# Упражнения: Сложни проверки

Задачи за упражнение в клас и за домашно към курса [„Основи на програмирането за учители“ @ СофтУни](https://softuni.bg/trainings/1672/programming-basics-for-teachers-june-2017).

## Празно Visual Studio решение (Blank Solution)

Създайте празно решение (**Blank Solution**) във Visual Studio за да организирате решенията на задачите от упражненията. Всяка задача ще бъде в отделен проект и всички проекти ще бъдат в общ solution.

1. Стартирайте **Visual Studio**.
2. Създайте нов **Blank Solution**: [File]🡪 [New] 🡪 [Project].



1. Изберете от диалоговия прозорец [Templates] 🡪 [Other Project Types] 🡪 [Visual Studio Solutions] 🡪 [**Blank Solution**] и дайте подходящо име на проекта, например “Complex-Conditions”:



Сега имате създаден **празен Visual Studio Solution** (без проекти в него):



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

## Обръщение според възраст и пол

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

Примери:

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

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 “Personal-Titles”:



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

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



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

 

1. **Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/615#0>. Трябва да получите **100 точки** (напълно коректно решение):





## Квартално магазинче

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 **град** (стринг), **продукт** (стринг) и **количество** (десетично число), въведени от потребителя, и пресмята и отпечатва **колко струва** съответното количество от избрания продукт в посочения град. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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. 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 “Small-Shop”:



Вече имате ново конзолно приложение и остава да напишете кода за решаване на задачата.

1. Отидете в тялото на метода Main(string[] args) и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу. Можете да прехвърлите всички букви в долен регистър с .ToLower() за да сравнявате продукти и градове без значение на малки / главни букви:



1. За **да активирате текущия проект** да стартира при [Ctrl+F5], избере “**Set StartUp Projects…**”:



Изберете първата опция:



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

 

1. **Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/615#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**”. Примери:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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/Practice/Index/615#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. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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/Practice/Index/615#3>.

\* **Подсказка**: използвайте условна if проверка с логическо „**или**“ – operator ||.

## 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**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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)* |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/615#4>.

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

## Точка върху страната на правоъгълник

Напишете програма, която проверява дали **точка {x, y}** се намира **върху някоя от страните на 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**). Да се отпечата “Border” (точката лежи на някоя от страните) или “Inside / Outside” (в противен случай). Примери:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **визуализация** |  | **вход** | **изход** | **визуализация** |
| 2  -3  12  3  8  -1 | Inside / Outside |  | 2  -3  12  3  12  -1 | Border |  |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/615#5>.

\* **Подсказка**: използвайте една или няколко условни if проверки с логически операции. Точка **{x, y}** лежи върху някоя от страните на правоъгълник **{x1, y1} – {x2, y2}**, ако е изпълнено едно от следните условия:

* **x** съвпада с **x1** или **x2** и същевременно **y** е между **y1** и **y2**
* **y** съвпада с **y1** или **y2** и същевременно **x** е между **x1** и **x2**

Можете да проверите горните условия с една по-сложна if-else конструкция или с няколко по-прости проверки или с **вложени** if-else проверки.

## 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**.

Примери:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| 1 | Monday |  | 5 | Friday |
| 2 | Tuesday |  | 6 | Saturday |
| 3 | Wednesday |  | 7 | Sunday |
| 4 | Thursday |  | 8 | Error |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/615#6>.

## Клас животно

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

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

Примери:

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

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/615#7>.

## \* 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. Примери:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| Premiere  10  12 | 1440.00 leva | Normal  21  13 | 2047.50 leva | Discount  12  30 | 1800.00 leva |

**Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/615#8>.

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