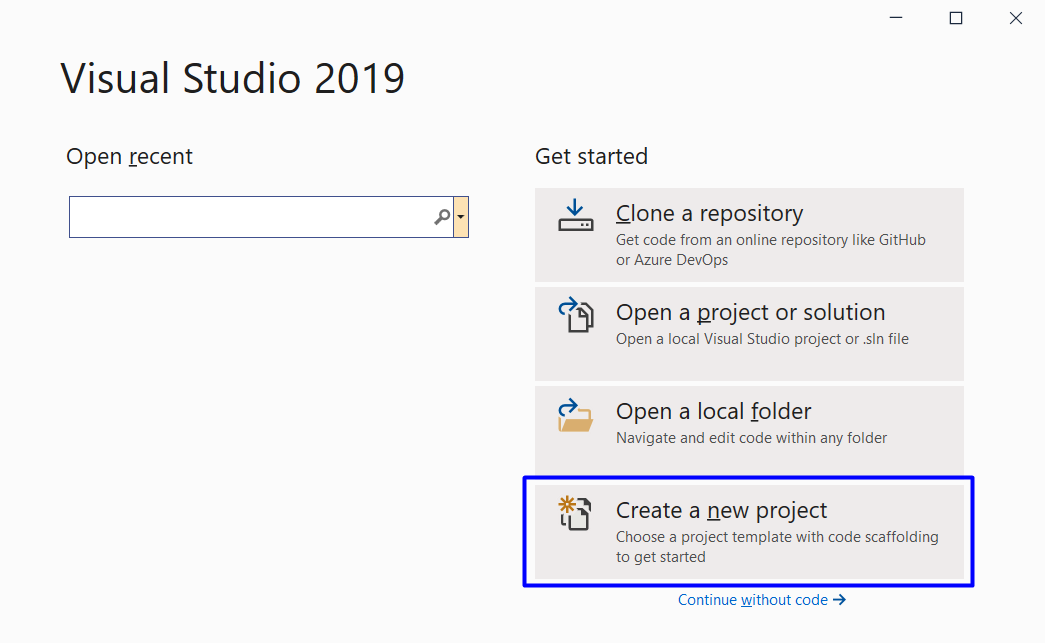
# Упражнения: Условни конструкции

## Празно Visual Studio решение (Blank Project)

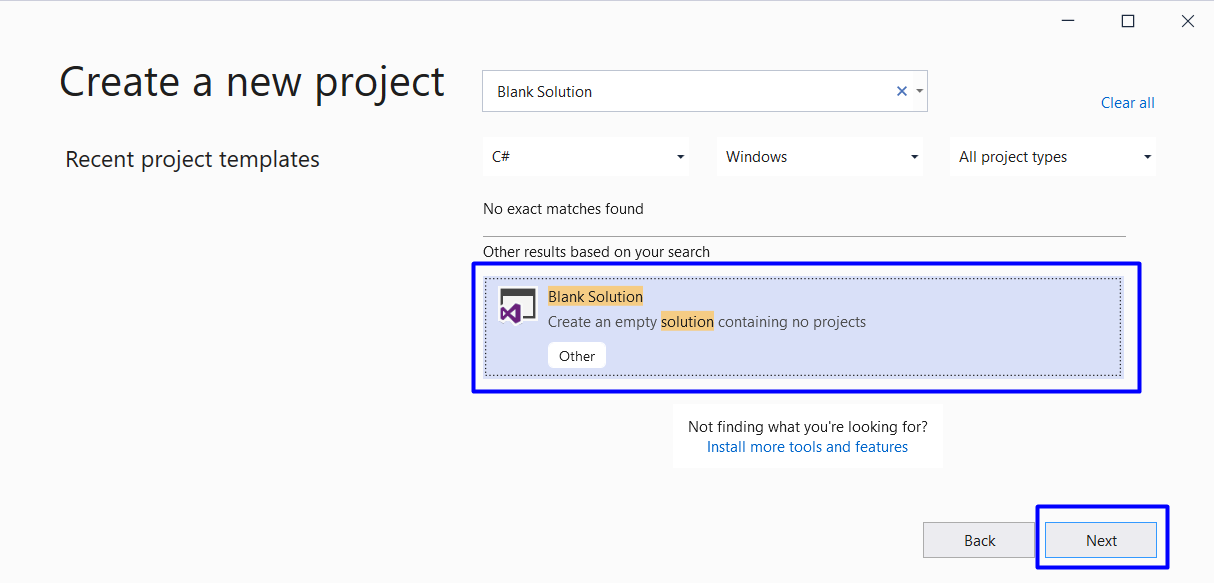
Създайте празно решение (**Blank Solution**) във Visual Studio. Решенията (solutions) във Visual Studio обединяват **група проекти**. Тази възможност е изключително удобна, когато искаме да работим по няколко проекта и бързо да превключваме между тях или искаме да обединим логически няколко взаимосвързани проекта.

В настоящото практическо занимание ще използваме **Blank Solution с няколко проекта,** за да организираме решенията на задачите от упражненията – всяка задача в отделен проект и всички проекти в общ solution.

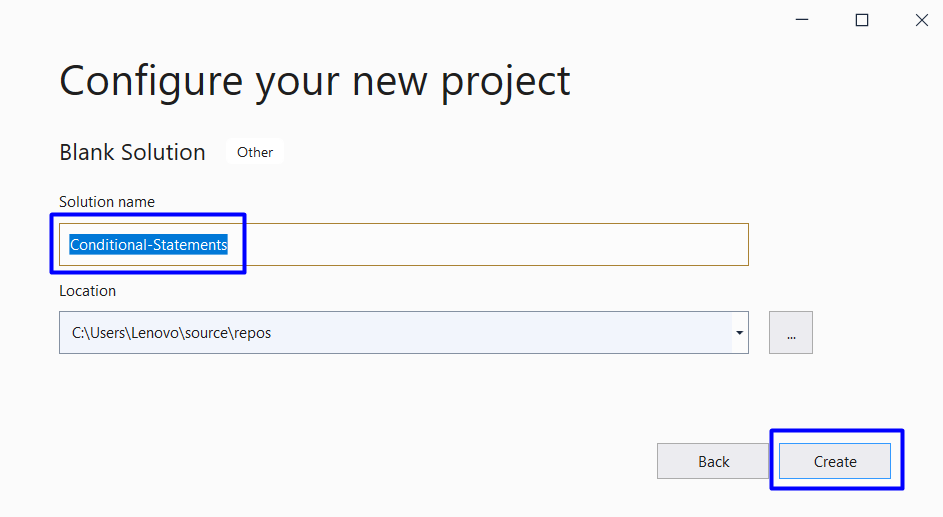
1. Стартирайте Visual Studio.
2. Създайте нов проект: [Create a new project].



1. Изберете [Black solution], ако не го виждате, в търсачката изпишете ["Blank solution"].



1. Задайте подходящо име на проекта, например "Conditional-Statements":



Сега имате създаден **празен Visual Studio Solution** (с 0 проекта в него):

A screenshot of a social media post

Description automatically generated

Целта на този blank solution e да добавяте в него **по един проект за всяка задача** от упражненията.

## Проверка за отлична оценка

Първата задача от тази тема е да се напише **конзолна програма**, която **чете оценка** (десетично число), въведена от потребителя и отпечатва "**Excellent!",** ако оценката е **5.50** или по-висока.

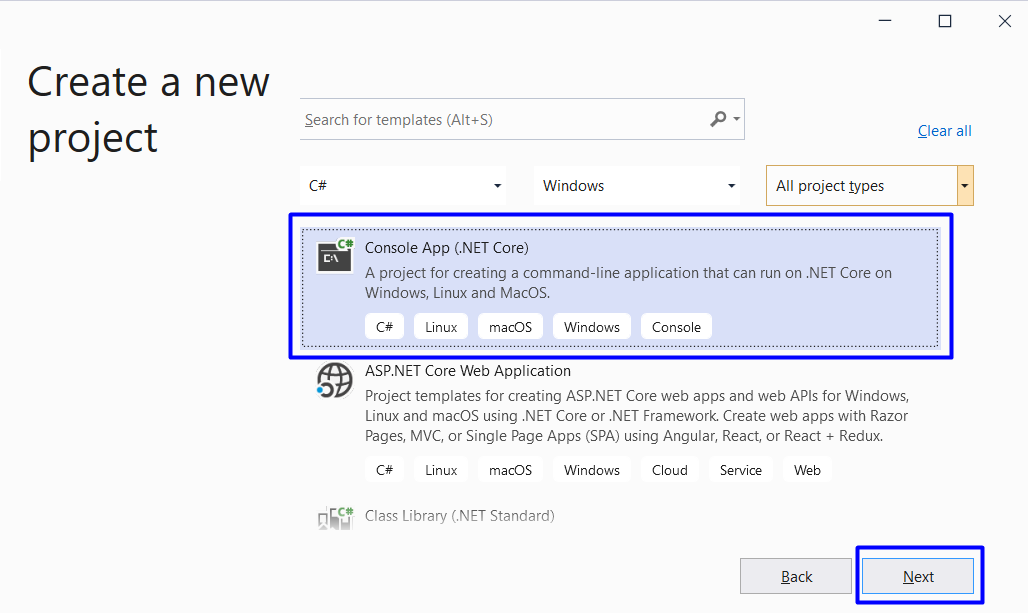
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 6 | Excellent! | 5 | *(няма изход)* | 5.50 | Excellent! | 5.49 | *(няма изход)* |

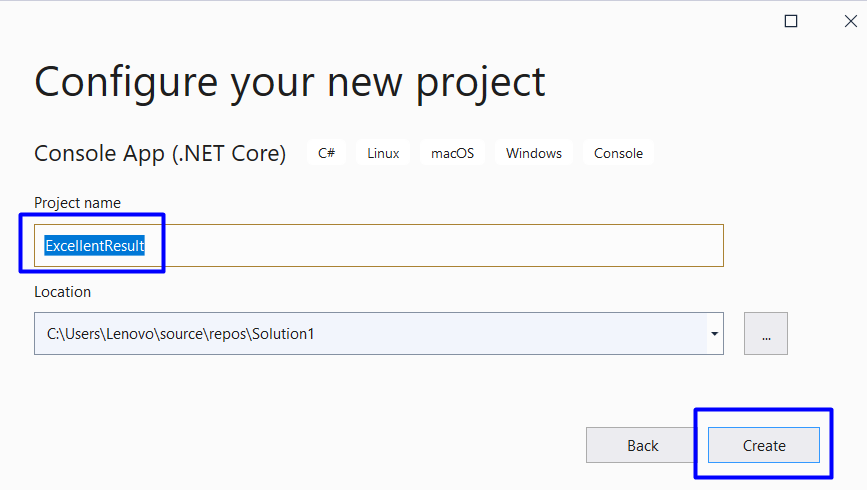
### Насоки

1. Създайте **нов C# конзолен проект** с име "ExcellentResult" в **Blanc Project** с име " Conditional-Statements ", като натиснем с десен бутон на мишката в/у **Solution Condition-Statements->**  **add -> New Project...**:

A screenshot of a social media post

Description automatically generated





1. Вече имате създаден клас със Main **метод**

A screenshot of a cell phone

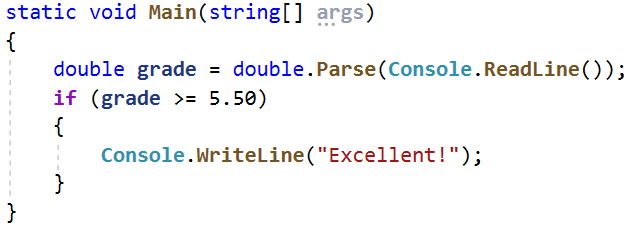
Description automatically generated

1. Отидете в тялото на метода Main(string[] args) (между къдравите скоби). Създайте една променлива, в която да запазите **реално** **число** – оценката, което ще прочетете от конзолата:

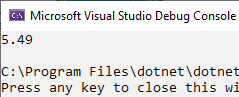
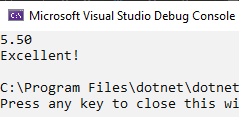
A screenshot of a cell phone

Description automatically generated

1. Направете проверка за стойността на оценката. Ако тя е по-голяма или равна на 5.50 отпечатайте изхода по условие:



1. Стартирайте програмата с **Ctrl + F5** и я **тествайте** с различни входни стойности:



## Намиране на по-голямото число

Да се напише програма, която чете **две цели числа** въведени от потребителя и отпечатва **по-голямото от двете**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 5  3 | 5 | 3  5 | 5 | 10  10 | 10 | -5  5 | 5 |

### Насоки

1. Прочетете две цели числа от конзолата:

A screenshot of a cell phone

Description automatically generated

1. Сравнете, дали първото число num1 e по-голямо от второто num2. Отпечатайте по-голямото число.

A screenshot of a cell phone

Description automatically generated

## Четно или нечетно

Да се напише програма, която чете **цяло число** въведено от потребителя и отпечатва на конзолата, дали е **четно** или **нечетно**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 2 | even | 3 | odd | 25 | odd | 1024 | even |

### Насоки

1. Първо добавете **нов конзолен проект** към съществуващия проект
2. Прочетете едно цяло число от конзолата:

A screenshot of a cell phone

Description automatically generated

1. Проверете, дали числото е четно, като използвате модул оператора с 2 и проверите, дали има остатък от целочисленото деление. Отпечатайте изхода по условие – текста "even":

A screenshot of a cell phone

Description automatically generated

1. В противен случай отпечатайте "odd":

A screenshot of a cell phone

Description automatically generated

## Число от 100 до 200

Да се напише програма, която **чете цяло число**, въведено от потребителя и проверява, дали е **под 100**, **между 100 и 200** или **над 200**. Да се отпечатат съответно съобщения, като в примерите по-долу:

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 95 | Less than 100 | 120 | Between 100 and 200 | 210 | Greater than 200 |

## Познай паролата

Да се напише програма, която **чете парола** (един ред с произволен текст), въведена от потребителя и проверява, дали въведеното **съвпада** с фразата "s3cr3t!P@ssw0rd".При съвпадение да се изведе "**Welcome**". При несъвпадение да се изведе "**Wrong password!**".

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| qwerty | Wrong password! | s3cr3t!P@ssw0rd | Welcome | s3cr3t!p@ss | Wrong password! |

## Лица на фигури

Да се напише програма, в която потребителят **въвежда вида и размерите на геометрична** фигура и пресмята лицето й. Фигурите са четири вида: квадрат (**square**), правоъгълник (**rectangle**), кръг (**circle**) и триъгълник (**triangle**). На първия ред на входа се чете вида на фигурата (square, rectangle, circle или triangle).

* Ако фигурата е **квадрат**, на следващия ред се чете едно число - дължина на страната му.
* Ако фигурата е **правоъгълник**, на следващите два реда четат две числа - дължините на страните му.
* Ако фигурата е **кръг**, на следващия ред чете едно число - радиусът на кръга.
* Ако фигурата е **триъгълник**, на следващите два реда четат две числа - дължината на страната му и дължината на височината към нея.

Резултатът да се закръгли до **3 цифри след десетичната точка**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| square  5 | 25.000 | rectangle  7  2.5 | 17.500 | circle  6 | 113.097 | triangle  4.5  20 | 45.000 |

## Ден от седмицата

Напишете програма, която чете **цяло** **число**, въведено от потребителя, и отпечатва **ден от седмицата** (на английски език), в граници [1...7] или отпечатва "Error" в случай, че въведеното число е **невалидно**.

### Примерен вход и изход

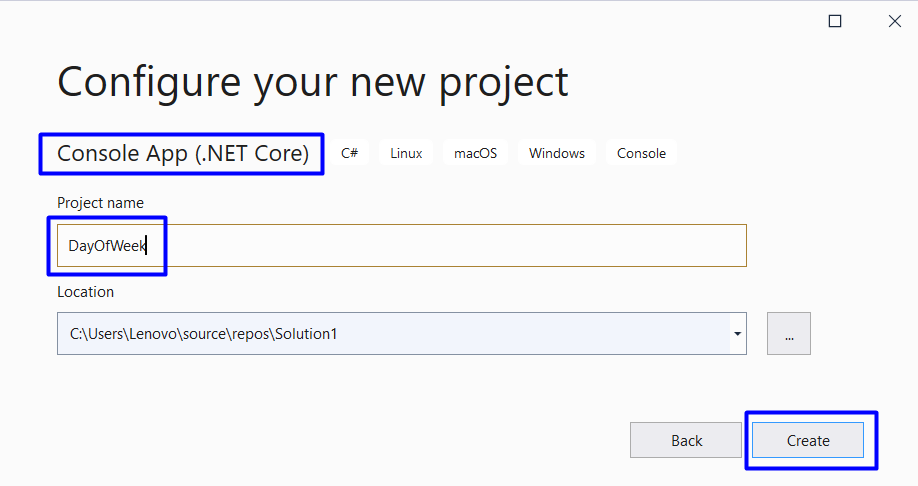
|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 | Monday |
| 2 | Tuesday |
| 3 | Wednesday |
| 4 | Thursday |
| 5 | Friday |
| 6 | Saturday |
| 7 | Sunday |
| -1 | Error |

### Насоки

1. Създайте **нов проект** в съществуващото Visual Studio решение. В Solution Explorer кликнете с десен бутон на мишката върху **Solution** реда и изберете [Add] 🡪 [New Project…]:



1. Ще се отвори диалогов прозорец за избор на тип проект за създаване. Изберете **C#** **конзолно приложение** и задайте подходящо име, например "DayOfWeek":

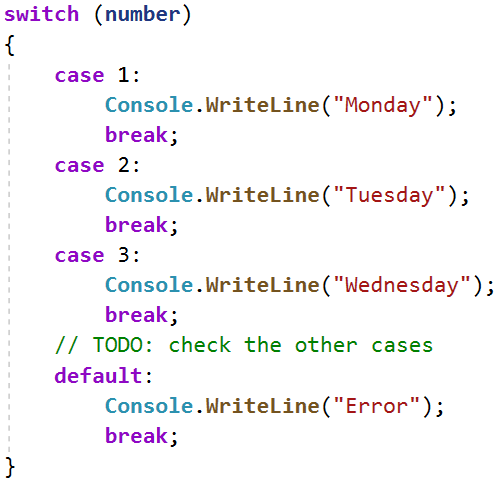


Вече имате solution с едно конзолно приложение в него. Остава да напишете кода за решаване на задачата.

1. Прочетете едно цяло число от конзолата:



1. Отпечатайте денят от седмицата според въведеното число. Ако то е невалидно, отпечатайте "**Error**".



## Почивен или работен ден

Напишете програма която, чете ден от седмицата (**текст**), на английски език - въведен от потребителя. Ако денят е работен отпечатва на конзолата - "**Working day**", ако е почивен - "**Weekend**". Ако се въведе текст различен от ден от седмицата да се отпечата - "**Error**".

### Примерен вход и изход

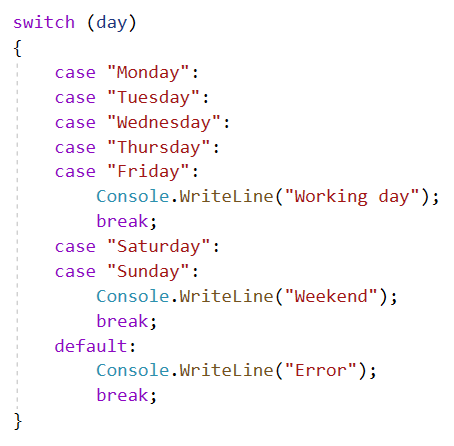
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| Monday | Working day |  | Sunday | Weekend |  | April | Error |

### Насоки

1. Прочетете ден от седмицата(**текст**) от конзолата:



1. Отпечатайте работен или почивен ден, според въведения ден, ако денят е невалиден отпечатайте "**Error**":



## Клас животно

Напишете програма, която отпечатва класа на животното според неговото име, въведено от потребителя.

1. **dog -> mammal**
2. **crocodile, tortoise, snake -> reptile**
3. **others -> unknown**

### Примерен вход и изход

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| dog | mammal |
| snake | reptile |
| cat | unknown |

### Насоки

1. Прочетете входните данни:



1. Проверете от какъв вид е животното. Ако то е невалидно, отпечатайте "**unknown**".



## Обръщение според възраст и пол

Да се напише **конзолна програма**, която **прочита възраст** (реално число) и **пол** ('m' или 'f'), въведени от потребителя, и отпечатва **обръщение** измежду следните:

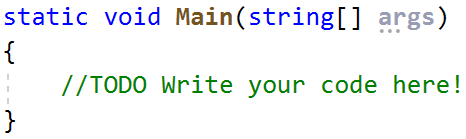
* "Mr." – мъж (пол 'm') на 16 или повече години
* "Master" – момче (пол 'm') под 16 години
* "**Ms.**" – жена (пол 'f') на 16 или повече години
* "**Miss**" – момиче (пол 'f') под 16 години

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 12  f | Miss | 17  m | Mr. | 25  f | Ms. | 13.5  m | Master |

### Насоки

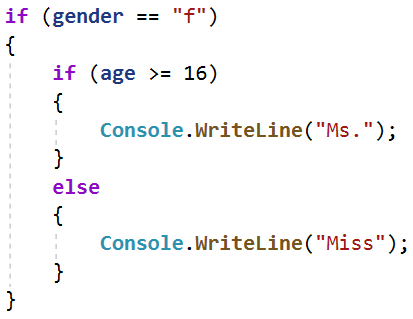
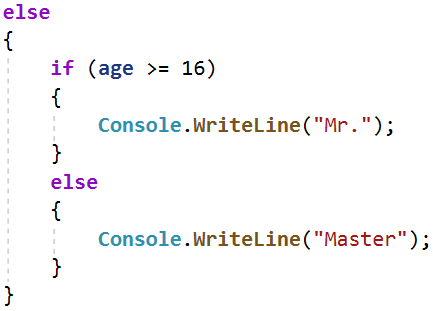
1. Отидете в тялото на метода Main(string[] args) и напишете решението на задачата. Можете да си помогнете с кода от картинките по-долу:



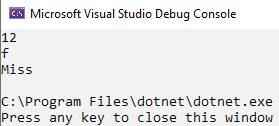
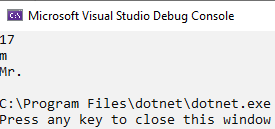
1. Прочетете **реално число** от конзолата "**age**", на следващия ред прочетете **string / текст** за "**gender**".



1. Направете проверка за пола, и ако върне резултат true**,** направете проверка за годините. В тялото на проверките за възраст принтирайте желаното обръщение.

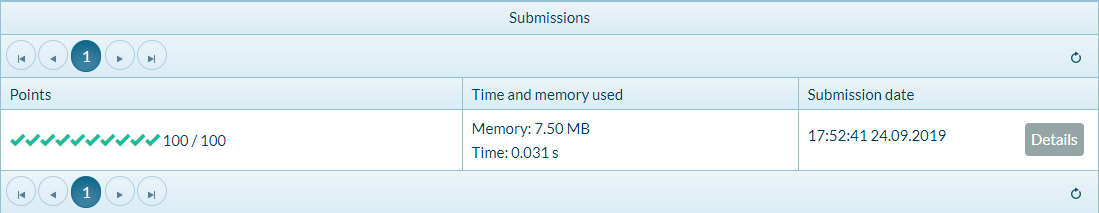
 

1. **Стартирайте** програмата с [Ctrl+F5] и я **тествайте** с различни входни стойности:

Трябва да получите **100 точки** (напълно коректно решение):





## Квартално магазинче

Предприемчив българин отваря **квартални магазинчета** в **няколко града** и продава на **различни цени според града**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| град / продукт | **coffee** | **water** | **juice** | **sweets** | **peanuts** |
| **Sofia** | 0.50 | 0.80 | 1.20 | 1.45 | 1.60 |
| **Plovdiv** | 0.40 | 0.70 | 1.15 | 1.30 | 1.50 |
| **Varna** | 0.45 | 0.70 | 1.10 | 1.35 | 1.55 |

Напишете програма, която чете **продукт** (низ), **град** (низ) и **количество** (десетично число), въведени от потребителя, и пресмята и отпечатва **колко струва** съответното количество от избрания продукт в посочения град.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| coffee  Varna  2 | 0.9 | peanuts  Plovdiv  1 | 1.5 | juice  Sofia  6 | 7.2 | water  Plovdiv  3 | 2.1 | sweets  Sofia  2.23 | 3.2335 |

## Число в интервала

Да се напише програма, която проверява дали въведеното от потребителя число е в интервала [-100, 100] и е различно от 0 и извежда "**Yes**", ако отговаря на условията, или "**No**" ако е извън тях.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| -25 | Yes | 0 | No | 25 | Yes |

## Работно време

Да се напише програма, която чете час от денонощието(**цяло число**) и ден от седмицата(**текст**) - въведени от потребителя и проверява дали офисът на фирма е отворен, като работното време на офисът е от **10-18** часа, от **понеделник** до **събота** включително

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 11  Monday | open | 19  Friday | closed | 11  Sunday | closed |

## Билет за кино

Да се напише програма която чете ден от седмицата (текст) – въведен от потребителя и принтира на конзолата цената на билет за кино според деня от седмицата:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** | **Sunday** |
| 12 | 12 | 14 | 14 | 12 | 16 | 16 |

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| Monday | 12 | Friday | 12 | Sunday | 16 |

## Плод или зеленчук

Да се напише програма, която **чете име на продукт**, въведено от потребителя, и проверява дали е **плод** или **зеленчук**.

* Плодовете "fruit" имат следните възможни стойности: **banana**, **apple**, **kiwi**, **cherry**, **lemon** и **grapes**
* Зеленчуците "vegetable" имат следните възможни стойности: **tomato**, **cucumber**, **pepper** и **carrot**
* Всички останали са "unknown"

Да се изведе "fruit”, "vegetable" или "unknown" според въведения продукт.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| banana | fruit | apple | fruit | tomato | vegetable | water | unknown |

## Невалидно число

Дадено **число е валидно**, ако е в диапазона [**100**…**200**] или е **0**. Да се напише програма, която **чете цяло число**, въведено от потребителя, и печата "invalid" ако въведеното число **не е валидно**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 75 | invalid | 150 | *(няма изход)* | 220 | invalid | 199 | *(няма изход)* |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| -1 | invalid | 100 | *(няма изход)* | 200 | *(няма изход)* | 0 | *(няма изход)* |

## Магазин за плодове

Магазин за плодове през **работните дни** работи на следните **цени**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **плод** | **banana** | **apple** | **orange** | **grapefruit** | **kiwi** | **pineapple** | **grapes** |
| **цена** | 2.50 | 1.20 | 0.85 | 1.45 | 2.70 | 5.50 | 3.85 |

**Събота** и **неделя** магазинът работи на **по-високи** **цени**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **плод** | **banana** | **apple** | **orange** | **grapefruit** | **kiwi** | **pineapple** | **grapes** |
| **цена** | 2.70 | 1.25 | 0.90 | 1.60 | 3.00 | 5.60 | 4.20 |

Напишете програма, която чете от конзолата **плод** (banana / apple / orange / grapefruit / kiwi / pineapple / grapes), **ден от седмицата** (Monday / Tuesday / Wednesday / Thursday / Friday / Saturday / Sunday) и **количество** (реално число) , въведени от потребителя, и пресмята **цената** според цените от таблиците по-горе. Резултатът да се отпечата **закръглен с 2 цифри** след десетичната точка. При невалиден ден от седмицата или невалидно име на плод да се отпечата "error".

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| apple  Tuesday  2 | 2.40 | orange  Sunday  3 | 2.70 | kiwi  Monday  2.5 | 6.75 | grapes  Saturday  0.5 | 2.10 | tomato  Monday  0.5 | error |

## Търговски комисионни

Фирма дава следните **комисионни** на търговците си според **града**, в който работят и обема на **продажбите**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Град** | **0 ≤ s ≤ 500** | **500 < s ≤ 1 000** | **1 000 < s ≤ 10 000** | **s > 10 000** |
| Sofia | 5% | 7% | 8% | 12% |
| Varna | 4.5% | 7.5% | 10% | 13% |
| Plovdiv | 5.5% | 8% | 12% | 14.5% |

Напишете **конзолна програма**, която чете име на **град** (стринг) и обем на **продажби** (реално число) , въведени от потребителя, и изчислява и извежда размера на търговската **комисионна** според горната таблица. Резултатът да се изведе форматиран до **2 цифри след десетичната точка**. При **невалиден** град или обем на продажбите (отрицателно число) да се отпечата "error".

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| Sofia  1500 | 120.00 | Plovdiv  499.99 | 27.50 | Varna  3874.50 | 387.45 | Kaspichan  -50 | error |

## Магазин за детски играчки

Петя има магазин за детски играчки. Тя получава голяма поръчка, която трябва да изпълни. С парите, които ще спечели иска да отиде на екскурзия. Да се напише програма, която пресмята печалбата от поръчката.

**Цени на играчките:**

* **Пъзел - 2.60 лв.**
* **Говореща кукла - 3 лв.**
* **Плюшено мече - 4.10 лв.**
* **Миньон - 8.20 лв.**
* **Камионче - 2 лв.**

Ако поръчаните играчки са **50 или повече** магазинът прави **отстъпка 25%** **от общата цена**. От спечелените пари Петя трябва да даде **10% за наема** на магазина. Да се пресметне дали парите ще ѝ стигнат да отиде на екскурзия.

От конзолата се четат **6 реда**:

1. **Цена на екскурзията - реално число в интервала [1.00 … 10000.00]**
2. **Брой пъзели - цяло число в интервала [0… 1000]**
3. **Брой говорещи кукли - цяло число в интервала [0 … 1000]**
4. **Брой плюшени мечета - цяло число в интервала [0 … 1000]**
5. **Брой миньони - цяло число в интервала [0 … 1000]**
6. **Брой камиончета - цяло число в интервала [0 … 1000]**

На конзолата се отпечатва:

* Ако **парите са достатъчни** се отпечатва:
  + **"Yes! {оставащите пари} lv left."**
* Ако **парите НЕ са достатъчни** се отпечатва:
  + **"Not enough money! {недостигащите пари} lv needed."**

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

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 40.8  20  25  30  50  10 | Yes! 418.20 lv left. | **Сума**: 20 \* 2.60 + 25 \* 3 + 30 \* 4.10 + 50 \* 8.20 + 10 \* 2 = **680** лв.  **Брой на играчките**: 20 + 25 + 30 + 50 + 10 = **135**  **135 > 50 => 25% отстъпка**; 25% от 680 = **170 лв. отстъпка**  **Крайна цена**: 680 – 170 = **510** лв.  **Наем**: 10% от 510 лв. = **51** лв.  **Печалба**: 510 – 51 = **459** лв.  **459 > 40.8** =>459 – 40.8= **418.20** лв. **остават** |
| **Вход** | **Изход** | **Обяснения** |
| 320  8  2  5  5  1 | Not enough money! 238.73 lv needed. | **Сума**: **90.3** лв.  **Брой на играчките**: **21**  **21 < 50 => няма отстъпка**  **Наем**: 10% от 90.3 = **9.03** лв.  **Печалба**: 90.3 – 9.03 = **81.27** лв.  **81.27 < 320** => 320 – 81.27= **238.73** лв. **не достигат** |

## \*Ски почивка

Атанас решава да прекара отпуската си в Банско и да кара ски. Преди да отиде обаче, трябва да резервира хотел и да изчисли **колко ще му струва престоя**. Съществуват следните видове помещения, със следните цени за престой:

* + - **"room for one person" – 18.00 лв. за нощувка**
    - "**apartment**" – **25.00 лв. за нощувка**
    - **"president** **apartment**" – **35.00 лв. за нощувка**

Според **броят на дните**, в които ще остане в хотела (**пример: 11 дни = 10 нощувки**) и **видът на помещението**, което ще избере, той може да ползва различно **намаление**. Намаленията са както следва:

|  |  |  |  |
| --- | --- | --- | --- |
| **вид помещение** | **по-малко от 10 дни** | **между 10 и 15 дни** | **повече от 15 дни** |
| **room for one person** | не ползва намаление | не ползва намаление | не ползва намаление |
| **apartment** | **30%** от крайната цена | **35%** от крайната цена | **50%** от крайната цена |
| **president apartment** | **10%** от крайната цена | **15%** от крайната цена | **20%** от крайната цена |

След престоя, оценката на Атанас за услугите на хотела може да е **позитивна (positive)** или **негативна (negative)** . Ако оценката му е **позитивна**, към цената **с вече приспаднатото намаление** Атанас добавя **25%** от нея. Ако оценката му е **негативна** приспада от цената **10%**.

### Вход

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

* **Първи ред** - **дни за престой** - **цяло число** в интервала **[0...365]**
* **Втори ред** - **вид помещение -** "room for one person", "apartment" или "**president** apartment"
* **Трети ред** - **оценка** - "positive" или "negative"

### Изход

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

* **Цената за престоят му в хотела, форматирана до втория знак след десетичната запетая.**

### Примерен вход и изход

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | | | |
| 14  apartment  positive | 264.06 | **14 дни => 13 нощувки** => 13 \* 25.00 = 325 лв.  **10 < 13 дни < 15** => 325 – 35%= 211.25 лв.  **Оценката е positive** => 211.25 + 25% = 264.0625 -> 264.06 лв. | | | |
| **Вход** | **Изход** | **Вход** | **Изход** | **Вход** | **Изход** |
| 30  president apartment  negative | 730.80 | 12  room for one person  positive | 247.50 | 2  apartment  positive | 21.88 |

## 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, а след делението с остатък (%) ще получим оставащите секунди(14).**



1. След като вече знаете **колко минути и секунди** е общия сбор, трябва да ги принтираме в правилния формат **(минути : секунди**), като ако секундите са **по-малки от 10** трябва да отпечатаме **0 преди секундите**, в противен случай просто си отпечатваме **резултата в дадения формат**. За целта **направете проверка** (if). За принтирането може да използвате **шаблон**.



## 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.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 0.0**.



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**.
   * aко числото **е четно към до момента натрупания бонус добавете 1**
   * 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**.

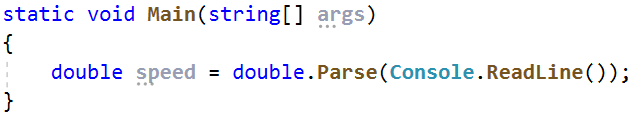


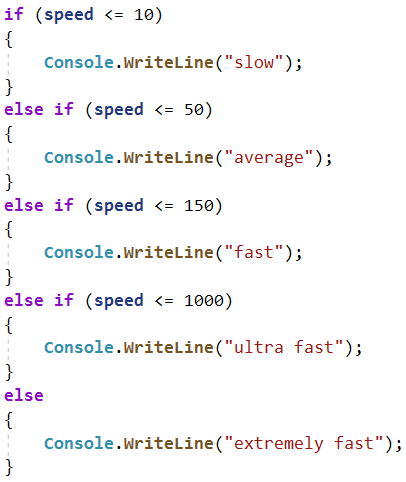
## Информация за скоростта

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 |

## 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.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

## Годзила срещу Конг

Снимките за дългоочаквания филм "Годзила срещу Конг" започват. Сценаристът Адам Уингард ви моли да **напишете програма**, която да изчисли, **дали предвидените средства са достатъчни** за снимането на филма. За снимките ще бъдат нужни **определен брой статисти, облекло** за всеки един статист и **декор.**

Известно е, че:

* + Декорът за филма е **на стойност 10% от бюджета.**
  + При **повече от 150 статиста, има отстъпка за облеклото на стойност 10%.**

### Вход

От конзолата се четат **3 реда**:

1. **Бюджет за филма – реално число в интервала [1.00 … 1000000.00]**
2. **Брой на статистите – цяло число в интервала [1 … 500]**
3. **Цена за облекло на един статист – реално число в интервала [1.00 … 1000.00]**

### Изход

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

* Ако парите за декора и дрехите **са повече от бюджета**:
  + "Not enough money!"
  + "Wingard needs {парите недостигащи за филма} leva more."
* Ако парите за декора и дрехите са **по малко или равни на бюджета**:
  + "Action!"
  + "Wingard starts filming with {останалите пари} leva left."

Резултатът трябва да е форматиран до втория знак след десетичната запетая.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 20000  120  55.5 | Action!  Wingard starts filming with 11340.00 leva left. | Сума за декор: 10% от 20000 = 2000 лв.  Сума за облекло: 120 \* 55.5 = 6660 лв.  Обща сума за филма: 2000 + 6660 = 8660 лв.  20000 – 8660 = 11340 лева остават. |
| 15437.62  186  57.99 | Action!  Wingard starts filming with 4186.33 leva left. | Сума за декор: 10% от 15437.62 = 1543.762 лв.  Сума за облекло: 186 \* 57.99 = 10786.14 лв.  Статистите са повече от 150 следователно има 10% отстъпка на облеклото.  10% от 10786.14 е 1078.614  10786.14 – 1078.614 = 9707.526 лв. за облекло  Обща сума за филма: 1543.762 + 9707.526 = 11251.288  15437.62 – 11251.288 = 4186.331 лева остават |
| 9587.88  222  55.68 | Not enough money!  Wingard needs 2495.77 leva more. | Сума за декор: 10% от 9587.88 = 958.788 лв.  Сума за облекло: 11124.864 лв.  Обща сума за филма: 958.788 + 11124.864 = 12083.652  9587.88 – 12083.652 = 2495.77 лева не достигат |

## 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** **(времето му е по-малко от рекорда)** отпечатваме:
  + **"** **Yes, he succeeded! The new world record is {времето на Иван} seconds."**
* Ако **НЕ е подобрил рекорда (времето му е по-голямо или равно на 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 **само** социална стипендия:  
  **"You get a Social scholarship {стойност на стипендия} BGN"**
* Ако ученикът **има право да получава само стипендия за отличен успех**:

**"You get a scholarship for excellent results {стойност на стипендията} BGN"**

* **Ако ученикът има право да получава и двата типа стипендии, ще получи по-голямата по сума, а ако са равни ще получи тази за отличен успех.**

**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. |

## Кино

В една кинозала столовете са наредени в правоъгълна форма в r реда и c колони. Има три вида прожекции с билети на различни цени:

* Premiere – премиерна прожекция, на цена 12.00 лева.
* Normal – стандартна прожекция, на цена 7.50 лева.
* Discount – прожекция за деца, ученици и студенти на намалена цена от 5.00 лева.

Напишете програма, която чете **тип прожекция** (стринг), брой **редове** и брой **колони** в залата (цели числа), въведени от потребителя, и изчислява общите приходи от билети при пълна зала. Резултатът да се отпечата във формат като в примерите по-долу, с 2 знака след десетичната точка.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| Premiere  10  12 | 1440.00 leva | Normal  21  13 | 2047.50 leva | Discount  12  30 | 1800.00 leva |

\* **Подсказка**: използвайте прости проверки и елементарни изчисления. За да изведете резултата с точно 2 цифри след десетичната точка, използвайте Console.WriteLine("{0:f2} leva", income).

### Насоки

1. Прочетете входните данни от конзолата.



1. Инициализирайте променлива "income" от тип "double" с начална стойност 0.



1. Направете серия от проверки, като за всеки тип прожекция ("Premiere, Normal, Discount"), присвоявайте съответната цена към променливата "income" и накрая отпечатайте крайния резултат.



## Лятно облекло

Лято е с много променливо време и Виктор има нужда от вашата помощ. Напишете програма която **спрямо времето от денонощието** **и градусите** да препоръча на Виктор какви дрехи да си облече. Вашия приятел има различни планове за всеки етап от деня, които изискват и различен външен вид, тях може да видите от **таблицата.**

**От конзолата се четат точно два реда:**

* **Градусите - цяло число в интервала [10…42]**
* **Текст, време от денонощието - с възможности - "Morning", "Afternoon", "Evening"**

|  |  |  |  |
| --- | --- | --- | --- |
| **Време от денонощието / градуси** | Morning | Afternoon | Evening |
| 10 <= градуси <= 18 | Outfit = Sweatshirt  Shoes = Sneakers | Outfit = Shirt  Shoes = Moccasins | Outfit = Shirt  Shoes = Moccasins |
| 18 < градуси <= 24 | Outfit = Shirt  Shoes = Moccasins | Outfit = T-Shirt  Shoes = Sandals | Outfit = Shirt  Shoes = Moccasins |
| градуси >= 25 | Outfit = T-Shirt  Shoes = Sandals | Outfit = Swim Suit  Shoes = Barefoot | Outfit = Shirt  Shoes = Moccasins |

**Да се отпечата на конзолата на един ред:** **"It's {градуси} degrees, get your {облекло} and {обувки}."**

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| 16  Morning | It's 16 degrees, get your Sweatshirt and Sneakers. | Сутрин когато градусите са 16, Виктор си взима суитчър и маратонки. | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 22  Afternoon | It's 22 degrees, get your T-Shirt and Sandals. | 28  Evening | It's 28 degrees, get your Shirt and Moccasins. |

### Насоки

1. Прочетете входните данни от конзолата и инициализирайте две променливи "outfit, shoes" от тип "string", с начална стойност "".



1. Направете проверка за градусите използвайки логически оператор "и" – "&&" - "degrees >= 10 && degrees <= 18" и в тялото на проверката за градусите проверявайте за времето от деня "Morning, Afternoon, Evening", като за всеки етап от деня променяйте стойността на променливите "outfit, shoes".



1. Отпечатайте на конзолата крайния резултат във формата описан в условието на задачата.



## Нов дом

Марин и Нели си купуват къща не далеч от София. Нели толкова много обича цветята, че Ви убеждава да **напишете програма** която да **изчисли колко ще им струва**, да си засадят определен брой цветя и **дали наличния бюджет ще им е достатъчен. Различните цветя са с различни цени.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **цвете** | **Роза** | **Далия** | **Лале** | **Нарцис** | **Гладиола** |
| **Цена на брой в лева** | 5 | 3.80 | 2.80 | 3 | 2.50 |

**Съществуват следните отстъпки:**

* **Ако Нели купи повече от 80 Рози - 10% отстъпка от крайната цена**
* **Ако Нели купи повече от 90 Далии - 15% отстъпка от крайната цена**
* **Ако Нели купи повече от 80 Лалета - 15% отстъпка от крайната цена**
* **Ако Нели купи по-малко от 120 Нарциса - цената се оскъпява с 15%**
* **Ако Нели Купи по-малко от 80 Гладиоли - цената се оскъпява с 20%**

**От конзолата се четат 3 реда:**

* **Вид цветя - текст с възможности - "Roses", "Dahlias", "Tulips", "Narcissus", "Gladiolus"**
* **Брой цветя - цяло число в интервала [10…1000]**
* **Бюджет - цяло число в интервала [50…2500]**

Да се **отпечата** на конзолата **на един ред:**

* **Ако бюджета им е достатъчен - "Hey, you have a great garden with {броя цвета} {вид цветя} and {останалата сума} leva left."**
* **Ако бюджета им е НЕ достатъчен -** **"Not enough money, you need {нужната сума} leva more."**

**Сумата да бъде форматирана до втория знак след десетичната запетая.**

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| Roses  55  250 | Not enough money, you need 25.00 leva more. | Нели иска 55 броя Рози. Цената на една роза е 5лв., следователно за 55 броя Нели ще трябва да плати: 55 \* 5 = 275.  Тя обаче разполага с 250 лв. бюджет.  Понеже 275 > 250 , то не и достигат 25 лв. | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| Tulips  88  260 | Hey, you have a great garden with 88 Tulips and 50.56 leva left. | Narcissus  119  360 | Not enough money, you need 50.55 leva more. |

## Лодка за риболов

Тони и приятели много обичали да ходят за риба, те са толкова запалени по риболова, че решават да отидат на риболов с кораб. Цената за наема на кораба зависи от **сезона и броя рибари.**

**Цената зависи от сезона:**

* **Цената за наем на кораба през пролетта е 3000 лв.**
* **Цената за наем на кораба през лятото и есента е 4200 лв.**
* **Цената за наем на кораба през зимата е 2600 лв.**

**В зависимост от броя си групата ползва отстъпка:**

* **Ако групата е до 6 човека включително – отстъпка от 10%.**
* **Ако групата е от 7 до 11 човека включително – отстъпка от 15%.**
* **Ако групата е от 12 нагоре – отстъпка от 25%.**

Рибарите ползват допълнително **5% отстъпка, ако са четен брой освен ако не е есен - тогава нямат допълнителна отстъпка, която се начислява след като се приспадне отстъпката по горните критерии.**

**Напишете програма**, която да **пресмята** далирибаритеще **съберат достатъчно пари.**

### Вход

**От конзолата се четат точно три реда.**

* **Бюджет на групата – цяло число в интервала [1…8000]**
* **Сезон – текст: "Spring", "Summer", "Autumn", "Winter"**
* **Брой рибари – цяло число в интервала [4…18]**

### Изход

Да се **отпечата** на конзолата **един ред**:

* Ако **бюджетът е достатъчен:**

"**Yes! You have {останалите пари} leva left.**"

* Ако **бюджетът НЕ Е достатъчен**:

"**Not enough money! You need {сумата**, **която** **не** **достига} leva.**"

**Сумите** трябва да са **форматирани с точност до два знака след десетичната запетая**.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| 3000  Summer  11 | Not enough money! You need 570.00 leva. | Лятото риболовния туризъм струва 4200 лв., **11 рибари ползват 15% отстъпка -> 4200 - 15% = 3570** лв., **нечетен брой са и не ползват допълнителна отстъпка,.**  3000 <= 3570, следователно не им достигат 570.00 лв. | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 3600  Autumn  6 | Not enough money! You need 180.00 leva. | 2000  Winter  13 | Yes! You have 50.00 leva left. |

## Пътешествие

Странно, но повечето хора си плануват от рано почивката. Млад програмист разполага с **определен бюджет** и свободно време в даден **сезон**. Напишете програма, която да приема **на входа бюджета и сезона**, а **на изхода** да изкарва, **къде ще почива** програмиста и **колко ще похарчи**.

**Бюджета определя дестинацията, а** **сезона определя колко от бюджета ще изхарчи**. Ако е **лято** ще почива на **къмпинг**, а **зимата в хотел**. Ако е в **Европа**, **независимо от сезона** ще почива в **хотел**. Всеки **къмпинг** или **хотел**, **според дестинацията**, има **собствена цена** която отговаря на даден **процент от бюджета**:

* При **100лв. или по-малко** – някъде в **България**
  + **Лято** – **30%** от бюджета
  + **Зима** – **70%** от бюджета
* При **1000лв. или по малко** – някъде на **Балканите**
  + **Лято** – **40%** от бюджета
  + **Зима** – **80%** от бюджета
* При **повече от 1000лв**. – някъде из **Европа**
  + При пътуване из Европа, независимо от сезона ще похарчи **90% от бюджета**.

### Вход

Входът се чете от конзолата и се състои от **два реда**, въведени от потребителя:

* **Първи ред** – Бюджет, **реално число** в интервала **[10.00...5000.00].**
* **Втори ред** – Един от двата възможни сезона: **„summer”** или **“winter”**

### Изход

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

* **Първи ред** – „**Somewhere in [дестинация]**“ измежду “**Bulgaria**", **"Balkans**” и ”**Europe**”
* **Втори ред** – “{**Вид почивка**} – {**Похарчена сума**}“
  + **Почивката** може да е между „**Camp**” и „**Hotel**”
  + **Сумата** трябва да е **закръглена с точност до вторият знак след запетаята**.

### Примерен вход и изход

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 50  summer | Somewhere in Bulgaria  Camp - 15.00 |
| 75  winter | Somewhere in Bulgaria  Hotel - 52.50 |
| 312  summer | Somewhere in Balkans  Camp - 124.80 |
| 678.53  winter | Somewhere in Balkans  Hotel - 542.82 |
| 1500  summer | Somewhere in Europe  Hotel - 1350.00 |

## Операции между числа

Напишете програма, която чете **две цели числа (N1 и N2)** и **оператор,** с който да се **извърши** дадена **математическа операция** с тях. Възможните операции са: **Събиране(+)**, **Изваждане(-)**, **Умножение(\*)**, **Деление(/)** и **Модулно деление(%).** При **събиране**, **изваждане** и **умножение** на конзолата **трябва да се отпечатат резултата** и дали той е **четен** или **нечетен**. При **обикновеното деление** – **резултат**а. При **модулното деление** – **остатъка**. Трябва да се има предвид, че **делителят може да е равен на 0(нула)**, а **на нула не се дели**. В този случай трябва да се отпечата **специално съобщениe**.

### Вход

От конзолата се прочитат **3 реда**, въведени от потребителя:

* **N1** – **цяло число** в интервала **[0...40 000]**
* **N2** – **цяло число** в интервала **[0...40 000]**
* **Оператор** – **един символ** измежду: „**+**“, „**-**“, „**\***“, „**/**“, „**%**“

### Изход

Да се отпечата на конзолата **един ред**:

* Ако операцията е **деление**:
  + "{N1} / {N2} = {резултат}" – резултатът е **форматиран** до **вторият знак след дес.запетая**
* Ако операцията е **модулно деление**:
  + "{N1} % {N2} = {остатък}"
* В случай на **деление с 0 (нула)**:
  + "Cannot divide {N1} by zero"

### Примерен вход и изход

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** | **Вход** | **Изход** |
| 10  12  + | 10 + 12 = 22 - even | 123  12  / | 123 / 12 = 10.25 | 112  0  / | Cannot divide 112 by zero |
| 10  1  - | 10 – 1 = 9 - odd | 10  3  % | 10 % 3 = 1 | 10  0  % | Cannot divide 10 by zero |
| 7  3  \* | 7 \* 3 = 21 - odd |

## Хотелска стая

Хотел предлага **2 вида стаи**: **студио и апартамент**. Напишете програма, която изчислява **цената за целия престой за студио и апартамент**. **Цените** зависят от **месеца** на престоя:

|  |  |  |
| --- | --- | --- |
| **Май и октомври** | **Юни и септември** | **Юли и август** |
| Студио – **50** лв./нощувка | Студио – **75.20** лв./нощувка | Студио – **76** лв./нощувка |
| Апартамент – **65** лв./нощувка | Апартамент – **68.70** лв./нощувка | Апартамент – **77** лв./нощувка |

Предлагат се и следните **отстъпки**:

* За **студио**, при **повече** от **7** нощувки през **май и октомври** : **5% намаление**.
* За **студио**, при **повече** от **14** нощувки през **май и октомври** : **30% намаление**.
* За **студио**, при **повече** от **14** нощувки през **юни и септември**: **20% намаление**.
* За **апартамент**, при **повече** от **14** нощувки**, без значение от месеца : 10% намаление.**

### Вход

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

* На **първия** ред е **месецът** – **May, June,** **July,** **August,** **September** или **October**
* На **втория** ред е **броят на нощувките** – **цяло число в интервала [0 ... 200]**

### Изход

Да се **отпечатат** на конзолата **2 реда**:

* На **първия ред**: “**Apartment: {цена за целият престой} lv.**”
* На **втория ред**: “**Studio: {цена за целият престой} lv.**“

**Цената за целия престой** **форматирана с точност до два знака след десетичната запетая**.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** | |
| May  15 | Apartment: 877.50 lv.  Studio: 525.00 lv. | **През май**, при повече от **14** нощувки, намаляваме цената на студиото с **30%** (50 – 15 = 35), а на апартамента – с **10%** (65 – 6.5 =58.5).  Целият престой в **апартамент** – **877.50 лв**.  Целият престой в **студио** – **525.00 лв**. | |
| **Вход** | **Изход** | **Вход** | **Изход** |
| June  14 | Apartment: 961.80 lv.  Studio: 1052.80 lv. | August  20 | Apartment: 1386.00 lv.  Studio: 1520.00 lv. |

## Навреме за изпит

Студент трябва да отиде **на** **изпит** **в** **определен час** (например в 9:30 часа). Той идва в изпитната зала в даден **час на пристигане** (например 9:40). Счита се, че студентът е дошъл **навреме**, ако е пристигнал в часа на изпита или до половин час преди това. Ако е пристигнал по-рано повече от 30 минути, той е **подранил**. Ако е дошъл след часа на изпита, той е **закъснял**. Напишете програма, която прочита време на изпит и време на пристигане и отпечатва дали студентът е дошъл **навреме**, дали е **подранил** или е **закъснял** и **с колко часа или минути** е подранил или закъснял.

### Вход

От конзолата се четат **4 цели числа** (по едно на ред), въведени от потребителя:

* Първият ред съдържа **час на изпита** – цяло число от 0 до 23.
* Вторият ред съдържа **минута на изпита** – цяло число от 0 до 59.
* Третият ред съдържа **час на пристигане** – цяло число от 0 до 23.
* Четвъртият ред съдържа **минута на пристигане** – цяло число от 0 до 59.

### Изход

На първият ред отпечатайте:

* “**Late**”, ако студентът пристига по-късно от часа на изпита.
* “**On time**”, ако студентът пристига точно в часа на изпита или до 30 минути по-рано.
* “**Early**”, ако студентът пристига повече от 30 минути преди часа на изпита.

Ако студентът пристига с поне минута разлика от часа на изпита, отпечатайте на следващия ред:

“mm **minutes after the start**” за закъснение под час.

* “hh:mm **hours after the start**” за закъснение от 1 час или повече. Минутите винаги печатайте с 2 цифри, например “1:03”.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 9  30  9  50 | Late  20 minutes after the start | 9  00  10  30 | Late  1:30 hours after the start | 10  00  10  00 | On time |
| 9  00  8  30 | On time  30 minutes before the start |  | 14  00  13  55 | On time  5 minutes before the start | 11  30  10  55 | Early  35 minutes before the start |
| 16  00  15  00 | Early  1:00 hours before the start |  | 11  30  8  12 | Early  3:18 hours before the start | 11  30  12  29 | Late  59 minutes after the start |

## \*Волейбол

Влади е студент, живее в София и си ходи от време на време до родния град. Той е много запален по волейбола, но е зает през работните дни и играе **волейбол** само през **уикендите** и в **празничните дни**. Влади играе **в София** всяка **събота**, когато **не е на работа** и **не си пътува до родния град**, както и в **2/3 от празничните дни**. Той пътува до **родния си град** h **пъти** в годината, където играе волейбол със старите си приятели в **неделя**. Влади **не е на работа 3/4 от уикендите**, в които е в София.Отделно, през **високосните години** Влади играе с **15% повече** волейбол от нормалното. Приемаме, че годината има точно **48 уикенда**, подходящи за волейбол.

Напишете програма, която изчислява **колко пъти Влади е играл волейбол** през годината. **Закръглете резултата** надолу до най-близкото цяло число (например 2.15 🡪 2; 9.95 🡪 9).

Входните данни се въвеждат от потребителя, в следния вид:

* Първият ред съдържа думата "leap" (високосна година) или "normal" (невисокосна).
* Вторият ред съдържа цялото число p – брой празници в годината (които не са събота и неделя).
* Третият ред съдържа цялото число h – брой уикенди, в които Влади си пътува до родния град.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментари** |
| leap  5  2 | 45 | 48 уикенда в годината, разделени по следния начин:   * 46 уикенда в София 🡪 46 \* 3 / 4 🡪 **34.5** съботни игри в София * 2 уикенда в родния си град 🡪 2 недели 🡪 **2** игри в неделя в родния град   5 празника:   * 5 \* 2/3 🡪 **3.333** игри в София в празничен ден   Общо игри през уикенди и празници в София и в родния град: 34.5 + 2 + 3.333 🡪 **39.833**  Годината е високосна:   * Влади играе допълнителни 15% \* 39.833 🡪 **5.975** игри волейбол   Общо игри през цялата година:   * 39.833 + 5.975 = **45.808** игри * Резултатът е **45** (закръгля се надолу) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| normal  3  2 | 38 | leap  2  3 | 43 | normal  11  6 | 44 | leap  0  1 | 41 | normal  6  13 | 43 |