# Лаб: Вложени условни конструкции

**Тествайте** решението си в **judge системата**:

<https://judge.softuni.org/Contests/531/Conditional-Statements-Advanced-Lab>

## Ден от седмицата

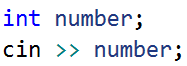
Напишете програма, която чете **цяло** **число**, въведено от потребителя, и отпечатва **ден от седмицата** (на английски език), в граници [1...7] или отпечатва "Error" в случай, че въведеното число е **невалидно**.

### Примерен вход и изход

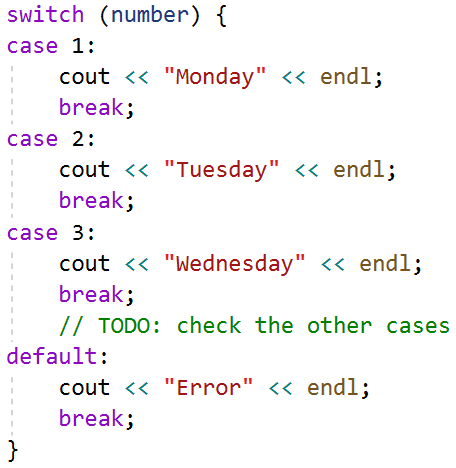
|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 | Monday |
| 2 | Tuesday |
| 3 | Wednesday |
| 4 | Thursday |
| 5 | Friday |
| 6 | Saturday |
| 7 | Sunday |
| -1 | Error |

### Насоки

1. Прочетете едно цяло 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** |
| Monday | Working day |

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Sunday | Weekend |

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| April | 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 |

## 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:

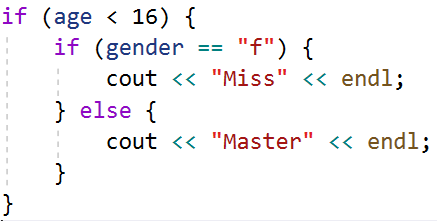
* "Mr." – мъж (пол 'm') на 16 или повече години
* "Master" – момче (пол 'm') под 16 години
* "**Ms.**" – жена (пол 'f') на 16 или повече години
* "**Miss**" – момиче (пол 'f') под 16 години

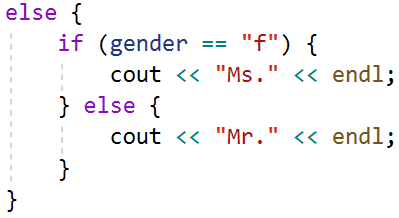
### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 12  f | Miss | 17  m | Mr. | 25  f | Ms. | 13.5  m | Master |

### Насоки

1. Прочетете **реално число** от конзолата "**age**", на следващия ред прочетете **string / текст** за "**gender**".
2. Направете проверка за пола и ако върне резултат true**,** направете проверка за годините. В тялото на проверките за възраст принтирайте желаното обръщение.





## Квартално магазинче

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| град / продукт | **coffee** | **water** | **beer** | **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 |

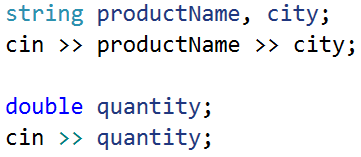
Напишете програма, която чете **продукт** (текст), **град** (текст) и **количество** (десетично число), въведени от потребителя и пресмята и отпечатва **колко струва** съответното количество от избрания продукт в посочения град.

### Примерен вход и изход

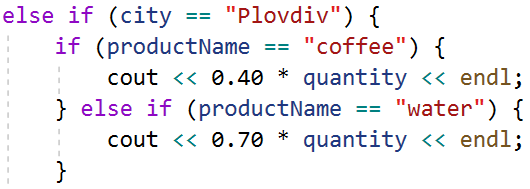
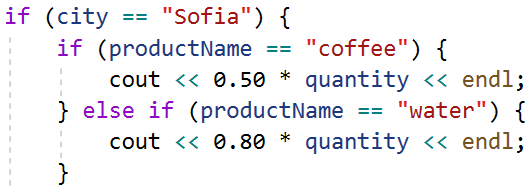
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| coffee  Varna  2 | 0.9 | peanuts  Plovdiv  1 | 1.5 | beer  Sofia  3 | 3.6 | water  Plovdiv  2 | 1.4 | sweets  Sofia  2.23 | 3.2335 |

### Насоки

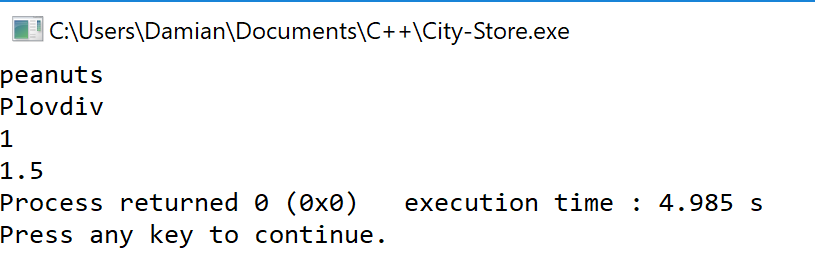
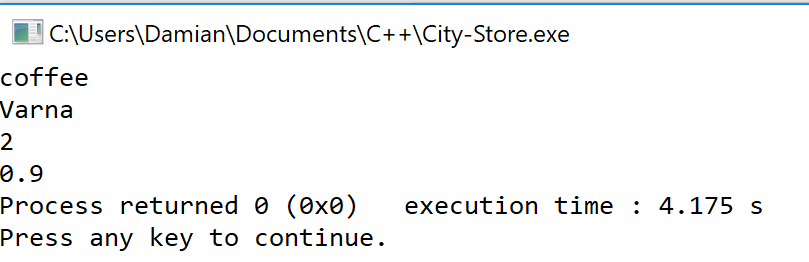
1. Отидете в тялото на метода **m**ain() и напишете решението на задачата, като първо прочетете входните данни от конзолата, създайте си и една променлива price, като й зададете стойност 0.



1. Направете серия от проверки, като за всеки град проверете за дадения продукт. Във всяка проверка за продукт, принтирайте резултата. Вижте примера по-долу:



1. Тествайте решението си с различни входни стойности:



## Число в интервалa

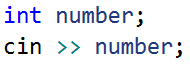
Да се напише програма, която проверява дали въведеното от потребителя число е в интервала [-100, 100] и е различно от 0 и извежда "**Yes**", ако отговаря на условията, или "**No**" ако е извън тях.

### Примерен вход и изход

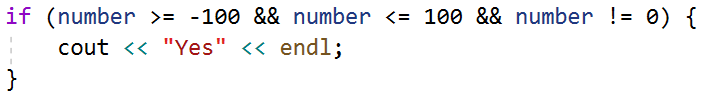
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| -25 | Yes | 0 | No | 25 | Yes |

### Насоки

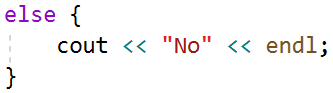
1. Прочетете едно цяло от конзолата:



1. Проверете дали числото е в интервала [-100, 100] и е различно от 0, и ако проверката върне "true", отпечатайте на конзолата "Yes".



1. Ако горепосочената проверка е грешна, отпечатайте "**No**".



## 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"са: **banana**, **apple**, **kiwi**, **cherry**, **lemon** и **grapes**
* Зеленчуците "vegetable" са: **tomato**, **cucumber**, **pepper** и **carrot**
* Всички останали са: "unknown"

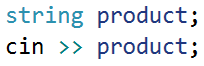
Да се изведе "fruit", "vegetable" или "unknown" според въведения продукт.

### Примерен вход и изход

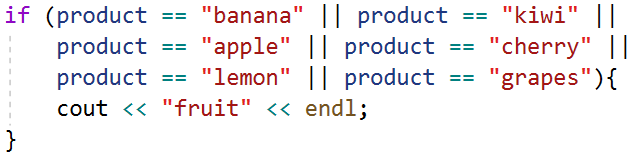
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| banana | fruit | apple | fruit | tomato | vegetable | water | unknown |

### Насоки

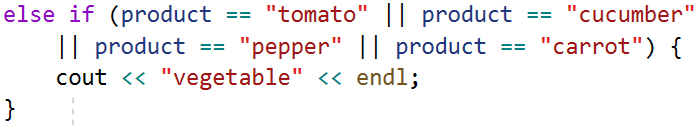
1. Прочетете входните данни (име на продукт) от конзолата под формата на текст:



1. Проверете името на продукта дали съвпада с имената на плодове с помощта на логическо "или" - оператор "||" и ако проверката върне "true", отпечатайте на конзолата "fruit".



1. Ако горепосочената проверка е грешна, проверете дали името на продукта не съвпада с някое от имената на зеленчуците отново с помощта на логическо "или" - оператор "||", при "true" отпечатайте на конзолата "vegetable".



## 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**.

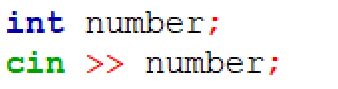
### Примерен вход и изход

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

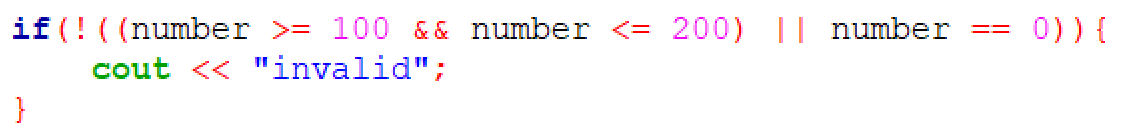
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| -1 | invalid | 100 | *(няма изход)* | 200 | *(няма изход)* | 0 | *(няма изход)* |

### Насоки

1. Прочетете цяло число от конзолата:



1. Направете проверка дали дадено число **не** е валидно като в if проверка използвате логически оператор за отрицание "!" и проверите дали даденото число "number" е по-голямо или равно на 100 и същевременно е по-малко или равно на 200 "number >= 100 && number <= 200", или числото е равно на 0, "number == 0". Ако проверката върне "true", принтирайте на конзолата "invalid".



## 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 |

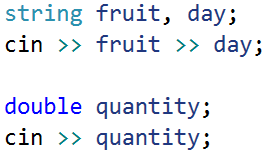
Напишете програма, която чете от конзолата **плод** (banana / apple / orange / grapefruit / kiwi / pineapple / grapes), **ден от седмицата** (Monday / Tuesday / Wednesday / Thursday / Friday / Saturday / Sunday) и **количество** (реално число) , въведени от потребителя, и пресмята **цената** според цените 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 |

### Насоки

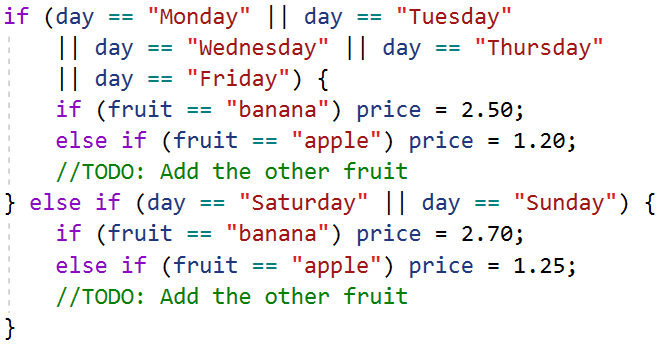
1. Прочетете от входа името на плода и деня от седмицата



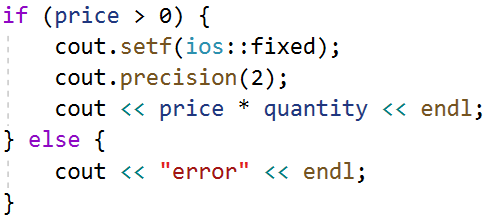
1. Първоначално задайте цена **0**:



1. Използвайте вложени if проверки, като в условието на проверката можете да използвате и логическия оператор "или" - "||", за да изчислите цената за дадения плод и ден от седмицата:



1. Накрая проверете цената. Ако все още е **0**, значи даденият плод или денят от седмицата е **невалиден**. За да отпечатате точно **2 цифри след десетичната запетая.** Кодът може да е подобен на следния:



## 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% |

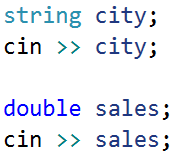
Напишете **конзолна програма**, която чете име на **град** (стринг) и обем на **продажби** (реално число) , въведени от потребителя, и изчислява и извежда размера на търговската **комисионна** според горната таблица. Резултатът да се изведе форматиран до **2 цифри след десетичната точка**. При **невалиден**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

### Насоки

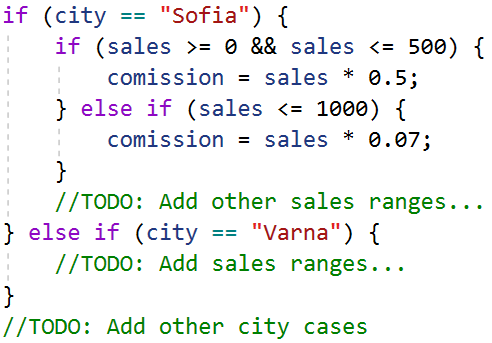
1. Прочетете входа



1. Първоначално задайте **комисионна -1**. Тя ще бъде променена, ако градът и ценовият диапазон бъдат намерени в таблицата с комисионните.



1. Използвайте вложени if проверки, за **да изчислите комисионната** според града и според обема на продажбите. Може да си помогнете с кода по-долу:



1. Ако градът е невалиден отпечатайте "error".