# Упражнения: Серия от проверки

## Празно Visual Studio решение (Blank Solution)

Създайте празно решение от Microsoft Visual Studio, за да организирате решенията на задачите от упражненията, като всяка задача ще бъде в отделен проект и всички проекти ще бъдат в общото решение.

1. Стартирайте Visual Studio.
2. Създайте нов проект [Create New Project].

Graphical user interface, text, application

Description automatically generated

1. Търсете и изберете: Blank Solution

Graphical user interface, text, application, email

Description automatically generated

1. Именувайте решението: Simple-Calculations

Graphical user interface, text, application, email

Description automatically generated

Сега имате създаденo празно решение без проекти в него.

Целта e да добавяте в него по един проект за всяка задача от упражненията.

## Изписване на число до 9 с думи

Да се напише програма, която чете цяло число в диапазона [1…9], въведено от потребителя, и го изписва с думи на английски език. Ако числото е извън диапазона, изписва “number too big”. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| 5 | five | 1 | one | 9 | nine | 10 | number too big |

Подсказка: можете да напишете дълга if-else-if-else…else, с която да разгледате възможните 10 случая.

## Бонус точки

Дадено е цяло число – брой точки. Върху него се начисляват бонус точки по правилата, описани по-долу. Да се напише програма, която пресмята бонус точките за това число и общия брой точки с бонусите.

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

Примери:

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

Ето как би могло да изглежда решението на задачата в действие:



Подсказка:

* Основните бонус точки можете да изчислите с if-else-if-else-if конструкция (имате 3 случая).
* Допълнителните бонус точки можете да изчислите с if-else-if конструкция (имате още 2 случая).

## Сумиране на секунди

Трима спортни състезатели финишират за някакъв брой секунди (между 1 и 50). Да се напише програма, която чете времената на състезателите, 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"). Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Подсказка:

* Сумирайте трите числа и получете резултата в секунди. Понеже 1 минута = 60 секунди, ще трябва да изчислите броя минути и броя секунди в диапазона от 0 до 59.
* Ако резултатът е между 0 и 59, отпечатайте 0 минути + изчислените секунди.
* Ако резултатът е между 60 и 119, отпечатайте 1 минута + изчислените секунди минус 60.
* Ако резултатът е между 120 и 179, отпечатайте 2 минути + изчислените секунди минус 120.
* Ако секундите са по-малко от 10, изведете водеща нула преди тях.

## Конвертор за мерни единици

Да се напише програма, която преобразува разстояние между следните 8 мерни единици: m, mm, cm, mi, in, km, ft, yd. Използвайте съответствията от таблицата по-долу:

|  |  |
| --- | --- |
| входна единица | изходна единица |
| 1 meter (m) | 1000 millimeters (mm) |
| 1 meter (m) | 100 centimeters (cm) |
| 1 meter (m) | 0.000621371192 miles (mi) |
| 1 meter (m) | 39.3700787 inches (in) |
| 1 meter (m) | 0.001 kilometers (km) |
| 1 meter (m) | 3.2808399 feet (ft) |
| 1 meter (m) | 1.0936133 yards (yd) |

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

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

Примерен вход и изход:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |
| 12  km  ft | 39370. 07886932 ft | 150 mi in | 9503999.99393599 in | 450  yd  km | 0.41148 km |

## 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:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |
| 95 | Less than 100 | 120 | Between 100 and 200 | 210 | Greater than 200 |

Подсказка: използвайте if-else-if-else конструкция за да проверите всеки от трите случая.

## Информация за скоростта

Да се напише програма, която чете скорост (десетично число), въведена от потребителя, и 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 |

Подсказка: използвайте серия от if-else-if-else-… конструкции, за да хванете всичките 5 случая.

## 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.

Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| square  5 | 25 | rectangle  7  2.5 | 17.5 | circle  6 | 113.097 | triangle  4.5  20 | 45 |

Подсказка: използвайте серия от if-else-if-else-… конструкции, за да обработите 4-те вида фигури.

## \* Изписване на число до 100 с думи

Да се напише програма, която превръща число [0…100] в текст.

Подсказка: използвайте серия от if-else-if-else-… конструкции, като помислите дали може да са на повече от едно ниво.