

Упражнения: Прости проверки

Задачи за упражнение в клас и за домашно към курса [„Основи на програмирането“ @ СофтУни](#).

0. 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].
3. Изберете от диалоговия прозорец [Templates] → [Other Project Types] → [Visual Studio Solutions] → [**Blank Solution**] и дайте подходящо име на проекта, например “Simple-Conditions”:

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

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

1. 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	(<i>няма изход</i>)	5.50	Excellent!	5.49	(<i>няма изход</i>)

1. Създайте **нов проект** в съществуващото Visual Studio решение. В Solution Explorer кликнете с десен бутон на мишката върху **Solution 'Simple-Conditions'**. Изберете [Add] → [New Project...]:
2. Ще се отвори диалогов прозорец за избор на тип проект за създаване. Изберете C# конзолно приложение и задайте име “Excellent-Result”:
Вече имате solution с едно конзолно приложение в него. Остава да напишете кода за решаване на задачата.
3. Отидете в тялото на метода Main(string[] args) и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:
4. **Стартирайте** програмата с [Ctrl+F5] и я **тествайте** с различни входни стойности:

5. Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#0>. Трябва да получите **100 точки** (напълно коректно решение):

2. Отлична оценка или не

Следващата 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, или **“Not excellent.”** в противен случай.

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
6	Excellent!	5	Not excellent.	5.50	Excellent!	5.49	Not excellent.

- Първо създайте **нов C# конзолен проект** в решението **“Simple-Conditions”**.
 - Кликнете с мишката върху решението в Solution Explorer и изберете [Add] → [New Project...].
 - Изберете [Visual C#] → [Windows] → [Console Application] и задайте име **“Excellent-or-Not”**.
- Напишете кода** на програмата. Може да си помогнете с примерния код от картинката:
- Включете режим на **автоматично превключване към текущия проект** като кликнете върху главния solution с десния бутон на мишката и изберете [Set StartUp Projects...]:
Ще се появи диалогов прозорец, от който трябва да се избере [Startup Project] → [Current selection]:

- Сега **стартирайте програмата**, както обикновено с [Ctrl+F5] и я тествайте:
- Тествайте в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#1>.
Решението би трябвало да бъде прието като напълно коректно:

3. Четно или нечетно

Да се напише програма, която чете **цяло число**, въведено от потребителя, и печата дали е **четно** или **нечетно**. Примери:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
2	even	3	odd	25	odd	1024	even

- Първо добавете **нов C# конзолен проект** в съществуващия solution.
- Напишете кода** на програмата. Проверката за четност може да се реализира чрез проверка на **остатъка при деление на 2** по следния начин: `var even = (num % 2 == 0)`.
- Стартирайте програмата** с [Ctrl+F5] и я тествайте:

4. Намиране на по-голямото число

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

ВХОД	ИЗХОД
5 3	5

ВХОД	ИЗХОД
3 5	5

ВХОД	ИЗХОД
10 10	10

ВХОД	ИЗХОД
-5 5	5

1. Първо добавете **нов C# конзолен проект** в съществуващия solution.
2. **Напишете кода** на програмата. Необходима е единична if-else конструкция.
3. **Стартирайте** програмата с [Ctrl+F5] и я тествайте:
4. Тествайте решението си в **judge системата**:
<https://judge.softuni.bg/Contests/Practice/Index/152#3>.

Подсказка: може да си помогнете частично с кода от картинката, който е нарочно замъглен, за да помислите как да си го напишете сами:

5. Изписване на число до 10 с думи

Да се напише програма, която чете **цяло число в диапазона [0...9]**, въведено от потребителя, и го **изписва с думи** на английски език. Ако числото е извън диапазона, изписва **“number too big”**. Примери:

ВХОД	ИЗХОД
5	five

ВХОД	ИЗХОД
1	one

ВХОД	ИЗХОД
9	nine

ВХОД	ИЗХОД
10	number too big

6. 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 $\rightarrow + 2$ т.

Примери:

ВХОД	ИЗХОД
20	6 26

ВХОД	ИЗХОД
175	37 212

ВХОД	ИЗХОД
2703	270.3 2973.3

ВХОД	ИЗХОД
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).

7. 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** ($2 \rightarrow "02"$, $7 \rightarrow "07"$, $35 \rightarrow "35"$). Примери:

ВХОД	ИЗХОД
35 45 44	2:04

ВХОД	ИЗХОД
22 7 34	1:03

ВХОД	ИЗХОД
50 50 49	2:29

ВХОД	ИЗХОД
14 12 10	0:36

Тествайте решението си в **judge системата**:

<https://judge.softuni.bg/Contests/Practice/Index/152#6>.

Подсказка:

Сумирайте трите числа и получите 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.

8. 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)

Резултатът да се **форматира** до **осмия знак след десетичната запетая**.

Примерен вход и изход:

ВХОД	ИЗХОД
12 km ft	39370. 07886932

ВХОД	ИЗХОД
150 mi in	9503999.99393599

ВХОД	ИЗХОД
450 yd km	0.41148000

Тествайте решението си в **judge системата**:
<https://judge.softuni.bg/Contests/Practice/Index/152#7>.

9. 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!**”.
 Примери:

ВХОД	ИЗХОД
qwerty	Wrong password!

ВХОД	ИЗХОД
s3cr3t!P@ssw0rd	Welcome

ВХОД	ИЗХОД
s3cr3t!p@ss	Wrong password!

Тествайте решението си в **judge системата**:
<https://judge.softuni.bg/Contests/Practice/Index/152#8>.

Подсказка: използвайте if-else конструкцията.

10. 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 отпечатаат съответно съобщения като в примерите по-долу:

ВХОД	ИЗХОД
95	Less than 100

ВХОД	ИЗХОД
120	Between 100 and 200

ВХОД	ИЗХОД
210	Greater than 200

Тествайте решението си в **judge системата**:
<https://judge.softuni.bg/Contests/Practice/Index/152#9>.

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

11. Еднакви думи

Да се напише програма, която **чете две думи**, въведени от потребителя, и проверява дали са еднакви. Да не се прави разлика между главни и малки думи. Да се изведе “yes” или “no”. Примери:

ВХОД	ИЗХОД
Hello Hello	yes

ВХОД	ИЗХОД
SoftUni softuni	yes

ВХОД	ИЗХОД
Soft Uni	no

ВХОД	ИЗХОД
beer vodka	no

ВХОД	ИЗХОД
HeLlO hELLo	yes

Тествайте решението си в **judge системата**:
<https://judge.softuni.bg/Contests/Practice/Index/152#10>.

Подсказка: използвайте if-else конструкция. Преди сравняване на думите ги обърнете в долен регистър: word = word.ToLower().

12. 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
8	slow

ВХОД	ИЗХОД
49.5	average

ВХОД	ИЗХОД
126	fast

ВХОД	ИЗХОД
160	ultra fast

ВХОД	ИЗХОД
3500	extremely fast

Тествайте решението си в **judge системата**:

<https://judge.softuni.bg/Contests/Practice/Index/152#11>.

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

13. Лица на фигури

Да се напише програма, в която потребителят **въвежда вида и размерите на геометрична** фигура и пресмята лицето ѝ. 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
square 5	25

ВХОД	ИЗХОД
rectangle 7 2.5	17.5

ВХОД	ИЗХОД
circle 6	113.097

ВХОД	ИЗХОД
triangle 4.5 20	45

Тествайте решението си в **judge системата**:

<https://judge.softuni.bg/Contests/Practice/Index/152#12>.

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

14. 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 hh:mm. 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. Примери:

ВХОД	ИЗХОД
1 46	2:01

ВХОД	ИЗХОД
0 01	0:16

ВХОД	ИЗХОД
23 59	0:14

ВХОД	ИЗХОД
11 08	11:23

ВХОД	ИЗХОД
12 49	13:04

Тествайте решението си в **judge системата**:
<https://judge.softuni.bg/Contests/Practice/Index/152#13>.

Подсказка: добавете 15 минути и направете няколко проверки. Ако минутите надвишат 59, увеличете часовете с 1 и намалете минутите със 60. По аналогичен начин разгледайте случая, когато часовете надвишат 23. При печатането на минутите проверете за водеща нула.

15. Еднакви 3 числа

Три еднакви числа: да се въведат 3 числа и да се отпечата дали са еднакви (yes / no).

16. *Изписване на число до 100 с думи

Да се напише програма, която превръща число [0...100] в текст: 25 → “twenty five”