

# Подготовка за изпит

Задачи за подготовка за **онлайн** практически приемен изпит  
към курса "[Основи на програмирането](#)" @ СофтУни

## Задача 1. Пазар за плодове

Линк към Judge: <https://judge.softuni.org/Contests/Practice/Index/1654#0>

Мария решава да мине на диета и отива до близкия пазар, за да купи ягоди, банани, портокали и малини. На конзолата се въвежда цената на ягодите в лв./кг. и количеството на бананите, портокалите, малините и ягодите, които трябва да закупи. Да се напише програма, която пресмята колко пари са ѝ необходими за да плати сметката, като знаете, че:

- цената на малините е на половина по-ниска от тази на ягодите;
- цената на портокалите е с 40% по-ниска от цената на малините;
- цената на бананите е с 80% по-ниска от цената на малините.

### Вход

От конзолата се четат 5 реда:

1. Цена на ягодите в лева – реално число в интервала [0.00 ... 10000.00]
2. Количество на бананите в килограми – реално число в интервала [0.00 ... 1 0000.00]
3. Количество на портокалите в килограми – реално число в интервала [0.00 ... 10000.00]
4. Количество на малините в килограми – реално число в интервала [0.00 ... 10000.00]
5. Количество на ягодите в килограми – реално число в интервала [0.00 ... 10000.00]

### Изход

Да се отпечата на конзолата **едно** число:

- парите, които са необходими на Мария.

Резултатът да се **форматира** до вторта цифра след десетичната запетая.

### Примерен вход и изход

Вход	Изход	Обяснения
48 10 3.3 6.5 1.7	333.12	Цена на малините за килограм: <b>24</b> лв. Цена на портокалите за килограм: $24 - (0.4 * 24) = 14.4$ лв. Цена на бананите за килограм: $24 - (0.8 * 24) = 4.8$ лв. Сума за малините: $6.5 * 24 = 156$ лв. Сума за портокалите: $3.3 * 14.4 = 47.52$ лв. Сума за бананите: $4.8 * 10 = 48$ лв. Сума за ягодите: $1.7 * 48 = 81.6$ лв.

		Обща сума: $156 + 47.52 + 48 + 81.6 = 333.12$ лв.
Вход	Изход	
63.5 3.57 6.35 8.15 2.5	561.15	

## Задача 2. Семейна почивка

Линк към Judge: <https://judge.softuni.org/Contests/Practice/Index/1745#1>

Семейство Иванови планират семейната си почивка. Вашата задача е да напишете програма, която да **изчислява** дали предвидения от тях **бюджет** ще им **стигне**, като знаете колко **нощувки** са планирали, каква е **цената за нощувка** и колко **процента от бюджета** са предвидили за **допълнителни** разходи. Трябва да се има предвид, че ако **броят** на нощувките е **по-голям** от 7, **цената** за нощувка се **намаля** с 5%.

### Вход

От конзолата се четат **4 реда**:

- Бюджетът, с който разполагат – реално число в интервала [1.00 ... 10000.00]
- Брой нощувки – цяло число в интервала [0 ... 1000]
- Цена за нощувка – реално число в интервала [1.00 ... 500.00]
- Процент за допълнителни разходи – цяло число в интервала [0 ... 100]

### Изход

Отпечатването на конзолата зависи от резултата:

- Ако **сумата е достатъчна**:
  - "Ivanovi will be left with {останали пари след почивката} leva after vacation."
- Ако **НЕ е достигната сумата**:
  - "{парите нужни до достигане на целта} leva needed."

Сума трябва да се форматира до втората цифра след десетичния знак.

### Примерен вход и изход

Вход	Изход	Обяснения
800.50 8 100	Ivanovi will be left with 24.49 leva after vacation.	8 нощувки > 7, следователно има 5% отстъпка от цената за нощувка, която е 100лв. 5% от 100 е 5 лв. -> 100 – 5 -> 95 лв. за нощувка след отстъпката, 8 нощувки по 95лв. -> 760 лв.

2		2% от бюджета е предвиден за допълнителни разходи. 2% от 800.50 е 16.01 лв. $760 + 16.01 = 776.01 \leq 800.50$ , следователно парите са им достатъчни и след почивката се връщат със $800.50 - 776.01 = 24.49$ лв.
Вход	Изход	Обяснения
500 7 66 15	37.00 leva needed.	7 нощувки, няма отстъпка 7 нощувки по 66 -> 462 лв. 15% от бюджета е предвиден за допълнителни разходи. 15% от 500 е 75 лв. $462 + 75 = 537 > 500$ , следователно парите не са достатъчни. Не достигат $537 - 500 = 37$ лв.

### Задача 3. Дестинация за филм

Линк към Judge: <https://judge.softuni.org/Contests/Practice/Index/1699#4>

Режисьорът на голяма кино продукция иска да разбере дали бюджетът, който са му отпуснали ще стигне за заснемане на филма. Помогнете му, като **напишете програма**, която изчислява **колко ще му струва да заснеме филма**, като **знаете колко излиза един снимачен ден**. Цената за един ден се определя от **сезона** и **дестинацията**:

Дестинация \ Сезон	Dubai	Sofia	London
Winter	45 000 lv.	17 000 lv.	24 000 lv.
Summer	40 000 lv.	12 500 lv.	20 250 lv.

Съществуват следните данъчни облекчения/облагания:

- Ако дестинацията е Дубай – 30% отстъпка от крайната цена
- Ако дестинацията е София – цената се оскъпява с 25%

### Вход

От конзолата се четат 4 реда:

1. Бюджет на филма – реално число в диапазона [100 000.0... 2 000 000.0]
2. Дестинация – текст, с възможности "Dubai", "Sofia" и "London"
3. Сезон – текст, с възможности "Summer" и "Winter"
4. Брой дни – цяло число в диапазона [1... 40]

## Изход

На конзолата да се отпечата **един ред**:

- Ако бюджета е **достатъчен**:

"The budget for the movie is enough! We have {остатък от бюджета} leva left!"

- Ако бюджета **НЕ** е достатъчен:

"The director needs {нужна сума} leva more!"

Резултатът да се закръгли до втората цифра след десетичния знак.

## Примерен вход и изход

Вход	Изход	Обяснения
400000 Sofia Winter 20	The director needs 25000.00 leva more!	Дестинацията е <b>Sofia</b> през сезон <b>Winter</b> => цената на ден е <b>17000</b> <b>20</b> дена * <b>17000</b> = <b>34000 lv.</b> Дестинацията е <b>София</b> , което означава че <b>имаме 25% оскъпяване на цената</b> => <b>34000 + 25% = 425000 lv.</b> Бюджета 400000 - 425000 => <b>не е достатъчен</b> <b>Нужни са още 25000 lv.</b>
1000000 Dubai Summer 5	The budget for the movie is enough! We have 860000.00 leva left!	
200000 London Summer 7	The budget for the movie is enough! We have 58250.00 leva left!	

## Задача 4. Храна за домашни любимци

Линк към Judge: <https://judge.softuni.org/Contests/Practice/Index/2275#6>

Ани има два домашни любимеца - куче и котка. Напишете програма, която изготвя статистика за **храната на домашните любимци за определен брой дни**. Всеки ден кучето и котката изяждат **различно** количество от общата им храна. На всеки **трети** ден получават награда - **бисквитки**. Количеството на бисквитките е **10%** от общо **изядената** храна за **деня**.

Вашата програма трябва да отпечата **статистика** за **количеството бисквитки, които са изяли, колко процента от първоначалното количество** обща храна са **изяли** и колко процента от **изядената храна** е изяло кучето и колко е изяла котката.



## Вход

Първоначално се чете **един** ред:

- Брой дни – **цяло число** в диапазона [1...30]
- Общо количество храна – **реално число** в диапазона [0.00...10000.00]

След това **за всеки ден** се чете:

- Количество изядена храна от кучето – **цяло число** в диапазона [10...500]
- Количество изядена храна от котката – **цяло число** в диапазона [10...500]

## Изход

На конзолата да се отпечатват **четири** реда:

- "Total eaten biscuits: {количество изядени бисквитки}gr."
- "{процент изядена храна}% of the food has been eaten."
- "{процент изядена храна от кучето}% eaten from the dog."
- "{процент изядена храна от котката}% eaten from the cat."

Количеството изядени бисквитки трябва да бъде закръглено до **най – близкото цяло число**, а процентът храна трябва да бъде форматиран до **втората цифра след десетичния знак**.

## Примерен вход и изход

Вход	Изход	Обяснения
3 1000 300 20 100 30 110 40	Total eaten biscuits: 15gr. 60.00% of the food has been eaten. 85.00% eaten from the dog. 15.00% eaten from the cat.	Имаме 3 дена и общо количество храна 1000гр. На първия ден: Кучето изяжда 300гр, котката изяжда 20гр. На втория ден: Кучето изяжда 100гр, котката изяжда 30гр. На третия ден: Кучето изяжда 110гр, котката изяжда 40гр. На този ден трябва да получат и награда – бисквитки: 10% от 110 + 40 – 15гр. Общо изядена храна: 600гр. Изядена храна от кучето: 510гр, от котката: 90гр. 600гр от 1000гр = 60% от храната е изядена. 510гр от 600гр = 85% е изяло кучето. 90гр от 600гр = 15% е изяла котката.
3 500 100 30 110 25 120	Total eaten biscuits: 16gr. 84.00% of the food has been eaten. 78.57% eaten from the dog. 21.43% eaten from the cat.	

35		
----	--	--

## Задача 5. Битката на великденските яйца

Линк към Judge: <https://judge.softuni.org/Contests/Practice/Index/1637#6>

На Великден семейството на Деси се събира и тя решава да организира "битка" между великденски яйца.

Правилата на "битката" са следните:

- Участват двама играчи
- Всеки от тях започва с определен брой яйца
- При получаване на команда "one" -> първият играч печели => яйцата на втория намаляват с едно
- При получаване на команда "two" -> вторият играч печели => яйцата на първия намаляват с едно
- Играта приключва, ако някой от играчите остане без яйца или до получаване на команда "End of battle"

### Вход

Първоначално се четат два реда:

1. Брой яйца, които има първият играч - цяло число в интервала [1 ... 99]
2. Брой яйца, които има вторият играч - цяло число в интервала [1 ... 99]

След това до получаване на команда "End of battle" се чете многократно един ред:

3. Победител - текст - "one" или "two"

### Изход

Ако първият играч остане без яйца:

- "Player one is out of eggs. Player two has {брой останали яйца на втория играч} eggs left."

Ако вторият играч остане без яйца:

- "Player two is out of eggs. Player one has {брой останали яйца на първия играч} eggs left."

При команда "End of battle" да се отпечата два реда:

- "Player one has {брой останали яйца на първия играч} eggs left."
- "Player two has {брой останали яйца на втория играч} eggs left."

## Примерен вход и изход

Вход	Изход	Обяснения
5 4 one two one two two End of battle	Player one has 2 eggs left. Player two has 2 eggs left.	<b>Играч 1 има 5 яйца. Играч 2 има 4 яйца.</b>  Получаваме one -> Първият печели, а яйцата на втория остават $4 - 1 = 3$ Получаваме two -> Вторият печели, а яйцата на първия остават $5 - 1 = 4$ Получаваме one -> Първият печели, а яйцата на втория остават $3 - 1 = 2$ Получаваме two -> Вторият печели, а яйцата на първия остават $4 - 1 = 3$ Получаваме two -> Вторият печели, а яйцата на първия остават $3 - 1 = 2$  <b>Край на битката =&gt; Първият играч има 2 яйца. Вторият играч има 2 яйца.</b>
2 6 one two two	Player one is out of eggs. Player two has 5 eggs left.	<b>Играч 1 има 2 яйца. Играч 2 има 6 яйца.</b>  Получаваме one -> Първият печели, а яйцата на втория остават $6 - 1 = 5$ Получаваме two -> Вторият печели, а яйцата на първия остават $2 - 1 = 1$ Получаваме two -> Вторият печели, а яйцата на първия остават $1 - 1 = 0$  <b>Край на битката =&gt; Първият играч остава без яйца. Вторият играч има 5 яйца.</b>
6 3 one two two one one	Player two is out of eggs. Player one has 4 eggs left.	<b>Играч 1 има 6 яйца. Играч 2 има 3 яйца.</b>  Получаваме one -> Първият печели, а яйцата на втория остават $3 - 1 = 2$ Получаваме two -> Вторият печели, а яйцата на първия остават $6 - 1 = 5$ Получаваме two -> Вторият печели, а яйцата на първия остават $5 - 1 = 4$ Получаваме one -> Първият печели, а яйцата на втория остават $2 - 1 = 1$

		<p>Получаваме <b>one</b> -&gt; Първият печели, а яйцата на втория остават <b>1 - 1 = 0</b></p> <p><b>Край на битката =&gt; Вторият играч остава без яйца. Първият играч има 4 яйца.</b></p>
--	--	---

## Задача 6. Баркод Генератор

Линк: <https://judge.softuni.bg/Contests/Compete/Index/2507#5>

Техниката в магазин за коледни украси се разваля. Артикулите, които съдържат **четни числа** в своя баркод не могат да бъдат маркирани от касиерите. Вашата задача е, да напишете програма, която **генерира** всички баркодове, които **НЕ съдържат** четни цифри в себе си.

### Вход:

- **Две четирицифрени числа**, които показват обхвата на баркодовете, които трябва да промените.
- **Първи ред** – четирицифрено число – началото на обхвата. **Цяло число в интервала [1000...9999]**
- **Втори ред** – четирицифрено число – края на обхвата. **Цяло число в интервала [1000...9999]**

### Изход:

На конзолата трябва да се отпечата **всички "баркодове"**, които **НЕ съдържат четна цифра** в себе си, разделени с **интервал**.

### Примерен вход и изход:

Вход	Изход	Обяснения
2345 6789	3355 3357 3359 3375 3377 3379 3555 3557 3559 3575 3577 3579 3755 3757 3759 3775 3777 3779 5355 5357 5359 5375 5377 5379 5555 5557 5559 5575 5577 5579 5755 5757 5759 5775 5777 5779	Взимаме <b>първата, втората, третата и четвъртата</b> цифра от двете <b>четирицифрени числа</b> , съответно от първото като начална и от второто като крайна стойност. Проверяваме в интервалите между всяка от цифрите на първото четирицифрено число и всяка от цифрите на второто дали съществува четна цифра и ако съществува я игнорираме в резултата.



3256 6579	3357 3359 3377 3379 3557 3559 3577 3579 5357 5359 5377 5379 5557 5559 5577 5579
1365 5877	1375 1377 1575 1577 1775 1777 3375 3377 3575 3577 3775 3777 5375 5377 5575 5577 5775 5777