# Подготовка за изпит

Задачи за подготовка за **онлайн** практически приемен изпит на **28 и 29 Юли 2018 г.** към курса [„Основи на програмирането“ @ СофтУни](https://softuni.bg/courses/programming-basics)

В този документ са поместени избрани задачи от предишни издания на курса "Основи на програмирането". 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. Живот на котките

*Трета задача от изпит " Основи на програмирането" –* ***16 и 17 Юни 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/1045#4)*.*

Средната продължителност на живота на котките зависи от породата, като **6 човешки месеца се равняват на 1 котешки месец**. В таблицата са дадени различни породи котки и продължителността на живота им спрямо пола.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Порода** | **British**  **Shorthair** | **Siamese** | **Persian** | **Ragdoll** | **American Shorthair** | **Siberian** |
| **Мъжки пол**  **(m)** | 13 години | 15 години | 14 години | 16 години | 12 години | 11 години |
| **Женски пол**  **(f)** | 14 години | 16 години | 15 години | 17 години | 13 години | 12 години |

**Напишете програма, която изчислява колко котешки месеца живее всяка котка. При въвеждане на невалидна порода да се отпечатва:** {порода} is invalid cat!

## Вход

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

* **Порода на котката – текст – една от възможностите: "British Shorthair", "Siamese", "Persian", "Ragdoll", "American Shorthair" или "Siberian"**
* **Пол на котката – символ – 'm' или 'f'**

## Изход

Да се **отпечата** на конзолата:

**{котешки месецa} cat months**

**Където резултатът е закръглен до най-близкото цяло число надолу.**

## Насоки

1. **Прочетете данните от конзолата.**
2. **Запазете** получените порода и пол **в две променливи**.
3. **Проверете породата котка**.
4. **Към всяка проверка на породата** трябва да **проверите и какъв е нейният пол**.
5. **Изчислете годините** спрямо **породата** и **пола** на котката използвайки таблицата, която Ви е дадена. Вземете под внимание, че 6 човешки месеца се равняват на 1 котешки месец.
6. Ако породата котка, която получите е различна от изброените в таблицата трябва да отпечатате   
   "{породата} is invalid cat!"
7. Ако породата е коректно подадена от потребителя трябва да отпечатате изчислени котешки месеци.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| Persian  m | 28 cat months | Персийската мъжка котка живее 14 човешки години, което е 14 \* 12 = 168 човешки месеца => 168 / 6 = 28 котешки месеца | |
| Siamese  f | 32 cat months | Ragdoll  f | 34 cat months |
| Siberian  m | 22 cat months | Tom  m | Tom is invalid cat! |

# While – цикли

## Задача 4. Приют за котки

*Шеста задача от изпит " Основи на програмирането" –* ***16 и 17 Юни 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/1045#10)*.*

В приют за котки три пъти дневно всяка котка получава определено количество храна спрямо теглото си**.** Ще получите общото количество храна, което е закупено за котката и на всеки следващ ред – колко грама изяжда за хранене.Да се напише програма, която проверява **дали количеството храна, което е закупено за една кoтка, е достатъчно.** Програмата трябва да приключи при получаване на командата Adopted => което значи, че котката е осиновена и трябва да се изчисли дали храната е достатъчна или не.

**Вход**

От конзолата се прочитат:

* Закупеното количество храна за една котка в килограми – **цяло число** в интервала **[1.00…100.00]**
* На всеки следващ ред ще получавате колко грама изяжда котката на всяко хранене до получаване на команда **Adopted** - **цяло число** в интервала **[10.00…1000.00]**

**Изход**

На конзолата се отпечатва **1 ред**:

* Ако количеството храна е достатъчно да се отпечата:

**"Food is enough! Leftovers: {останала храна} grams."**

* Ако количеството храна не е достатъчно да се отпечата:

**"Food is not enough. You need {нужно количество храна} grams more."**

## Насоки

1. **Прочетете входните данни** от конзолата и ги **преобразувайте в подходящия тип**. Превърнете закупеното количество храна в **грамове** и ги **запазете в една променлива**.
2. **Инициализирайте променлива** с начална стойност **нула**, към която всеки път да прибавяме храната.
3. **Инициализирайте празна стринг променлива** с име command.
4. **Направете while цикъл**, който да се върти **докато променливата command е различна от "Adopted"**.
5. **В цикъла всеки път четете стринг** и го присвоявайте към command. Направете проверка дали стринга, който сте прочели е **различен от "Adopted"**, и **ако е го преобразувайте към число** и **го прибавяйте към променливата**, която създадохме преди цикъла.
6. **След цикъла направете проверка** **дали закупеното количество е по-голямо или равно на храната**, която котката е изяла. Ако е, принтирайте, че храната е достатъчна. Ако не е - принтирайте, че храната не е достатъчна.

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 4  130  345  400  180  230  120  **Adopted** | Food is enough! Leftovers: 2595 grams. | Закупеното количество храна е 4 кг = 4 \* 1000 = 4000 грама.  Общото количество храна, което е изяла котката преди да бъде осиновена е:  130 + 345 + 400 + 180 + 230 + 120 = 1405 грама. Това количество е по-малко от първоначалното закупено (4000 грама) => остава храна, която е 4000 – 1405 = 2595 грама. |
| 3  1000  1000  1000  **Adopted** | Food is enough! Leftovers: 0 grams. |  |
| 2  999  456  999  999  123  456  **Adopted** | Food is not enough. You need 2032 grams more. |  |

# For - цикли

## Задача 5. Кредитна система

*Четвърта задача от изпит " Основи на програмирането" –* ***28 и 29 Април 2018****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/989#6)*.*

Да се напише програма, която за определен брой курсове пресмята средна оценка и взети кредити.

Първоначално трябва да се прочете едно число от конзолата, което представлява броят записани курсове. След това последователно за всеки курс се въвежда по едно число:

* **Цифрите на единиците** на това число **представляват оценката**. Те са в интервала **[2…6]**.
* **Останалите цифри са броят кредити**, които студентът взима при успешното завършване на този курс.

**Кредитната система се скалира на база оценката:**

* Студент с **оценка 2 взима 0% от кредитите.**
* Студент с **оценка 3 взима 50% от кредитите.**
* Студент с **оценка 4 взима 70% от кредитите.**
* Студент с **оценка 5 взима 85% от кредитите.**
* Студент с **оценка 6 взима 100% от кредитите.**

**Вход**

От конзолата се прочита:

На първият ред:

* **n - Броят** **курсове** - **цяло число** в интервала **[1…10]**

На следващите **n реда**:

* **Числото, представляващо кредитите и оценката** - **цяло число** в интервала **[32…306]**

**Изход**

На конзолата се отпечатват **2 редa**:

* Броят **взети** **кредити**, **форматиран** до **втората цифра след десетичната запетая**
* **Средноаритметичната** оценка за всички курсове, **форматирана до втората цифра след десетичната запетая**

## Насоки

1. Прочитаме **броя** на курсовете
2. Построяваме **цикъл**, който трябва да се изпълни **толкова на брой пъти, колкото са курсовете**
3. При всяко изпълнение на цикъла **прочитаме** **едно число**
4. Използваме **модулно деление** за да вземем цифрата на единиците, която представлява оценката от курса.
5. Използваме **целочислено деление** за да вземем **броя кредити**.
6. **Запазваме** в две външни променливи **резултата** от всеки курс, в едната променлива пазим всичките оценки, а в другата броя кредити.
7. **В тялото на цикъла** правим **условни проверки** за това **колко кредита са взети от всеки курс** и **добавяме текущите кредити към общия брой**.
8. След изпълнението на цикъла **форматираме** **изхода**, като на първия ред извеждаме броя кредити, а на втория средната аритметична оценка на студента.

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3  103  103  103 | 15.00  3.00 | Имаме общо 3 курса.  За всеки един от тях студентът е получил съответно:  🡪 103 - последна цифра 3 🡪 оценка 3  🡪 103 - без последна цифра 🡪 кредити 10.  Но понеже оценката е 3, студентът взима 50% от кредитите  🡪 3 курса \* (50% от 10 кредити) = 15 кредити.  Средната оценка на студентът е 3. |
| 5  122  156  202  214  185 | 45.00  3.80 |  |
| 2  204  206 | 34.00  5.00 |  |

# По-сложни цикли

### Задача 6. Комбинации от букви

*Шеста задача от изпит "Основи на програмирането"* ***- 18 Декември 2016****.**Тествайте решението си* [*тук*](https://judge.softuni.bg/Contests/Practice/Index/368#5)*.*

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.**

## Насоки

1. Прочетете входните данни и ги преобразувайте в подходящ тип**,** нека например са c1,c2 и c3.

**2. Инициализирайте една променлива от тип integer, с която ще броим валидните комбинации.**

3. С **три вложени for цикъла**, всеки **от c1 до c2** **ще генерираме всички възможни комбинации между от буквите в този интервал.**

4. Понеже обаче в задачата се иска да печатаме само онези, които не съдържат c3, **в най-вътрешния for цикъл** трябва да направим проверка **дали една от трите букви не е c3**. Ако е, то искаме тази комбинация да не се отпечата (може да използвате ключовата дума **continue**).

5. **Ако комбинацията е валидна,** т.е **не съдържа c3,** то т**рябва да я отпечатаме на конзолата и да увеличим стойността на брояча ни с единица**.

6. След като всички итерации приключат, трябва да отпечатаме **стойността на брояча**, която трябва да е равна на валидните комбинации.

### 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 | |