# Упражнение: Условни конструкции

Задачи за упражнение към курса ["Основи на програмирането" @ СофтУни](https://softuni.bg/courses/programming-basics).

Тествайте решенията си в **judge системата**: [https://judge.softuni.bg/Contests/Compete/Index/1161](https://judge.softuni.bg/Contests/Compete/Index/1161" \l "2)

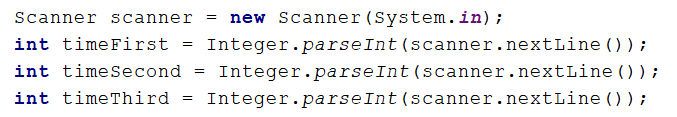
## Сумиране на секунди

Трима спортни състезатели финишират за някакъв **брой секунди** (между **1** и **50**). Да се напише програма, която чете времената на състезателите в секунди, въведени от потребителя и пресмята **сумарното им време** във формат "минути:секунди". Секундите да се изведат с **водеща нула** (2 🡪 "02", 7 🡪 "07", 35 🡪 "35").

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 35  45  44 | 2:04 | 22  7  34 | 1:03 | 50  50  49 | 2:29 | 14  12  10 | 0:36 |

### Насоки:

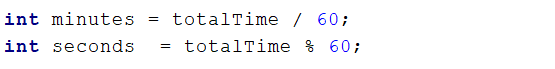
1. Прочетете входните данни (**секундите на състезателите**):



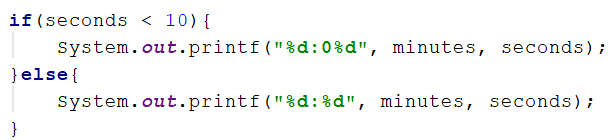
1. Създайте **нова променлива**, в която да съхранявате **сбора от секундите на тримата състезатели**:



1. След като сте намерили **сбора от секундите** трябва да ги **превърнете в минути и секунди** (например, ако сборът е **85 секунди това са 1 минута и 25 секунди, защото 1 минута има 60 секунди**). Създайте си **две нови променливи**. В първата изчислете **колко минути е сбора от секунди**, като **разделите сбора на 60**. Във втората променлива **изчислете секундите с помощта на деление с остатък (%)**. Използвайте **деление с остатък (%)**, за да вземете **остатъка при деление с 60**, което са останалите секунди. Например имате общ сбор от 134 секунди (2 минути и 14 секунди) **след целочисленото деление (/) на 60 ще получим 2 и остатък 14, който ще вземем с делението с остатък (%).**



1. След като вече знаете **колко минути и секунди** е общия сбор, трябва да ги изпринтираме в правилния формат **(минути : секунди**), като ако секундите са **по-малки от 10** трябва да изпечатаме **0 преди секундите**, в противен случай просто си изпечатваме **резултата в дадения формат**. За целта **направете проверка** (if). За принтирането може да използвате **printf и шаблонът за цяло число %d**.



**\*** **Печатането на резултата с водеща нула може да се осъществи и чрез шаблона %02d, чрез който можем да покажем, че искаме нашето цяло число (секундите) да се състои от две цифри:**



## Бонус точки

Дадено е **цяло число** – начален брой точки. Върху него се начисляват **бонус точки** по правилата, описани по-долу. Да се напише програма, която пресмята **бонус точките, които получава числото** и **общия брой точки** (числото + бонуса).

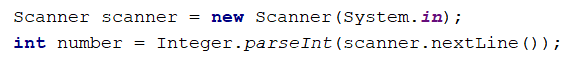
* Ако числото е **до 100** включително, бонус точките са **5**.
* Ако числото е **по-голямо от 100**, бонус точките са **20%** от числото.
* Ако числото е **по-голямо от 1000**, бонус точките са **10%** от числото.
* Допълнителни бонус точки (начисляват се отделно от предходните):
  + За **четно** число 🡪 + 1 т.
  + За число, което **завършва на 5** 🡪 + 2 т.

### Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 20 | 6  26 | 175 | 37  212 | 2703 | 270.3  2973.3 | 15875 | 1589.5  17464.5 |

### Насоки:

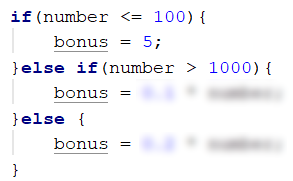
1. Прочетете входните данни (**числото**):



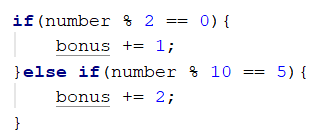
1. Създайте си **нова променлива от тип double**, в която ще си изчислявате **натрупаните бонус точки**, като и дадете **начална стойност 0**.



1. Направете **if-else-if конструкция** за първите **три проверки**, за да проверите големината числото и да изчислите бонуса.



1. Направете **нова if-else-if конструкция**, за да извършите проверките и **изчислите допълнителния бонус**. Ако числото **е четно към до момента натрупания бонус добавете 1**, а ако **завършва на 5 към бонуса добавете 2**. За да проверите дали едно число **е четно трябва да го разделите на 2 и ако получавате остатък при делението 0**, то значи числото е **четно**, но ако **получите остатък 1**, това означава, че числото е **нечетно**. Например числото 34 е четно, защото 34 / 2 = 17 и остатъкът е 0, а числото 35 е нечетно, защото 35 / 2 = 17 с остатък 1. За да проверите дали едно число завършва на 5 трябва **да разделите числото на 10** и ако **получите остатък при делението 5**, то значи числото завършва на 5. Например числото 245 / 10 = 24 с остатък 5.



1. Принтирайте **на два реда** резултатите. На първия ред **натрупания бонус**, а на втория **крайното число**, което ще намерите като **съберете началния брой точки и бонуса**.



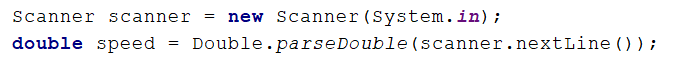
## Информация за скоростта

Да се напише програма, която **чете скорост** **(реално число)**, въведена от потребителяи отпечатва **информация за скоростта**. При скорост **до 10** (включително) отпечатайте “**slow**”. При скорост **над 10** и **до 50** отпечатайте “**average**”. При скорост **над 50** и **до 150** отпечатайте “**fast**”. При скорост **над 150** и **до 1000** отпечатайте “**ultra fast**”. При по-висока скорост отпечатайте “**extremely fast**”. Примери:

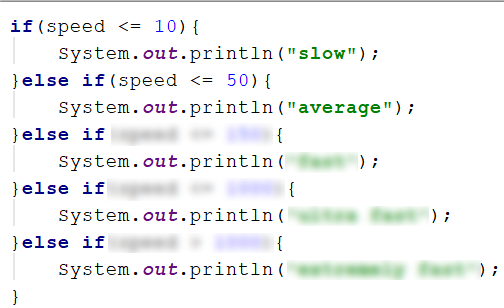
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 8 | slow | 49.5 | average | 126 | fast | 160 | ultra fast | 3500 | extremely fast |

### Насоки:

1. Прочетете входните данни (**скоростта**):



2. Използвайте if-else-if конструкция, за да проверите **стойностите за скоростта**.



## Конвертор за мерни единици

Да се напише програма, която **преобразува разстояние** между следните **3 мерни единици**: mm, cm, m. Използвайте съответствията от таблицата по-долу:

|  |  |
| --- | --- |
| **входна единица** | **изходна единица** |
| **1** meter (**m**) | **1000** millimeters (**mm**) |
| **1** meter (**m**) | **100** centimeters (**cm**) |

Входните данни се състоят от три реда, въведени от потребителя:

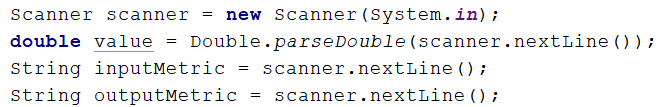
* **Първи ред:** число за преобразуване - **реално число**
* **Втори ред:** входна мерна единица - **текст**
* **Трети ред:** изходна мерна единица (за резултата) - **текст**

На конзолата да се отпечата **резултатът от преобразуването** на мерните единици **форматиран** до **третия знак след десетичната запетая**.

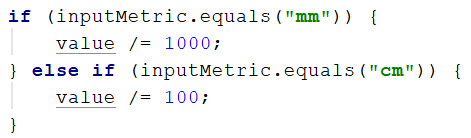
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 12  mm  m | 0.012 | 150  m  cm | 15000.000 | 45  cm  mm | 450.000 |

### Насоки:

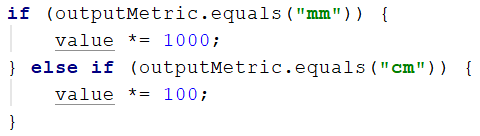
1. Прочетете входните данни (**числото за преобразуване, входна мерна единица и изходна мерна единица**):



1. Направете **if-else-if конструкция**, за да конвертирате числото от **входната мерна единица към метри**. **Превръщаме в метри, защото в таблицата имаме превръщанията от метри в останалите мерни единици.** Ако входната мерна единица е **милиметри, за да ги превърнем в метри трябва числото да го разделим на 1000**. Аналогично направете **проверките и изчисленията за сантиметрите**, като следвате стойностите за преобразуване от една мерна единица в друга от таблицата.



1. Направете нова **if-else-if конструкция**, чрез която ще преобразуваме **метрите, които изчислихме в предната стъпка, в изходната мерна единица.** За да преобразуваме **от метри в милиметри трябва числото да го умножим по 1000**. Аналогично направете проверките и изчисленията **за сантиметрите**, като следвате стойностите за преобразуване от една мерна единица в друга от таблицата.



1. Принтирайте преобразуваното число. За да го закръглите **до третия знак след десетичната запетая**, използвайте **printf и %.3f**.



## Време + 15 минути

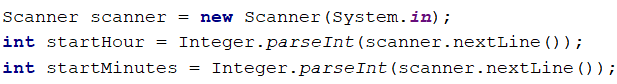
Да се напише програма, която **чете час и минути** от 24-часово денонощие, въведени от потребителя и изчислява колко ще е **часът след 15 минути**. Резултатът да се отпечата във формат часове:минути. Часовете винаги са между 0 и 23, а минутите винаги са между 0 и 59. Часовете се изписват с една или две цифри. Минутите се изписват винаги с по две цифри, с **водеща нула,** когато е необходимо.

### Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 1  46 | 2:01 | 0  01 | 0:16 | 23  59 | 0:14 | 11  08 | 11:23 | 12  49 | 13:04 |

### Насоки:

1. Прочетете входните данни (**началнния час и началните минути**):



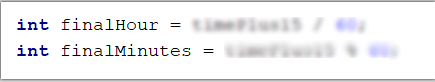
1. Превърнете получения час в минути. За целта си създайте нова променлива, в която ще направите изчислението. Ако часът е 1:45 това са 1 час \* 60 минути + 45 минути = 105 минути общо.



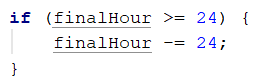
1. Създайте нова променлива, в която да добавите 15 минути към началния час, който превърнахте в минути. Ако часът ви е 1:25 (85 минути и добавите 15 минути = 100 минути).



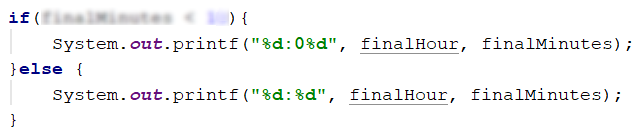
1. След като сте изчислили общото време след 15 минути трябва новополучените минути след добавянето да ги превърнете обратно в часове и минути (100 минути = 1 час и 40 минути). За да получите часовете трябва да разделите общите минути на 60, а за да получите минутите трябва да вземете остатъка от делението на общите минути на 60. Създайте си две променливи и извършете действията в тях.



1. Понеже часът след добавяне на 15-те минути може да стане със стойност 24 или повече, трябва да направите следната проверка:



1. Принтирайте вече изчислените краен час и крайни минути, като спазвате формата часове:минути и минутите да се принтират с водеща нула, т.е. ако минутите са по-малко от 10 да се принтира пред тях 0. За целта направете проверка и проверете минутите.



**\*Печатането на резултата с водеща нула може да се осъществи и чрез шаблона %02d, чрез който можем да покажем, че искаме нашето цяло число (минутите) да се състои от две цифри:**



**Примерни изпитни задачи**

## Трима братя

Трима братя решили да изненадат баща си, като почистят заедно гаража му. Големият брат чисти сам за определени часове - **А**. Средният брат чисти сам за **В** часа. По-малкият брат чисти сам за **С** часа. Бащата отива на риболов и няма да го има **D** часа. Към общото време за почистване се **добавят 15% за почивка.**

Напишете програма, която пресмята **дали тримата братя могат да изчистят заедно гаража и да изненадат баща си или не.**

### Вход

Входът се **чете от конзолата** и се състои от **4 реда**:

* **Времето на първият брат за чистене сам** – **реално число** в интервала **[0.00 … 99.00]**
* **Времето на вторият брат за чистене сам** - **реално число** в интервала **[0.00 … 99.00]**
* **Времето на третият брат за чистене сам** - **реално число** в интервала **[0.00 …99.00]**
* **Времето за риболов на бащата** - **реално число** в интервала **[0.00 … 99.00]**

### Изход

На конзолата трябва да се **отпечатат два реда**:

1. Времето за чистене и почивка, **форматирано до втори знак**: **"Cleaning time: {Времето за чистене }"**
2. Има ли изненада или не:

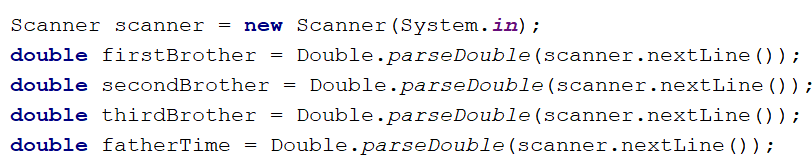
* Ако братята **СА изненадали бащата (time left >= 0)**: "Yes, **there is a surprise -** time left -> {остатък} hours." - резултатът трябва да е **закръглен към по-малко цяло число** (**пр. 1.90 -> 1**).
* Ако братята **НЕ СА изненадали бащата (time left < 0)**: "No, **there isn'сt a surprise** - shortage of time -> {недостиг} hours." - резултатът трябва да е **закръглен към по-голямо цяло число** (**пр. 1.10 -> 2**).

### Примерен вход и изход

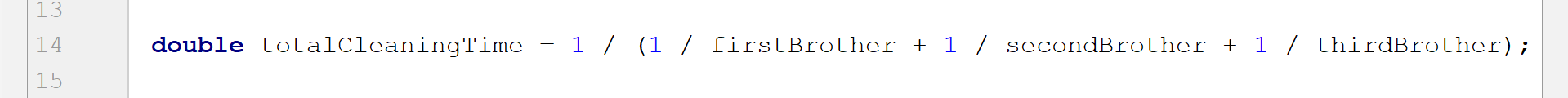
|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3  4  5  3.6 | Cleaning time: 1.47  Yes, there is a surprise - time left -> 2 hours. | **Приемаме, че цялата работа е 1.**  **Първият брат върши 1 / 3 част от работа.**  **Вторият брат върши 1 / 4 част от работа.**  **Третият брат върши 1 / 5 част работа.**  **общо време** **= 1 / (1 / 3 + 1 / 4 + 1 / 5) =** **1.27659574468085 часа**  **време с почивка = 1.27659574468085 \* 1.15 =** **1.46808510638298 часа**  **оставащо време:** **3.6 - 1.46808510638298 =** **2.13191489361702** **= 2 часа остават** |
| **Вход** | **Изход** | **Обяснения** |
| 3.2  2.5  1.24  0.5 | Cleaning time: 0.76  No, there isn't a surprise - shortage of time -> 1 hours. |  |

### Насоки:

1. Прочетете входните данни (**времето, за което чистят тримата братя и времето, в което бащата е на риболов**):



1. Създайте **нова променлива от тип double** и в нея изчислете **общото време за чистене на братята**:



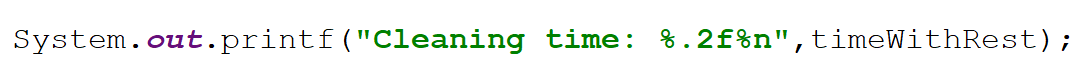
1. Създайте **нова променлива от тип double**, в която ще изчислите **времето за чистене заедно с почивката**:



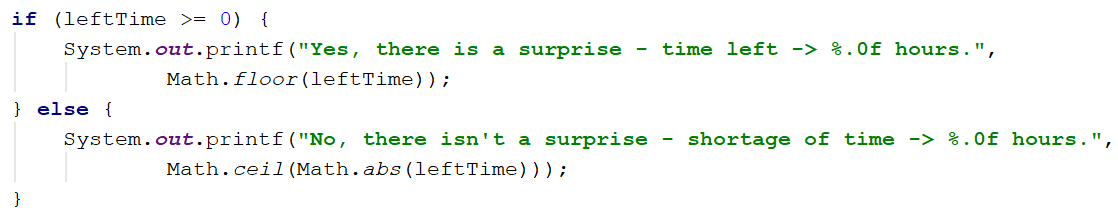
1. Създайте **нова променлива** и в нея изчислете колко е оставащото време:



1. Принтирайте **първия ред** от изхода (общото време за чистене):



1. Направете проверка за оставащото време, **ако то е положително или 0 значи са изненадали бащата, ако е отрицателно – няма изненада**. За да закръглите към **най-близкото цяло число надолу** използвайте **Math.floor**, a към **най-близкото число нагоре** използвайте **Math.ceil**. С **Math.abs** може да вземете **абсолютната стойност на едно число**.



## Хореография

Група танцьори се подготвя за финално състезание. Те трябва да научат нова хореография. Танцът се състои от **N - на брой стъпки**, които се **разпределят между танцьорите**. Цялата хореография трябва да се научи за **определен брой дни**. **Всички** танцьори могат да научат **не повече от 13% от общите стъпки на ден**. Да се напише програма която пресмята **дали танцьорите ще успеят да научат новия танц** и **по колко процента от стъпките следва да научи всеки един от тях**.

**При изчисляване на процента стъпки на ден, числото трябва да се закръгли към най - близкото цяло число нагоре.**

### Вход

От конзолата се четат **3 реда**:

1. **Брой стъпки - цяло число в интервала [1 … 100 000]**
2. **Брой танцьори - цяло число в интервала [1 … 50]**
3. **Брой дни за учене - цяло число в интервала [1 … 31]**

### Изход

Отпечатването на конзолата зависи от резултата:

* Ако общият процент стъпки **са по-малко или равни на 13%** отпечатваме:
  + **"Yes, they will succeed in that goal! {процент стъпки които трябва да научи всеки един танцьор на ден}%."**
* Ако общият процент стъпки **са повече от 13%** отпечатваме:
  + **"No, they will not succeed in that goal! Required {процент стъпки, които трябва да научи всеки един танцьор на ден}% steps to be learned per day."**

И двата отговора трябва да са форматирани **до втория** знак след десетичната запетая.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **коментари** |
| 10464  20  20 | Yes, they will succeed in that goal! 0.25%. | Общ процент стъпки на ден: (10464 / 20) / 10464 = 0.05 = 5%  Процент стъпки за всеки танцьор: 5 / 20 = 0.25%  5% < 13% -> успяват да научат танца |
| 55555  30  7 | No, they will not succeed in that goal! Required 0.50% steps to be learned per day. | Общ процент стъпки на ден: (55555 / 7) / 55555 = 0.1428.... = = 14.28...% = 15%  Процент стъпки за всеки танцьор: 15 / 30 = 0.50%  15% > 13% -> не успяват да научат танца |

## Световен рекорд по плуване

Иван решава да подобри Световния рекорд по плуване на дълги разстояния. **На конзолата се въвежда рекордът в секунди, който Иван трябва да подобри, разстоянието в метри, което трябва да преплува и времето в секунди, за което плува разстояние от 1 м.** Да се напише програма, която изчислява дали се е справил със задачата, като се има предвид, че: **съпротивлението на водата го забавя на всеки 15 м. с 12.5 секунди.** Когато се изчислява колко пъти Иванчо ще се забави, в резултат на съпротивлението на водата, **резултатът трябва да се закръгли надолу до най-близкото цяло число.**

**Да се изчисли времето в секунди, за което Иванчо ще преплува разстоянието и разликата спрямо Световния рекорд.**

### Вход

От конзолата се четат **3 реда**:

1. **Рекордът в секунди – реално число в интервала [0.00 … 100000.00]**
2. **Разстоянието в метри – реално число в интервала [0.00 … 100000.00]**
3. **Времето в секунди, за което плува разстояние от 1 м.** **- реално число в интервала [0.00 … 1000.00]**

### Изход

Отпечатването на конзолата зависи от резултата:

* Ако **Иван е подобрил Световния рекорд** отпечатваме:
  + **"** **Yes, he succeeded! The new world record is {времето на Иван} seconds."**
* Ако **НЕ е подобрил рекорда** отпечатваме:
  + **"No, he failed! He was {недостигащите секунди} seconds slower."**

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

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 10464  1500  20 | No, he failed! He was 20786.00 seconds slower. | **Иван трябва да преплува 1500 м.: 1500 \* 20 = 30000 сек.**  **На всеки 15 м. към времето му се добавят 12.5 сек.:**  **1500 / 15 = 100 \* 12.5 = 1250 сек.**  **Общо време: 30000 + 1250 = 31250 сек.**  **10464 < 31250**  **Времето, което не му е стигнало за да подобри рекорда:**  **31250 – 10464 = 20786 сек.** |
| **Вход** | **Изход** | **Обяснения** |
| 55555.67  3017  5.03 | Yes, he succeeded! The new world record is 17688.01 seconds. | **Иван трябва да преплува 3017 м.: 3017 \* 5.03 = 15175.51 сек.**  **На всеки 15 м. към времето му се добавят 12.5 сек.:**  **3017/ 15 = 201 \* 12.5 = 2512.50 сек.**  **Общо време: 15175.51 + 2512.50 = 17688.01 сек.**  **Рекордът е подобрен: 55555.67 > 17688.01** |

## \*Стипендии

Учениците могат да кандидатстват за **социална стипендия** или за **стипендия за отличен успех**. Изискване за социална стипендия - **доход на член от семейството по-малък от минималната работна заплата и успех над 4.5**. Размер на социалната стипендия - **35% от минималната работна заплата**. Изискване за стипендия за отличен успех - **успех над 5.5, включително**. Размер на стипендията за отличен успех - **успехът на ученика, умножен по коефициент 25**.

Напишете програма, която при въведени **доход**, **успех** и **минимална** **работна** **заплата**, дава информация дали ученик има право да получава стипендия, и стойността на стипендията, която **е по-висока** за него.

### Вход

Потребителят въвежда **3 числа**, по едно на ред:

1. **Доход в лева** - **реално число в интервала [0.00..6000.00]**
2. Среден успех- **реално число в интервала [2.00...6.00]**
3. Минимална работна заплата - реално **число в интервала [0.00..1000.00]**

### Изход

* Ако ученикът **няма право да получава стипендия**, се извежда:

**"You cannot get a scholarship!"**

* Ако ученикът **има право да получава социална стипендия** и тя **е по-висока** от стипендията за отличен успех:  
  **"You get a Social scholarship {стойност на стипендия} BGN"**
* Ако ученикът **има право да получава стипендия за отличен успех** и тя е **по-висока** или **равна** по стойност на социалната стипендия за него:
* **"You get a scholarship for excellent results {стойност на стипендията} BGN"**

**Резултатът се закръгля до по-малкото цяло число.**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 480.00  4.60  450.00 | You cannot get a scholarship! | Доходът от 480 лв. е по-висок от минималната работна заплата 450 лв. → ученикът не може да получава социална стипендия.  Успех 4.60 < 5.50 → ученикът не може да получава стипендия за отличен успех. |
| 300.00  5.65  420.00 | You get a Social scholarship 147 BGN | 300 лв. < 420 лв. и 5.65 >4.50 → ученикът може да получава социална стипендия 35% \* 420 лв. = 147 лв.  Успех 5.65 > 5.50 → ученикът може да получава стипендия за отличен успех 5.65 \* 25 = 141.25 лв.  147 лв. > 141.25 лв. → ученикът ще получава социална стипендия. |