## **Задача: парички**

Преди време **Пешо си е купил биткойни**. Сега ще ходи на екскурзия из Европа **и ще му трябва евро**. Освен биткойни има **и китайски юани**. Пешо иска **да обмени парите си в евро** за екскурзията. Напишете програма, която **да пресмята колко евро може да купи спрямо следните валутни курсове**:

* **1 биткойн = 1168 лева.**
* **1 китайски юан = 0.15 долара.**
* **1 долар = 1.76 лева.**
* **1 евро = 1.95 лева.**

Обменното бюро има **комисионна от 0 до 5 процента от крайната сума в евро**.

### **Входни данни**

На функцията се подават **3 аргумента**:

* **Броят биткойни** - цяло число в интервала [**0 … 20**].
* **Броят китайски юани** - реално число в интервала [**0.00 … 50 000.00**].
* **Комисионната** - реално число в интервала [**0.00 … 5.00**].

### **Изходни данни**

На конзолата да се отпечата 1 число - **резултатът от обмяната на валутите**. Резултатът да се **форматира до втората цифра след десетичния знак**.

### **Примерен вход и изход**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 5 5 | 569.67 |

**Обяснение**:

* 1 биткойн = 1168 лева
* 5 юана = 0.75 долара
* 0.75 долара = 1.32 лева
* **1168 + 1.32 = 1169.32 лева = 599.651282051282 евро**
* **Комисионна:** 5% от 599.651282051282 = **29.9825641025641**
* **Резултат**: 599.651282051282 - 29.9825641025641 = **569.668717948718 евро**

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 20 5678 2.4 | 12442.24 | 7 50200.12 3 | 10659.47 |

### **Насоки и подсказки**

Нека отново помислим първо за начина, по който можем да решим задачата, преди да започнем да пишем код.

#### **Идея за решение**

Виждаме, че ще ни бъдат подадени **броят биткойни** и **броят китайски юани**. За **изходната стойност** е указано да бъде в **евро**. В условието са посочени и валутните курсове, с които трябва да работим. Забелязваме, че към евро можем да преобразуваме само сума в лева, следователно трябва **първо да пресметнем цялата сума**, която Пешо притежава в лева, и след това да **изчислим изходната стойност**.

Тъй като ни е дадена информация за валутния курс на биткойни срещу лева, можем директно да направим това преобразуване. От друга страна, за да получим стойността на **китайските юани в лева**, трябва първо да ги **конвертираме в долари**, а след това **доларите - в лева**. Накрая ще **съберем двете получени стойности** и ще пресметнем на колко евро съответстват.

Остава последната стъпка: да **пресметнем колко ще бъде комисионната** и да извадим получената сума от общата. Като комисионна ще ни бъде подадено **реално число**, което ще представлява определен **процент от общата сума**. Нека още в началото разделим подаденото число на 100, за да изчислим **процентната му стойност**. Нея ще умножим по сумата в евро, а резултатът ще извадим от същата тази сума. Получената сума ще отпечатаме на конзолата.

#### **Избор на типове данни**

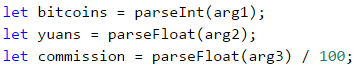
**Биткойните** са дадени като **цяло число**, следователно за тяхната стойност може да декларираме променлива преобразувана с метода **parseInt(…)**. Като брой **китайски юани и комисионна** ще получим **реално число**, следователно за тях използваме **parseFloat(…)**.

#### **Решение**

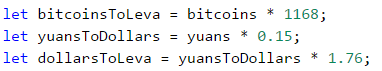
След като сме си изградили идея за решението на задачата и сме избрали структурите от данни, с които ще работим, е време да пристъпим към **писането на код**. Както и в предните задачи, можем да разделим решението на три подзадачи:

* **Прочитане на входните данни**.
* **Извършване на изчисленията**.
* **Извеждане на изход** на конзолата.

**Декларираме променливите**, които ще използваме, като отново внимаваме да изберем **смислени имена**, които подсказват какви данни съдържат те. Инициализираме техните стойности: създаваме си променливи, в който да запазим подадените на функцията стрингови аргументи, като ги конвертираме към цяло или дробно число:



Извършваме необходимите изчисления:





Накрая **пресмятаме стойността на комисионната** и я **изваждаме от сумата в евро**. Нека обърнем внимание на начина, по който можем да изпишем това: **euro -= commission \* euro** e съкратен начин за изписване на **euro = euro - (commission \* euro)**. В случая използваме **комбиниран оператор за присвояване** (**-=**), който **изважда стойността от операнда вдясно от този вляво**. Операторът за умножение (**\***) има **по-висок приоритет** от комбинирания оператор, затова изразът **commission \* euro** се изпълнява първи, след което неговата стойност се изважда.

Накрая остава да изведем резултата на конзолата. Забелязваме, че се изисква форматиране на числената стойност **до втория знак след десетичната точка**. За разлика от предходната задача, тук дори и числото да е цяло, **трябва винаги да има два знака след десетичната точка** (например **5.00**). За целта можем да използваме метода **toFixed(…)**. С него можем да преобразуваме числото в текст, запазвайки определен брой знаци след десетичната запетая:



Нека обърнем внимание на нещо, което важи за всички задачи от този тип: разписано по този начин, решението на задачата е доста подробно. Тъй като условието като цяло не е сложно, бихме могли на теория да напишем един голям израз, в който директно след получаване на входните данни да сметнем изходната стойност. Например такъв израз би изглеждал ето така:



Този код би дал правилен резултат, **но се чете трудно**. Няма да ни е лесно да разберем какво прави, дали съдържа грешки и ако има такива - как да ги поправим. По-добра практика е **вместо един сложен израз да напишем няколко прости** и да запишем резултатите от тях в променливи със подходящи имена. Така кодът е ясен, по-лесно четим и променяем.

### **Тестване в Judge системата**

Тествайте решението си тук: <https://judge.softuni.bg/Contests/Practice/Index/928#3>.