.1]

### Изход

Да се отпечата на конзолата на един ред точно както в примерите:

• Колко **общо лева** ще му трябват, за да си закупи частите.

Сумата трябва да се **форматира** до **втория знак след десетичната запетая**.

### Насоки

1. **Прочетете** **данните от конзолата** и ги **преобразувайте в подходящия тип**
2. **Преобразувайте** цените в **долари**, спрямо **курса** от условието на задачата
3. **Пресметнете цените** на компютърните части и отчетете **отстъпката**, която магазинът му дава
4. Принтирайте общата сума на парите, които ще му трябват, за да закупи всичко

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 500  200  80  2  0.05 | Money needed - 1295.25 leva. | Цена за процесор – 500 долара  Цена за процесор в лева – 500 \* 1.57 = 785 лева  Цена за видео карта – 200 долара  Цена за видео карта в лева – 200 \* 1.57 = 314 лева  Цена за платка рам памет – 80 долара  Цена за платка рам памет в лева – 80 \* 1.57 = 125.60 лева  Обща цена за платките рам памет – 125.60 \* 2 = 251.20 лева  Цена на процесора след отстъпката – 785 – 5% = 745.75 лева  Цена на видео картата след отстъпката – 314 – 5% = 298.30 лева  Обща цена за частите – 745.75 + 298.30 + 251.20 = 1295.25 лева |
| 1200  850  120  4  0.1 | Money needed - 3650.25 leva. |  |
| 200  100  80  1  0.01 | Money needed - 591.89 leva. |  |

# Логически проверки

## Задача 2. Пазаруване

*Втора задача от изпит " Основи на програмирането" –* ***10 и 11 Март 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/968#5)*.*

Иво има определен бюджет и с него трябва да купи няколко неща от магазина. Той има нужда от твоята помощ за написването на програма, която да изчисли дали наличната му сума ще е достатъчна за да купи всичко в списъка.

Първото нещо, което той трябва да закупи е N шоколадчета, по **65ст. всяко**, второто - M литра мляко по **2.70лв** **за литър** и **35%** по-малко броя мандарини от броя на шоколадите, като се има в предвид, че една мандарина струва **20ст**.

Да се изчисли дали Иво ще успее да закупи плануваните неща и ако успее колко пари са му останали, ако парите не му стигат, да се изчисли колко пари не му достигат. ,

### Вход

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

• На първия ред е бюджетът на Иво – **реално** число в интервала **[0.0..100000.0]**

• На втория ред е **броят шоколади** - **цяло** число в интервала **[0...999]**

• На третия ред е **количеството мляко** – **реално** число в интервала **[0.0...50.0]**

### Изход

На конзолата се отпечатва **1** ред, който изглежда по следния начин:

• Ако сумата пари след пазара е **повече или равна** на бюджета:

"**You got this, {останали пари} money left!**"

• Ако сумата пари след пазара е **по-малко** от бюджета:

"**Not enough money, you need {пари} more!**"

Резултатът да се **форматира** до **втория** знак **след десетичната** запетая.

### Насоки

1. **Прочетете данните от конзолата** и ги **преобразувайте в подходящия тип**
2. **Пресметнете сумите,** които трябва да бъдат заплатени, спрямо **цената на продуктите по условие**
3. **Сравнете бюджета**, с който разполага със **общата сума**, която трябва да заплати и принтирайте **зададения текст по условие**, ако сумата е:
   1. **По-малка или равна** на бюджета
   2. **По-голяма** от бюджета

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 10 5 1.5 | You got this, 2.10 money left! | Бюджетът, с който Иво разполага е **10лв**, броят шоколади, които трябва да закупи са 5 по 65ст всеки, което е **5 бр\*0.65ст = 3.25лв** за шоколадчета, след това количеството мляко е 1.5л, парите за мляко са **1.5л\*2.70лв=4.05лв**. Броят мандарини е **35%** по-малко от броят на шоколадите (5), което излиза, че броят на мандарините е **5 - (5\*0.35) = 3 бр**.  3 бр мандарини по 20ст всяка = **3 \* 0.20 = 0.60ст**. Сумата похарчени пари се смята по следния начин:  **3.25лв** (пари за шоколад) + **4.05лв** (пари за мляко) + **0.60ст** (пари за мандарини) = **7.90лв**. |
| 3 4 2.7 | Not enough money, you need 7.29 more! |  |

# По-сложни логически проверки

## Задача 3. Слот Машина

*Трета задача от изпит " Основи на програмирането" –* ***10 и 11 Март 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/968#6)*.*

Вашата задача е да напишете програма, за да разрешите следния проблем:

От конзолата се подават последователно три **двойки** от **символ** и **цяло число** и трябва да се изведе на конзолата **комбинацията** от символи, които са се образували след като сме събрали съответно **числото** и **ASCII кода** на придружаващия го символ. Има обаче и два частни случая, в които може да приключи играта:

* Ако получите крайна комбинация от три поредни седмици (“**777**”) печелите големия джакпот и играта приключва
* Ако получите крайна комбинация от три поредни маймунски а-та (“**@@@**”), означава че автоматично губите играта

Подборът на символи от ASCII таблицата ще бъде от **32-ри** до **126-ти** символ **включително**.

От следния линк можете да видите всички нужни за задачата символи: [ASCII Table](http://www.asciitable.com/)

### Вход

От конзолата се четат шест реда:

* **символ n** – първият начален символ от **32-ри** до **126-ти** символ от ASCII таблицата **включително**
* **число n1** – броят на промените по ASCII кода на **n – цяло число в диапазона [1… 93]**
* **символ m** – вторият начален символ от **32-ри** до **126-ти** символ от ASCII таблицата **включително**
* **число m1** – броят на промените по ASCII кода на **m – цяло число в диапазона [1… 93]**
* **символ k** – третият начален символ от **32-ри** до **126-ти** символ от ASCII таблицата **включително**
* **число k1** – броят на промените по ASCII кода на **k – цяло число в диапазона [1… 93]**

### Изход

Да се отпечата на конзолата **комбинацията** от променените символи или някой от частните случаи, посочени по-горе.

### Насоки

1. **Прочетете данните от конзолата** и ги **преобразувайте в подходящия тип**
2. Съберете **броят на промените** с **ASCII стойността** на всеки от прочетените символи, за които отговарят
3. Сравнете получените сборове и принтирайте текста, **зададен по условие за отделните случаи**, ако:
   1. Всеки сбор е равен на **ASCII стойността на символа '7'**
   2. Всеки сбор е равен на **ASCII стойността на символа '@'**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| a 2 a 3 a 4 | cde | ASCII кодът на ‘a’ е равен на 97, към него добавяме 2, което изглежда така: 97+2=99, който резултат отговаря на ASCII кода на символа ‘c’.След това отново имаме символа ‘a’ с ASCII код 97, добавяме към него 3: 97+3=100, който резултат отговаря на ASCII кода на символа ‘d’ и последната сметка, която трябва да направим е към ASCII кода на третия символ да добавим третото въведено число, което е 4, което изглежда така: 97+4=101, който резултат отговаря на ASCII кода на ‘e’.И като резултат получаваме “cde”. |
| ?  1  = 3  9  7 | @@@  !!! YOU LOSE EVERYTHING !!! | ASCII кодът на ‘?’ е равен на 63, към него добавяме 1, което изглежда така: 63+1=64, който резултат отговаря на ASCII кода на символа ‘@’. След това имаме символа ‘=’ с ASCII код 61, добавяме към него 3: 61+3=64, който резултат отговаря на ASCII кода на символа ‘@’ и последната сметка, която трябва да направим е към ASCII кода на третия символ ‘9’ да добавим третото въведено число, което е 7, което изглежда така: 57+7=64, който резултат отговаря на ASCII кода на ‘@’.И като резултат получаваме “@@@”. |
| /  8  5  2  5  2 | 777 \*\*\* JACKPOT \*\*\* |  |

# Повторения (цикли)

## Задача 4. Топки

*Четвърта задача от изпит " Основи на програмирането" –* ***10 и 11 Март 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/968#9)*.*

В кутия имаме неопределен брой топки с **различни** цветове, които ни носят **различен** **брой точки**. Задачата ни е да извадим **Х** бр. топки, които ще бъдат **въведени от конзолата**, като се има в предвид, че всеки различен цвят влияе на точките ни по следния начин:

* Ако топката е “red” точките ни се **повишават** с **5**.
* Ако топката е “orange” точките ни се **повишават** с **10**.
* Ако топката е “yellow” точките ни се **повишават** с **15**.
* Ако топката е “white” точките ни се **повишават** с **20**.
* Ако топката е “black” точките ни се **делят** на **2**.

Ако топката е с какъвто и да е цвят, **различен** от по-горните, точките **не се манипулират** и програмата **продължава** да работи.

### Вход

* От конзолата се чете **1 цяло число N,** което е **броят на топките** в диапазон **(0-1000).**
* След това се четат **N** на брой **цветове** - текст

### Изход

Отпечатват се следните редове:

**“**Total points: {**всичките събрани топки**}**”**

**“**Points from red balls {**броят червени топки**}**”**

**“**Points from orange balls {**броят** **оранжеви топки**}**”**

**“**Points from yellow balls {**броят жълти топки**}”

**“**Points from white balls {**броят бели топки**}**”**

**“**Other colors picked: {**броят на избраните топки извън зададените цветове**}**”**

**“**Divides from black balls: {**броят на пътите, в които точките са били разделяни на 2**}**”**

### Насоки

1. **Прочетете данните от конзолата** – **броят на топките**, които ще се подават като вход
2. Използвайте подходяща **конструкция**, **за да прочетете N на брой пъти текст**, който ще представлява **цветовете** на топките, по които ще манипулирате точките и начин, по който да пазите техния брой
3. **Проверявайте** **всеки получен цвят** като вход и **променяйте стойността на точките**, както е описано по условие
4. Принтирайте **зададения по условие текст**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3  white  black  pink | Total points: 10  Points from red balls: 0  Points from orange balls: 0  Points from yellow balls: 0  Points from white balls: 1  Other colors picked: 1  Divides from black balls: 1 | От конзолата се въвежда числото **3**, което е **броят** на топките, които ще извадим. Първата топка е “white”, което означава, че към точките ни добавяме **20**.  Втората топка е “black”, което означава, че разделяме точките ни на **2**, което изглежда така:  **20/2=10** – новата стойност на точкте ни.  Третата топка е “pink”, която не влиза в топките, които променят точките ни и заради това програмата продължава да работи **без** да маипулира точките ни. Няма повече топки за вадене и на конзолата извеждаме получения резултат. |
| 5  red  red  ddd  ddd  ddd | Total points: 10  Points from red balls: 2  Points from orange balls: 0  Points from yellow balls: 0  Points from white balls: 0  Other colors picked: 3  Divides from black balls: 0 |  |

## Задача 5. Комбинация от символи

*Четвърта задача от изпит " Основи на програмирането" –* ***10 и 11 Март 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/968#10)*.*

От конзолата се въвежда цяло число, отговарящо на броя символи, които ще бъдат въвеждани след това. От вас се изисква да намерите най-високата ASCII сума и съответно последователността от символите, които я образуват, и да ги изведете на конзолата. Има четири основни групи, за които трябва да следите:

* **Цифри** (0-9)
* **Главни латински** букви (A-Z)
* **Малки латински** букви (a-z)
* Всички **останали** **символи** (.,?@#$% и тн.)

### Вход

От конзолата се четат 1 **число и** брой символи, **равни на числото**,като всеки символ се чете на нов ред:

* На първи ред – **N символи** в интервала **[1..19]**
* На всеки следващ ред – **N** **символa от** [**ASCII таблицата**](http://www.asciitable.com/)

### Изход

Отпечатват се два реда:

1. **"**Biggest ASCII sum is:**{стойността на ASCII сумата}"**
2. **"**Combination of characters is:**{комбинацията от символи с най-висока ASCII сума от съответната група}"**

При наличие на различни комбинации от символи с **еднаква ASCII сума** да се изведе на конзолата комбинацията от символи със следните **приоритети**: **най-висок приоритет да имат цифрите**, **след това главните букви**, **последвани от малките букви и най-нисък приоритет всички останали символи**.

### Насоки

* 1. **Прочетете данните от конзолата** – **броят на символите**, които ще се получават като вход
  2. Използвайте подходяща **конструкция**, **за да прочетете N на брой пъти символи**, който ще
  3. **За всеки прочетен символ** правете **проверка** дали неговата **ASCII стойност попада в диапазона на описаните в условието групи символи**
  4. **Намерете най-голямата сума от ASCII стойности от всички групи символи**
  5. Конструирайте **серия от проверки**, които са наредени **приоритетно**, за да принтирате най-голямата сума **в реда**, **описан по условие**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 5  9  a  V  ]  7 | Biggest ASCII sum is:112  Combination of characters ASCII sum is:97 | Първо въвеждаме броя на символите, които ще участват в сравнението, който е **5**. След това въвеждаме символа “**9**”, който има ASCII стойност – **57**, вторият въведен символ е ‘**a**’, който има ASCII стойност – **97**, третият въведен е символът “V” с ASCII стойност – **86**, четвъртият въведен символ е “]” с ASCII стойност – **93**, последният въведен символ е “7”, който има стойност – **55**, който влиза в една група с първият въведен символ - 9 .  Това представлява крайният резултат от комбинации между символи от едни и същи групи и субраните ASCII суми:  Всички **цифри** -9(57)+7(55)=112  Всички **главни латински** букви:V(86)  Всички **малки латински** букви:a(97)  Всички останали символи: ](93)  Най-голямата сума е 112 получена от ASCII стойността на ‘9’ и ASCII стойността на ‘7’, съответно и комбинацията от символи с най-голяма сума е : “97” |
| 3 C  +  + | Biggest ASCII sum is: 86  Combinations of characters ASCII sum is:++ |  |

# Чертане на фигурки с цикли

## Задача 6. Sword

*Пета задача от изпит " Основи на програмирането" –* ***10 и 11 Март 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/968#13)*.*

Да се напише програма, която прочита от конзолата **цяло число** **n** и чертае **меч** с размери като в примерите по-долу. Ширината на фигурата е **(2\*n) + 1**

### Вход

Входът е **цяло число** **n** в интервала [**4**…**50**].

### Изход

Текстови редове, изобразяващи **меч**, точно както в примерите.

### Насоки

1. **Прочетете данните от конзолата** – числото **n**, което ще се подава като вход
2. Първият ред съдържа **отляво и отдясно** точно leftRight = **n – 1** символи от тип **#**.Средната (празна) част е с широчина **spaceWidth = 3 символа**
3. Можем да създадем нова променлива за ширина на фигурката **width = leftRight \* 2 + 3**
4. Всеки следващ ред **до n / 2** съдържа отляво и отдясно с **1 символ по-малко** от предходния. Средната част се **увеличава с 2** единици
5. Следващите 4 реда съдържат буквите **SOFT**, като всеки един от тях има **(n - 1) / 2** повторения на символи от тип **#**. Тук празните места между началото на меча и буквите са с широчина **(spaceWidth - 3) / 2**
6. Всеки следващи ред **до n-4** добавяме на конзолата **същият брой** символи от тип **#**. **Празното място** е отпечатано с ширина **(spaceWidth – 3 + 1)**
7. Следващите три реда повтаряме зависимостта от точка 4. Символите, които въвеждаме в средата са **UNI**
8. **Принтираме на конзолата** символа **@** **в началото и в края**, а между тях символи от тип **=** **с ширина width – 2**
9. Всеки следващ ред **до n / 2** съдържа отляво и отдясно с **((n+3)/2) на брой символа** от тип **#**. Оградени са с по един символ от тип **|**. Средната част е с ширина **width – (((n+3)/2) \* 2) - 2**
10. На последния ред използваме същата зависимост като **точка 9**. Символа **| е заменен със \ / от ляво и дясно**

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 4 | ###/^\###  ##. .##  #. .#  #| S |#  #| O |#  #| F |#  #| T |#  #| |#  #| U |#  #| N |#  #| I |#  @=======@  ###| |###  ###| |###  ###\./### | 7 | ######/^\######  #####. .#####  ####. .####  ###. .###  ###| S |###  ###| O |###  ###| F |###  ###| T |###  ###| |###  ###| |###  ###| |###  ###| U |###  ###| N |###  ###| I |###  @=============@  #####| |#####  #####| |#####  #####| |#####  #####\.../##### |

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 5 | ####/^\####  ###. .###  ##. .##  ##| S |##  ##| O |##  ##| F |##  ##| T |##  ##| |##  ##| U |##  ##| N |##  ##| I |##  @=========@  ####| |####  ####| |####  ####\./#### | 6 | #####/^\#####  ####. .####  ###. .###  ##. .##  ##| S |##  ##| O |##  ##| F |##  ##| T |##  ##| |##  ##| |##  ##| U |##  ##| N |##  ##| I |##  @===========@  ####| |####  ####| |####  ####| |####  ####\.../#### |
|  |  |  |  |

# Задача 7. Стрела

*Пета задача от изпит " Основи на програмирането" –* ***10 и 11 Март 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/968#14)*.*

Да се напише програма, която прочита от конзолата **цяло число** **n** и чертае **стрела** с размери като в примерите по-долу. Ширината на фигурата е **n+5**

### Вход

Входът е **цяло, четно число** **n** в интервала [**4**…**60**].

### Изход

Текстови редове, изобразяващи **стрела**, точно както в примерите.

### Насоки

1. **Прочетете данните от конзолата** – числото **n**, което ще се подава като вход
2. Можем да създадем нова променлива за ширина на фигурката **width =** **n + 5**
3. Първият ред съдържа **отляво и отдясно** точно leftRight = (**width – 1) / 2** символи от тип **\_**.В средата принтираме символа **^**.
4. На вторият ред принтираме **leftRight – 1** символа от тип **\_**. В средата принтираме **/|\**.
5. Всеки следващ ред **до n / 2** съдържа отляво и отдясно **\_** с ширина **leftRight с** **1 символ по-малко** от предходния. В средата принтираме символите **|||**. Между ръба на стрелата и нейната централна част имаме **i** **на брой** **„.“ символа**.
6. На следващите два реда завършваме острието на стрелата използвайки **статични** **/..|||..\** и **/.|||.\** символи.
7. Всеки следващ ред **до n** принтираме на конзолата **(n / 2 + 1)** на брой **\_ от ляво и отдясно**. **В средата** принтираме централната част **|||**.
8. Следващият ред го принтираме със същата ширина от ляво и от дясно както в **точка 7**. **Вместо |||** принтираме **~~~**.
9. Всеки следващ ред **до n / 2** принтираме от ляво и от дясно **width – 4 – 1 – i** символа от тип **\_**. Пробвайте да довършите алгоритъма за да изкарате необходимия резултат.

### Примерен вход и изход

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** | | **Вход** | **Изход** |
| 4 | \_\_\_\_^\_\_\_\_  \_\_\_/|\\_\_\_  \_\_/|||\\_\_  \_/.|||.\\_  /..|||..\  \_/.|||.\\_  \_\_\_|||\_\_\_  \_\_\_|||\_\_\_  \_\_\_|||\_\_\_  \_\_\_|||\_\_\_  \_\_\_~~~\_\_\_  \_\_//!\\\_\_  \_//.!.\\\_ | 6 | \_\_\_\_\_^\_\_\_\_\_  \_\_\_\_/|\\_\_\_\_  \_\_\_/|||\\_\_\_  \_\_/.|||.\\_\_  \_/..|||..\\_  \_/..|||..\\_  \_\_/.|||.\\_\_  \_\_\_\_|||\_\_\_\_  \_\_\_\_|||\_\_\_\_  \_\_\_\_|||\_\_\_\_  \_\_\_\_|||\_\_\_\_  \_\_\_\_|||\_\_\_\_  \_\_\_\_|||\_\_\_\_  \_\_\_\_~~~\_\_\_\_  \_\_\_//!\\\_\_\_  \_\_//.!.\\\_\_  \_//..!..\\\_ | | 8 | \_\_\_\_\_\_^\_\_\_\_\_\_  \_\_\_\_\_/|\\_\_\_\_\_  \_\_\_\_/|||\\_\_\_\_  \_\_\_/.|||.\\_\_\_  \_\_/..|||..\\_\_  \_/...|||...\\_  \_\_/..|||..\\_\_  \_\_\_/.|||.\\_\_\_  \_\_\_\_\_|||\_\_\_\_\_  \_\_\_\_\_|||\_\_\_\_\_  \_\_\_\_\_|||\_\_\_\_\_  \_\_\_\_\_|||\_\_\_\_\_  \_\_\_\_\_|||\_\_\_\_\_  \_\_\_\_\_|||\_\_\_\_\_  \_\_\_\_\_|||\_\_\_\_\_  \_\_\_\_\_|||\_\_\_\_\_  \_\_\_\_\_~~~\_\_\_\_\_  \_\_\_\_//!\\\_\_\_\_  \_\_\_//.!.\\\_\_\_  \_\_//..!..\\\_\_  \_//...!...\\\_ |
| **Вход** | **Изход** | | |
| 10 | \_\_\_\_\_\_\_^\_\_\_\_\_\_\_  \_\_\_\_\_\_/|\\_\_\_\_\_\_  \_\_\_\_\_/|||\\_\_\_\_\_  \_\_\_\_/.|||.\\_\_\_\_  \_\_\_/..|||..\\_\_\_  \_\_/...|||...\\_\_  \_/....|||....\\_  \_\_\_/..|||..\\_\_\_  \_\_\_\_/.|||.\\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_|||\_\_\_\_\_\_  \_\_\_\_\_\_~~~\_\_\_\_\_\_  \_\_\_\_\_//!\\\_\_\_\_\_  \_\_\_\_//.!.\\\_\_\_\_  \_\_\_//..!..\\\_\_\_  \_\_//...!...\\\_\_  \_//....!....\\\_ | | |

# По-сложни цикли

## Задача 8. Специални комбинации

*Шеста задача от изпит " Основи на програмирането" –* ***10 и 11 Март 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/968#15)*.*

Напишете програма, която генерира комбинации спрямо въведени числа – граници 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**).

### 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 отпечата на конзолата всички **трицифрени** **числа**, на които всяка една част отговаря на **условията** по-горе.

### Насоки

1. **Прочетете данните от конзолата** – цифрите за максимална стойност на **стотиците, десетиците и единиците**.
2. С **три вложени цикъла** ще създадем и използваме променливи (**p1, p2, p3**), който **ще започват от 2**. Тяхната максимална стойност можем да получим от въведените променливи в **точка 1**.
   1. Принтираме на конзолата всеки резултат, в който **p1** и **p3 са четни** и **p2** е **просто число**, в диапазона **(2 – 7).**

### 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 |  |

## Задача 9 . Prime pairs

*Шеста задача от изпит " Основи на програмирането" –* ***10 и 11 Март 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/968#16)*.*

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.

### Насоки

1. **Прочетете данните от конзолата** – числата за начална стойност на **първата** и на **втората двойка**. Прибавме и техния лимит.
2. С **два вложени цикъла** ще създадем и използваме променливи (**p1, p2**). **Първият цикъл** ще започва от въведената **първа двойка в точка 1**. Цикълът ще се изпълнява **до достигане на подадения лимит**. **Вторият** ще се изпълнява **между стойността на втората двойка и нейни лимит**.
   1. Принтираме на конзолата всеки резултат, в който **p1** и **p2 са прости 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**. |