# Лаб: Вложени условни конструкции

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

## Обръщение според възраст и пол

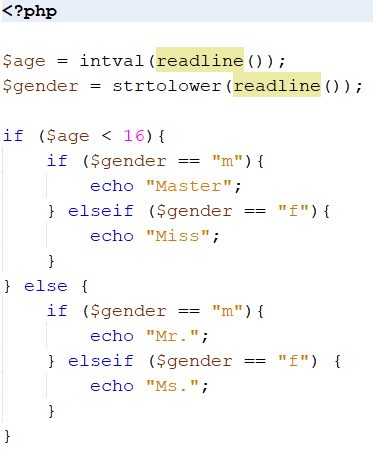
Първата задача от тази тема е да се напише **конзолна програма**, която **прочита възраст** (десетично число) и **пол** ("m" или "f"), въведени от потребителя, и отпечатва **обръщение** измежду следните:

* "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

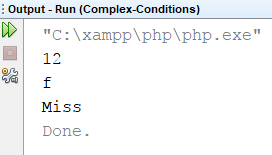
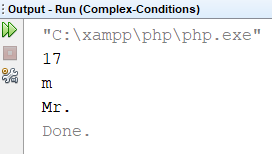
### Примерен вход и изход

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

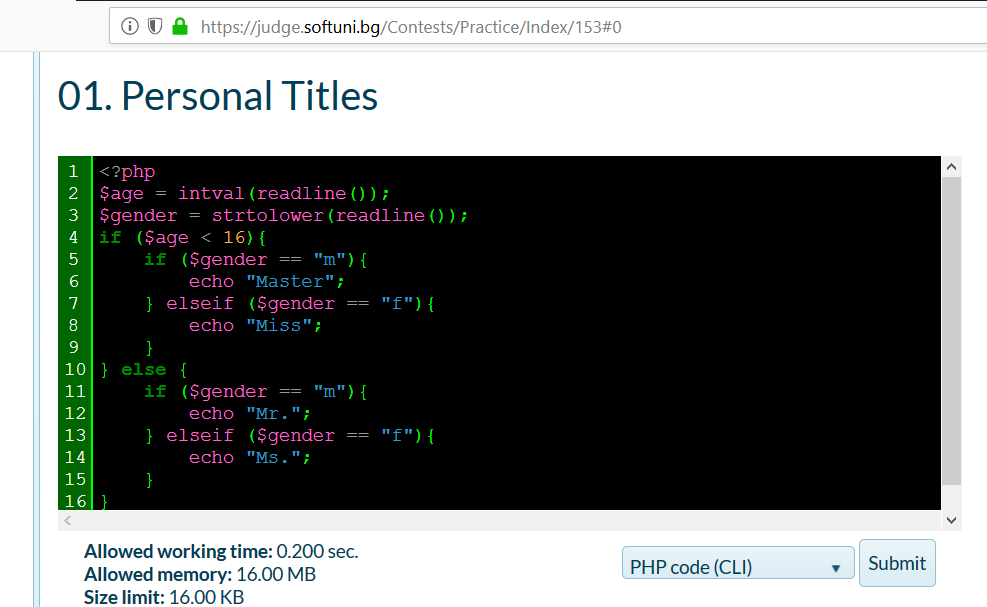
1. Можете да си помогнете с кода от картинката по-долу:



1. **Стартирайте** програмата с [Shift+F6] и я **тествайте** с различни входни стойности:

1. **Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compere/Index/1013#0>. Трябва да получите **100 точки** (напълно коректно решение):





## Квартално магазинче

Следващата задача има за цел да тренира работата с **вложени проверки** (nested if). Ето го и условието: предприемчив българин отваря **квартални магазинчета** в **няколко града** и продава на **различни цени**:

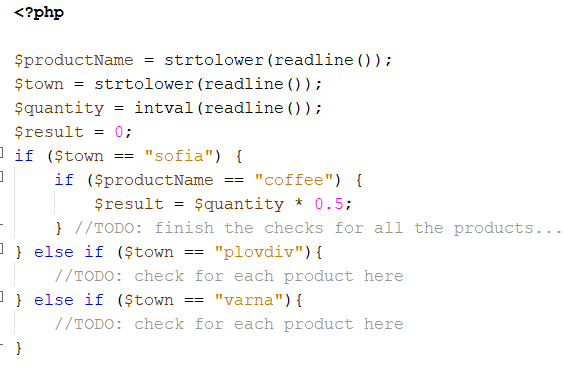
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| град / продукт | **coffee** | **water** | **beer** | **sweets** | **peanuts** |
| **Sofia** | 0.50 | 0.80 | 1.20 | 1.45 | 1.60 |
| **Plovdiv** | 0.40 | 0.70 | 1.15 | 1.30 | 1.50 |
| **Varna** | 0.45 | 0.70 | 1.10 | 1.35 | 1.55 |

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

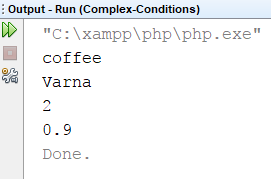
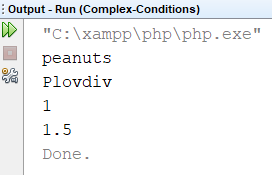
### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| coffee  Varna  2 | 0.9 | peanuts  Plovdiv  1 | 1.5 | beer  Sofia  6 | 7.2 | water  Plovdiv  3 | 2.1 | sweets  Sofia  2.23 | 3.2335 |

1. Можете да си помогнете с кода от картинката по-долу. Можете да прехвърлите всички букви в долен регистър с strtolower() за да сравнявате продукти и градове без значение на малки / главни букви:



1. **Стартирайте** програмата с [Shift+F6] и я **тествайте** с различни входни стойности:

1. **Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1013#1>

## Точка в правоъгълник

Напишете програма, която проверява дали **точка {x, y}** се намира **вътре в правоъгълник {x1, y1} – {x2, y2}**. Входните данни се четат от конзолата и се състоят от 6 реда, въведени от потребителя: десетичните числа **x1**, **y1**, **x2**, **y2**, **x** и **y** (като се гарантира, че **x1 < x2** и **y1 < y2**). Една точка е вътрешна за даден правоъгълник, ако се намира някъде във вътрешността му или върху някоя от страните му. Отпечатайте "**Inside**" или "**Outside**".

### Примерен вход и изход

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **визуализация** |  | **вход** | **изход** | **визуализация** |
| 2  -3  12  3  8  -1 | Inside |  | 2  -3  12  3  11  -3.5 | Outside |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **визуализация** |  | **вход** | **изход** | **визуализация** |
| -1  -3  4  1  0.5  1 | Inside |  | -1  -3  4  1  -1.2  1.4 | Outside |  |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1013#2>

\* **Подсказка**: една точка е вътрешна за даден многоъгълник, ако едновременно са изпълнени следните четири условия (можете да ги проверите с if проверка с логическо "**и**" – оператор &&):

* Точката е надясно от лявата стена на правоъгълника (**x >= x1**)
* Точката е наляво от дясната стена на правоъгълника (**x <= x2**)
* Точката е надолу от горната стена на правоъгълника (**y >= y1**)
* Точката е нагоре от долната стена на правоъгълника (**y <= y2**)

## 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 |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1013#3>

\* **Подсказка**: използвайте условна if проверка с логическо "**или**" – оператор "||".

## 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 |  | 220 | invalid | 199 |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| -1 | invalid | 100 |  | 200 |  | 0 |  |

**Тествайте** решението си в **judge системата**:<https://judge.softuni.bg/Contests/Compete/Index/1013#4>.

\* **Подсказка**: използвайте условна if проверка с **отрицание** и логически операции.

## 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** (десетично 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**".

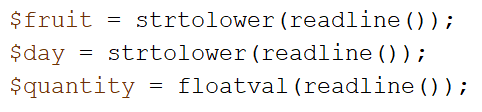
### Примерен вход и изход

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

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1013#5>.

\* **Подсказки**:

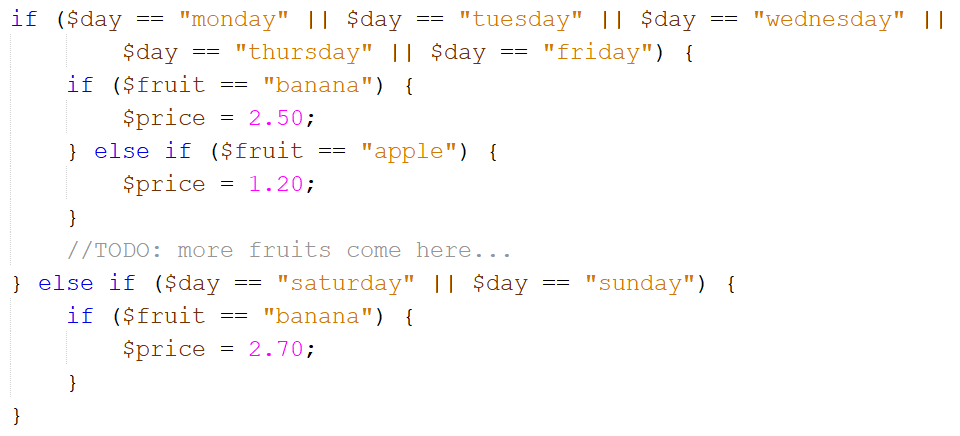
* Прочетете входа и обърнете името на плода и деня от седмицата в **малки букви**:



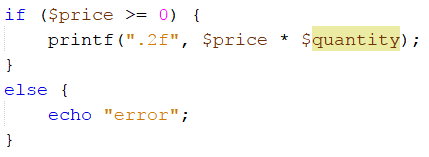
* Първоначално задайте цена **-1**:



* Използвайте вложени if проверки, за да изчислите цената за дадения плод и ден от седмицата:



* Накрая проверете цената. Ако все още е **-1**, значи даденият плод или денят от седмицата е **невалиден**. За да отпечатате точно **2 цифри след десетичната точка** (със закръгляне), използвайте форматиращ низ "%.2f". 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** s:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** и обем на **продажби** , въведени от потребителя, и изчислява и извежда размера на търговската **комисионна** според горната таблица. Резултатът да се изведе закръглен с **2 цифри след десетичната точка**. При **невалиден** град или обем на продажбите (отрицателно число) да се отпечата "**error**".

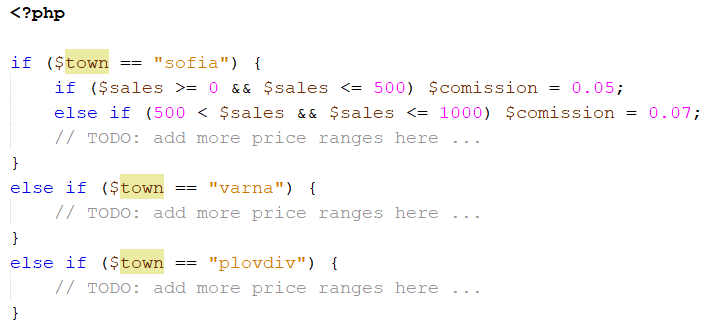
### Примерен вход и изход

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

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1013#6>

\* **Подсказки**:

* Прочетете входа и **обърнете града в** **малки букви** (като в предходната задача).
* Първоначално задайте **комисионна -1**. Тя ще бъде променена, ако градът и ценовият диапазон бъдат намерени в таблицата с комисионните.
* Използвайте вложени if проверки, за **да изчислите комисионната** според града и според обема на продажбите. Може да си помогнете с кода по-долу:



* Накрая проверете комисионната. Ако все още е **-1**, значи въведеният град или обем продажби не се срещат в таблицата с комисионните и трябва да се отпечата "**error**". В противен случай трябва да се изчисли комисионната (процент комисионна по обем на продажбите) и да се отпечата със закръгляне с точно **2 цифри след десетичната точка**. Може да използвате printf("%.2f", …).

## 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 |

**\*Подсказка:** Използвайте условната конструкция [**switch**](http://www.zentut.com/php-tutorial/php-switch-case-statement/).

**Тествайте** решението си в **Judge** системата<https://judge.softuni.bg/Contests/Compete/Index/1013#7>

## Клас животно

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

* **dog -> mammal**
* **crocodile, tortoise, snake -> reptile**
* **others -> unknown**

### Примерен вход и изход

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

**\*Подсказка:** Използвайте условната конструкция switch-case.

**Тествайте** решението си в **Judge** системата: <https://judge.softuni.bg/Contests/Compete/Index/1013#8>

# Примерна изпитна задача

## \* Билети за мач

**Тествайте** решението си [*тук*](https://judge.softuni.bg/Contests/Compete/Index/1013#9).

Когато пуснали **билетите** **за Евро 2016**, **група** запалянковци **решили да си закупят**. **Билетите** имат **две категории с различни цени**:

* **VIP** – **499.99** лева.
* **Normal** – **249.99** лева.

Запалянковците **имат** **определен бюджет**, a **броят на хората** в групата **определя какъв процент от бюджета** трябва **да се задели за транспорт**:

* **От 1 до 4** – **75% от бюджета**.
* **От 5 до** **9** – **60% от бюджета**.
* **От 10 до 24** – **50% от бюджета**.
* **От 25 до 49** – **40% от бюджета**.
* **50 или повече** – **25% от бюджета**.

**Напишете програма**, която да **пресмята** **дали с останалите пари от бюджета** могат да си **купят билети за избраната категория**. И **колко пари** ще им **останат или ще са им нужни**.

### Вход

Програмата чете **точно 3 реда** , въведени от потребителя:

* На **първия** ред е **бюджетът** – **реално число в интервала [1 000.00 ... 1 000 000.00]**
* На **втория** ред е **категорията** – текст с възможности "**VIP**" или "**Normal**"
* На **третия** ред е **броят на хората в групата** – **цяло число в интервала [1 ... 200]**

### Изход

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

* Ако **бюджетът е достатъчен**:
  + "**Yes! You have {N} leva left.**"– **N са останалите пари** на групата
* Ако **бюджетът НЕ Е достатъчен**:
  + "**Not enough money! You need {М} leva.**" – където **М е сумата, която не достига**

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

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 1000  Normal  1 | Yes! You have 0.01 leva left. | 30000  VIP  49 | Not enough money! You need 6499.51 leva. |
| **Обяснения** | | **Обяснения** | |
| **1 човек**: **75% от бюджета** отиват **за** **транспорт**  **Остават:** 1000 – 750 **= 250**  Категория **Normal**: билетът **струва 249.99 \* 1**  **249.99 < 250**: **остават** му 250 – 249.99 = **0.01** | | **49 човек**: **40% от бюджета** отиват **за** **транспорт**  **Остават:** 30000 – 12000 **= 18000**  Категория VIP: билета(ът) **струва 499.99 \* 49**  **24499.510000000002 < 18000**  **Не стигат** 24499.510000000002 - 18000 = **6499.51** | |