# Практически изпит по “Увод в програмирането” от 17.12.2017 г. (следобед)

Практически упражнения към курса ["Увод в програмирането" за ученици](https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Basics).

Тествайте решението си в Judge: [https://judge.softuni.bg/Contests/2646](https://judge.softuni.bg/Contests/2646/%D0%9F%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8-%D0%B8%D0%B7%D0%BF%D0%B8%D1%82-17-12-2017%D0%B3-%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B1%D0%B5%D0%B4).

## Задача 1. Обмен

Покрай нашумелите крипто валути Стефан инвеститорът решава да инвестира част от парите си в биткойн. Като всеки един инвеститор, Стефан иска да знае къде отива всеки един негов цент, но не иска да прави сметките сам, защото те отнемат време, а за това време той може да прави пари някъде другаде. Поради тази причина Стефан решава да наеме вас за написването на програма, която **да пресмята колко общо биткойн е купил с парите си и каква такса е платил за обмена.** Това обаче не е всичко. Стефан е наясно, че труда не е безплатен и затова е обещал **10% от общия брой на биткойн в сметката си** да бъдат за ваше възнаграждение.

Вашата задача е да **напишете програма, която:**

* Принтира на конзолата **колко общо биткойн има Стефан след като плати таксата за транзакцията и вашето възнаграждение.**
* Принтира на конзолата **колко долара е таксата за обмен, която Стефан е платил.** Като таксата се изчислява по следния начин:
  + **Такса в биткойн = броят биткойн който може да бъде закупен \* (сатоши за байт \* 1024) / 100 000 000**
* Принтира на конзолата **вашето възнаграждение в биткойн.**

### Вход

От конзолата се четат **3 числа**:

* На **първия ред** - **щатските долари, които Стефан ще инвестира**. **Реално число в интервала** **[1.00…1 000 000.00]**
* На **втория ред** - **цената за един биткойн в щатски долари**. **Реално число в интервала [0.00… 20 000.00]**
* На **третия ред** - **сатоши за байт. Цяло число в интервала [1 ... 1000]**

### Изход

На конзолата трябва да се отпечата следния **изход:**

* "Total bitcoin after expenses: {брой биткойни след всички разходи} BTC" - форматира се до петия знак след десетичната запетая.
* "Tax payed: {такса в долари} USD" - форматира се до втория знак след десетичната запетая.
* "Programmer`s payment: {комисионната изплатена на програмиста} BTC" - форматира се до петия знак след десетичната запетая.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 250000  10000  100 | Total bitcoin after expenses: 22.47696 BTC  Tax payed: 256.00 USD  Programmer`s payment: 2.49744 BTC | **Биткойн който може да бъде закупен = 250 000 / 10 000 = 25**  **Такса за покупката на биткойн = 25 \* (100 \* 1024) / 100 000 000 = 0.0256**  **Общо закупен биткойн = 25 - 0.0256 = 24.9744**  **Заплата за програмиста = 24.9744 \* 10% = 2.49744**  **Такса в долари = 0.0256 \* 10 000 = 256**  **Оставащ биткойн след всички разходи: 24.9744 - 2.49744 = 22.47696** |
| **Вход** | **Изход** |  |
| 10000  200  500 | Total bitcoin after expenses: 44.76960 BTC  Tax payed: 51.20 USD  Programmer`s payment: 4.97440 BTC | **Биткойн който може да бъде закупен: 50**  **Такса за покупка: 0.256**  **Общо закупен биткойн: 49.744**  **Заплата за програмиста: 4.9744**  **Такса в долари: 51.2**  **Оставащ биткойн след всички разходи: 44.7696** |

## Задача 2. Алтернативни монети

Стефчо инвеститорът е закупил **X биткойн на цена P**. След известно време цената за един биткойн става **T** и той решава, че със спечелените пари от разликата в цената ще инвестира в етереум и нео. Стефчо иска да закупи **E на брой етереум** с цена равна на **7.5%** от текущата цена за 1 биткойн и **N на брой нео** с цена равна на **40% от тази за 1 етереум.**

Вашата задача е да напишете **програма,** която пресмята дали **спечелените пари от разликата в цената на биткойн** ще бъдат достатъчни, за да може Стефчо да закупи **Е на брой етереум** и **N на брой нео**.

### Вход

Входът се чете от конзолата и се състои от **5 реда:**

1. **X** **-** **брой биткойн, които Стефчо е купил** - **реално число в интервала [1.00 …10000.00]**;
2. **P** - **началата цена за един биткойн** - **реално число в интервала [1.00 … 25000.00]**;
3. **Т** - **текущата цена за един биткойн** - **реално число в интервала [1.00 … 25000.00];**
4. **Е** - **количество етереум, което Стефчо иска да закупи** - **реално число в интервала [0.00 … 10000.00];**
5. **N - количество нео, което Стефчо иска да закупи** - **реално число в интервала [0.00 … 10000.00];**

### Изход

На конзолата трябва да се отпечата следното:

* Ако печалбата от разликата вцените **е по-малка от нужните пари** за инвестицията, **принтираме**:
  + "Stefcho does not have enough money to make this investment."
  + "He needs {недостигащи пари} more in profits." - като недостигащите пари трябва да бъдат закръглени до втория знак след десетичната запетая.
* Ако печалбата от разликата вцените **е по-голяма или равна на нужните пари** за инвестицията, **принтираме**:
  + "Stefcho bought {брой етереум} Ethereum at a price of {цена на етереум}" - броят и цената на закупения етерум трябва да бъдат закръглени до четвъртия знак след десетичната запетая.
  + "Stefcho bought {брой нео} Neo at a price of {цена на нео}" - броят и цената на закупеното нео трябва да бъде закръглен до четвъртия знак след десетичната запетая.
  + **"Stefcho has {оставащи пари след инвестицията} profits left to spend."** - **оставащите пари** трябва да бъдат закръглени до втория знак след десетичната запетая.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **коментари** |
| 2  5000  10000  2  1 | Stefcho bought 2.0000 Ethereum at a price of 750.0000  Stefcho bought 1.0000 Neo at a price of 300.0000  Stefcho has 8200.00 profits left to spend. | **Печалба от разлика в цените за 1 биткойн = 2\*10 000 - 2 \* 5000 = 10 000**  **Цена за 1 етереум = 10 000 \* 7.5% = 750**  **Цена за 1 нео = 750 \* 40% = 300**  **Обща цена на инвестицията = 750 \* 2 + 350 \* 1 = 1850**  **(10 000 > 1850) => Стефан може да направи инвестицията** |
| 2.5  6500.1  7500.6  3  4 | Stefcho does not have enough money to make this investment.  He needs 86.46 more in profits. | **Печалба = 2501.25**  **Цена за 1 етереум = 562.545**  **Цена за 1 нео = 225.018**  **Цена на инвестицията = 2587.707**  **(2501.707 < 2587.707) => Стефан няма да може да направи инвестицията** |

**Задача 3. Обменно Бюро**

Вдъхновен от интереса на хората към крипто валутите, Стефчо инвеститорът решава да създаде специално обменно бюро, чрез което хората ще могатбързо и лесно да инвестират парите си в **етереум(ETH), биткойн (BTC) или рипъл (XRP)**. Тъй като Стефчо иска да привлече повече хора, той урежда страхотни бонуси за всички, които ще обменят евро за крипто валути в неговото обменно бюро, но все пак всичко си има цена и затова минималният брой монети, които хората могат да обменят e:

* Минимална покупка на **XRP -> 80;**
* Минимална покупка на **BTC -> 0.001;**
* Минимална покупка на **ETH -> 0.0099;**

Обменния курс на валутите е:

* **XRP 1 = EUR 0.22;**
* **BTC 1 = EUR 6400**
* **ETH 1 = EUR 250**

Напишете **програма, която:**

* Приема **стринг,** който репрезентира желаната валута, в която потребителя иска да инвестира.
  + Като всичко друго освен **ETH, XRP или BTC** се счита за невалиден вход и следва да се принтира на конзолата „**{въведената валута} is not supported.**“.
* Приема **число,** което репрезентира **еврото, което потребителят иска да депозира.**
* Проверява дали еврото е **повече от 1000**:
  + Ако е**, добавя бонус 10% към него.**
* При избрана валута **XRP**:
  + Ако монетите са **повече от 1000, но по-малко от 2500**: Добавяме 5% към тях.
  + Ако монетите са **повече или равни на 2500**: Добавяме 10% към тях.
* При избрана валута **BTC:**
  + Ако монетите са повече от **10:** Добавяме **2%** към тях.

**Вход**

* **Първи ред** - **Валута в която потребителят иска да инвестира**. - **текст.**
* **Втори ред** - **Еврото, което потребителят ще инвестира в дадена валута**. **Реално число в интервала** **[1.00…1 000 000.00]**

**Изход**

На конзолата трябва да се отпечата следния **изход:**

* Ако еврото, което има потребителя:
  + Е достатъчно за да покрие минималния брой монети от дадена валута принтираме - **“Successfully purchased {брой закупени монети} {вида на закупените монети}”**
    - Броят на закупените монети се закръгля до третия знак след десетичната запетая.
  + Не е достатъчно за да покрие минималния брой монети от дадена валута принтираме -

**”Insufficient funds”.**

* Ако въведена валута е невалидна принтираме: „**EUR to {въведената валута} is not supported.**“

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| XRP  2500 | Successfully purchased 13750.000 XRP | **Евро > 1000 => добавяме 10% към еврото и то става 2750**  **XRP монети = 2750 / 0.22 = 12 500**  **Имаме повече от 2500 XRP монети => 12 500 + 10% = 13 750**  **Накрая принтираме информацията от обмена.** |
| **Вход** | **Изход** |  |
| Gosho  5000 | EUR to Gosho is not supported. | **Обменното бюро не работи с монети „Gosho“ и затова на конзолата се принтира „EUR to Gosho is not supported“.** |
| **Вход** | **Изход** |  |
| ETH  1 | Insufficient funds | **Монети които може да закупим с 1 евро =**  **1 / 250 = 0.004**  **0.004 < 0.0099 => Тази поръчка за обмен няма да се изпълни, защото еврото което искаме да обменим е твърде малко и не покрива минималните изисквания за обмен в етереум.** |

**Задача 4. Монетен рейтинг**

След създаването на крипто валутата ESTD, Стефан е заинтересован как се справя тя спрямо конкурентните монети. Стефан знае, че ESTD е в една лига с IOTA, NEO и Doge, но тази информация не му е достатъчна. Затова иска да му помогнете като напишете програма тип анкета, която:

* Чете от конзолата число **N - общия брой на участниците в анкетата;**
* Следващите **N** реда - чете **какъв вид крипто валута има всеки един** участник и **колко монети има от нея.**
* Накрая изчислява **каква част от целия пазар заема всяка една крипто валута,** като за целта са ни дадени **следните валутни курсове**:
  + **DOGE 1 = EUR 0.07**
  + **IOTA 1 = EUR 1.44**
  + **NEO 1 = EUR 28.50**
  + **ESTD 1 = EUR 25.0**

**Вход**

Входът се **чете от конзолата и съдържа**:

* **N - цяло число в интервала [1…20].**
* **N - брой реда като в тях:**
  1. **Прочитаме каква валута притежава конкретния участник в анкетата - текст**
  2. **Колко монети има от нея - реално число в интервал[0.000001…10 000 000.00].**

**Изход**

Изходът се състои от **5 реда:**

* **“Total votes = {брой на гласове}, Money in market = {обща стойност на монетите на пазара} EUR” -** като стойността трябва да бъде закръглена **до втория знак** след десетичната запетая.
* **„DOGE's contribution: {колко процента от пазара е съставен от DOGE монети}%; People who use DOGE: {брой хора който използват DOGE}“**
* **„IOTA's contribution: {колко процента от пазара е съставен от IOTA монети}%; People who use IOTA: {брой хора който използват IOTA}“**
* **„NEO's contribution: {колко процента от пазара е съставен от NEO монети}%; People who use NEO: {брой хора който използват NEO}“**
* **„ESTD's contribution: {колко процента от пазара е съставен от ESTD монети}%; People who use ESTD: {брой хора който използват ESTD}“**

**Забележка**

Всички проценти трябва да бъдат **закръглени до втория знак** след десетичната запетая.

**Примерен вход и изход**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 4  NEO  3  ESTD  1  DOGE  50  IOTA  30 | Total votes = 4, Money in market = 157.20 EUR  DOGE's contribution: 2.23%; People who use DOGE: 1  IOTA's contribution: 27.48%; People who use IOTA: 1  NEO's contribution: 54.39%; People who use NEO: 1  ESTD's contribution: 15.90%; People who use ESTD: 1 | **Обща цена на монетите в евро = 3\*28.5 + 1\*25 + 50\*0.07 + 30\*1.44 = 157.2**  **DOGE => (3.5/157.2) \* 100 = 2.23%**  **IOTA => (43.19/157.2) \* 100 = 27.48%**  **NEO => (85.5/157.2) \* 100 = 54.39%**  **ESTD => (25/157.2) \* 100 = 15.90%** |
| **Вход** | **Изход** | **Обяснения** |
| 2  ESTD  100  ESTD  150 | Total votes = 2, Money in market = 6250.00 EUR  DOGE's contribution: 0.00%; People who use DOGE: 0  IOTA's contribution: 0.00%; People who use IOTA: 0  NEO's contribution: 0.00%; People who use NEO: 0  ESTD's contribution: 100.00%; People who use ESTD: 2 | **Общо евро - 6250**  **DOGE => 0%**  **IOTA => 0%**  **NEO => 0%**  **ESTD => (6250/6250) \* 100 = 100.00%** |

## Задача 5. Нова монета

Стефан инвеститорът реши да отвори своя собствена крипто валута наречена ESTD(Easier Said Than Done) и има нужда от помощта ви. Помогнете на Стефан с програма, която ще генерира логото на неговата крипто валута в различни размери според въведеното число **N**.

### Подсказки

Ако числото е **четно** **- средните редове са N - 1.**

Ако числото е **нечетно - средните редове са N - 2.**

### Вход

Входът се **чете от конзолата** и съдържа **само едно число**:

**N - цяло число в интервала [2…27].**

### Изход

Да се отпечата **на конзолата лого**, точно както в примерите.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 3 | \\\///  \\------//  \------------/  |~~/// ESTD \\\~~|  /------------\  //------\\  ///\\\ | 4 | \\\\////  \\\------///  \\------------//  \------------------/  |---################---|  |~~~///// ESTD \\\\\~~~|  |---################---|  /------------------\  //------------\\  ///------\\\  ////\\\\ |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 6 | \\\\\\//////  \\\\\------/////  \\\\------------////  \\\------------------///  \\------------------------//  \------------------------------/  |-----########################-----|  |-----########################-----|  |~~~~~///////// ESTD \\\\\\\\\~~~~~|  |-----########################-----|  |-----########################-----|  /------------------------------\  //------------------------\\  ///------------------\\\  ////------------\\\\  /////------\\\\\  //////\\\\\\ | 5 | \\\\\/////  \\\\------////  \\\------------///  \\------------------//  \------------------------/  |----####################----|  |~~~~/////// ESTD \\\\\\\~~~~|  |----####################----|  /------------------------\  //------------------\\  ///------------\\\  ////------\\\\  /////\\\\\ |

## Задача 6. Генериране на кодове

Стефан, инвеститорът, реши да започне да генерира кодове за съмнителен бизнес с криптовалути. Вашата задача е да направите програма, която генерира кодове спрямо дадена парола.

Комбинациите са структурирани по следния начин:

* **Състоят се от общо 6 символа - 3 цифри, 2 букви и една цифра точно в този ред, като:**
  + **Първи символ - число 0 - 9**
  + **Втори символ - число 0 - 9**
  + **Трети символ - число 0 - 9**
  + **Четвърти символ - буква ‘a’ - ‘z’**
  + **Пети символ - буква ‘a’ - ‘z’**
  + **Шести символ - число от 0 - 9**
* **Стойността на сбора на всички символи трябва да е равна на паролата.**

### Вход

Входът се **чете от конзолата** и съдържа **два реда**:

* **M - цяло число в интервала [190...300] -** паролата, според която трябва да се принтират комбинациите.
* **N - цяло число в интервала [1…100] - броят кодове, които трябва да се генерират.**

### Изход

Да се принтират на конзолата всички комбинации, разделени с интервал, които отговарят на условията.

* Броят на генерираните кодове е в интервала **[0… 100]**
* Подредбата на кодовете е **лексикографска (или възходяща)**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **Обяснения** |
| 195  4 | 000aa1 000ab0 000ba0 001aa0 | **Започваме от самото начало на комбинациите:**  **0 + 0 + 0 + а(97) + а(97) + 0 = 194 -> не е равно на M => не принтираме.**  **Продължаваме да правим нови комбинации…**  **0 + 0 + 0 + a(97) + a(97) + 1 = 195**  **-> първо съвпадение => принтираме**  **…**  **други комбинации които не отговарят на изискванията**  **…**  **0 + 0 + 0 + a(97) + b(98) + 0 = 195**  **-> второ съвпадение => принтираме**  **…**  **0 + 0 + 0 + b(98) + a(97) + 0 = 195**  **трето съвпадение => принтираме**  **…**  **0 + 0 + 1 + a(97) + a(97) + 0 = 195 четвърто съвпадение => принтираме**  След като броя на съвпадения стане **N** спираме програмата. |

## Министерство на образованието и науката (МОН)

* Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".



* Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен** **лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

