

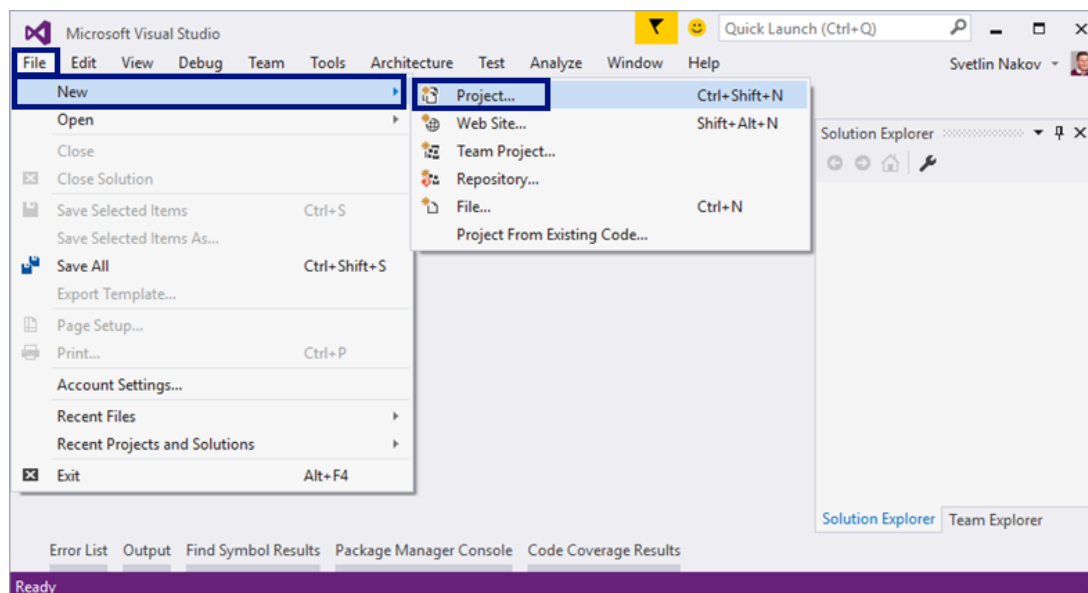
# Упражнения: Прости проверки

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

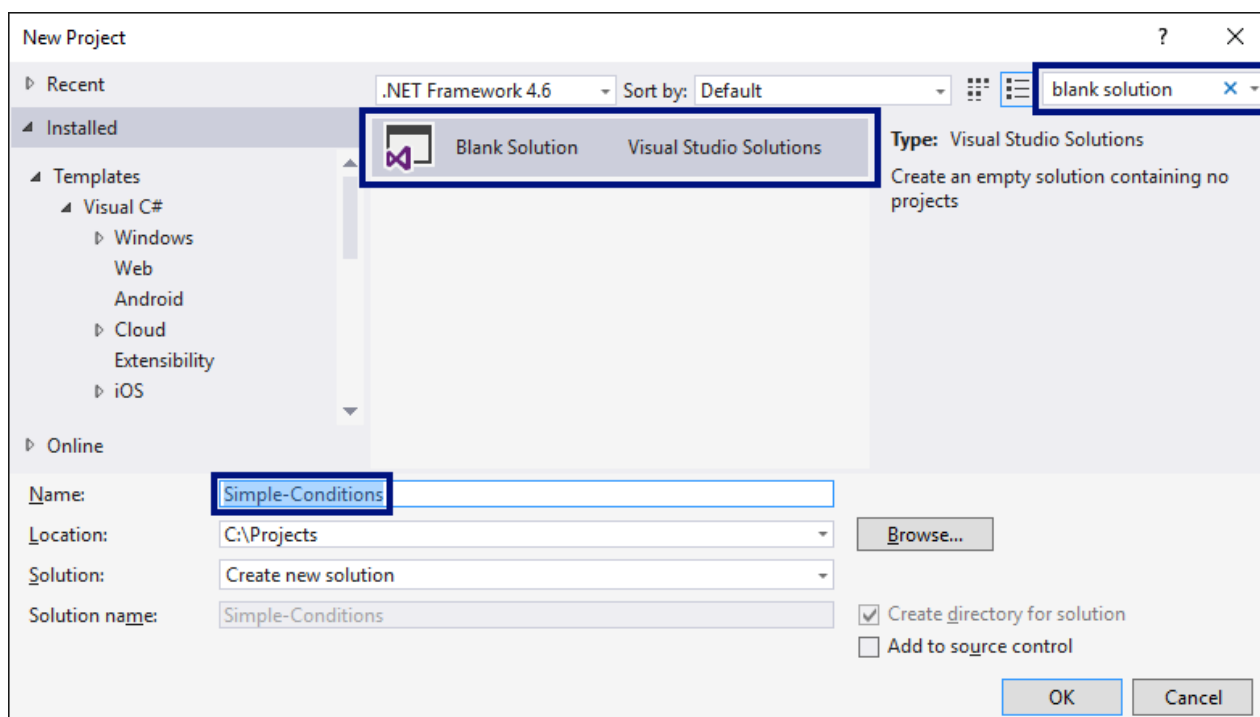
## 1. Празно Visual Studio решение (Blank Solution)

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.

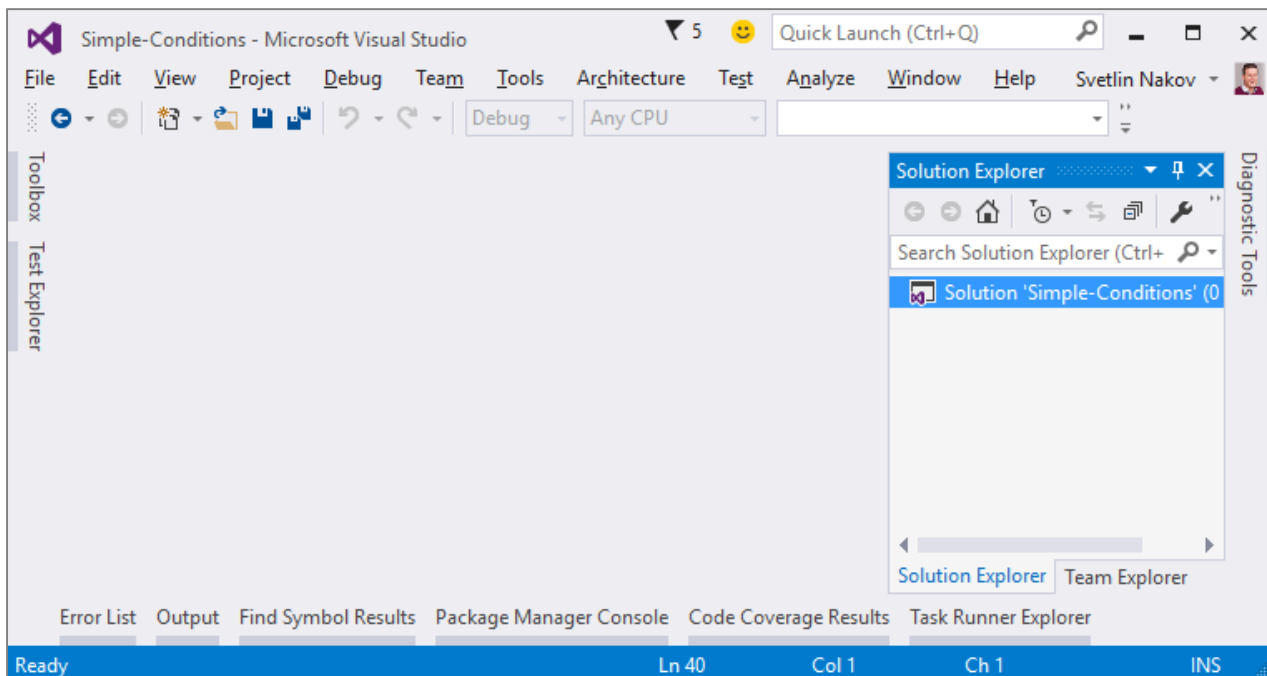
1. Стартирайте Visual Studio.
2. Създайте нов **Blank Solution**: [File] → [New] → [Project].



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):



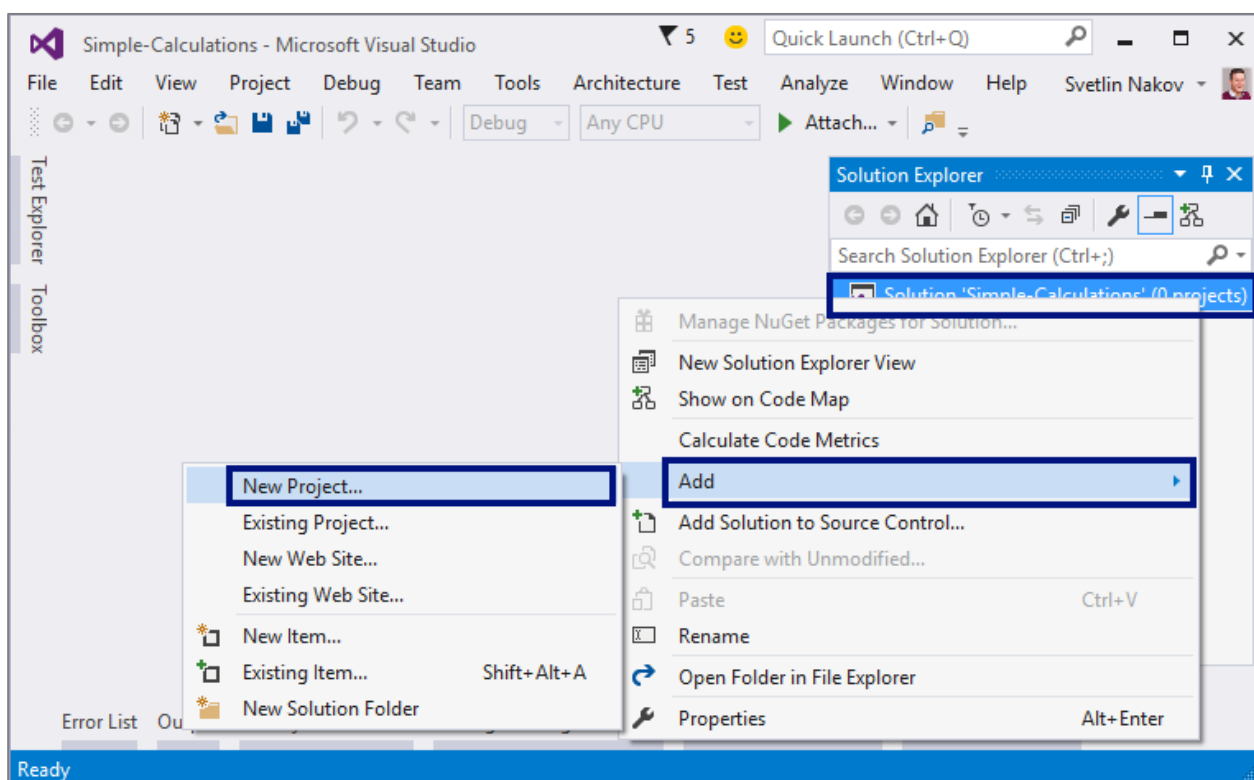
Целта на този **blank solution** е да добавяте в него **по един проект за всяка задача** от упражненията.

## 2. Проверка за отлична оценка

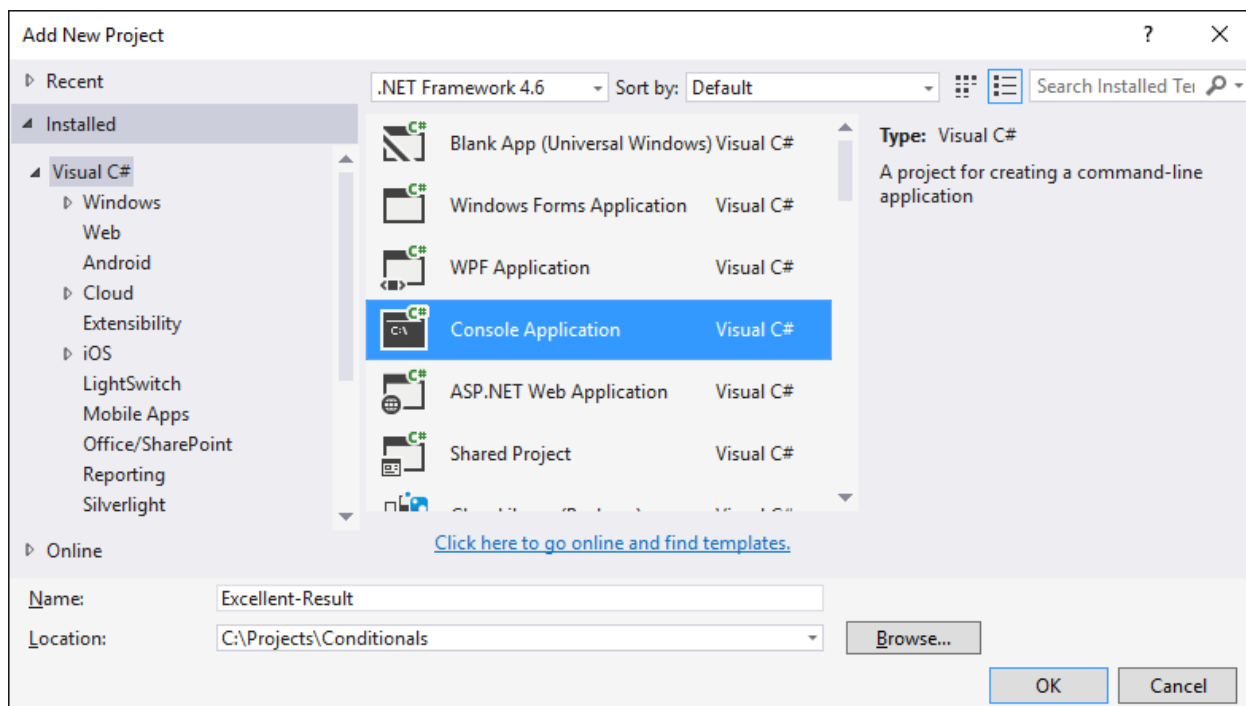
Първата задача от тази тема е да се напише **конзолна програма**, която **въвежда оценка** (десетично число) и отпечатва **“Excellent!”**, ако оценката е **5.50** или по-висока.

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
6	Excellent!	5	(няма изход)	5.50	Excellent!	5.49	(няма изход)

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”**:

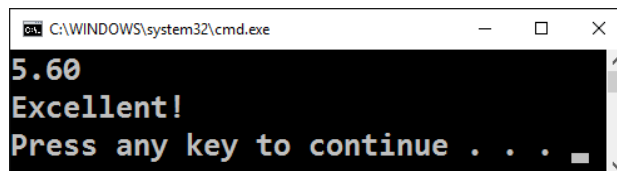
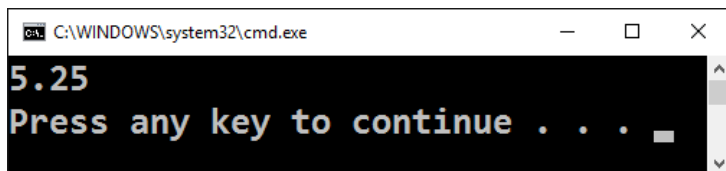


Вече имате solution с едно конзолно приложение в него. Остава да напишете кода за решаване на задачата.

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

```
namespace Excellent_Result
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            var grade = double.Parse(Console.ReadLine());
            if (grade >= 5.50)
            {
                Console.WriteLine("Excellent!");
            }
        }
    }
}
```

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):

Conditional Statements - 5 X

https://judge.softuni.bg/Contests/Practice/Index/152#0

# Excellent Result

Participants Tests Change Delete

Administration |

```

4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Excellent_Result
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            var grade = double.Parse(Console.ReadLine());
14            if (grade >= 5.50)
15            {
16                Console.WriteLine("Excellent!");
17            }
18        }
19    }
20 }
21

```

Allowed working time: 0.100 sec.  
Allowed memory: 16.00 MB

C# code Submit

Submissions			
<div> <div>1</div> <div> <div>⏮</div> <div>⏪</div> <div>⏩</div> <div>⏭</div> </div> </div>			
Points	Time and memory used	Submission date	
<div> <div>✓✓✓✓✓✓</div> <div>100 / 100</div> </div>	Memory: 7.81 MB Time: 0.029 s	17:51:38 26.01.2016	Details

### 3. Отлична оценка или не

Следващата задача от тази тема е да се напише **конзолна програма**, която **въвежда оценка** (десетично число) 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.

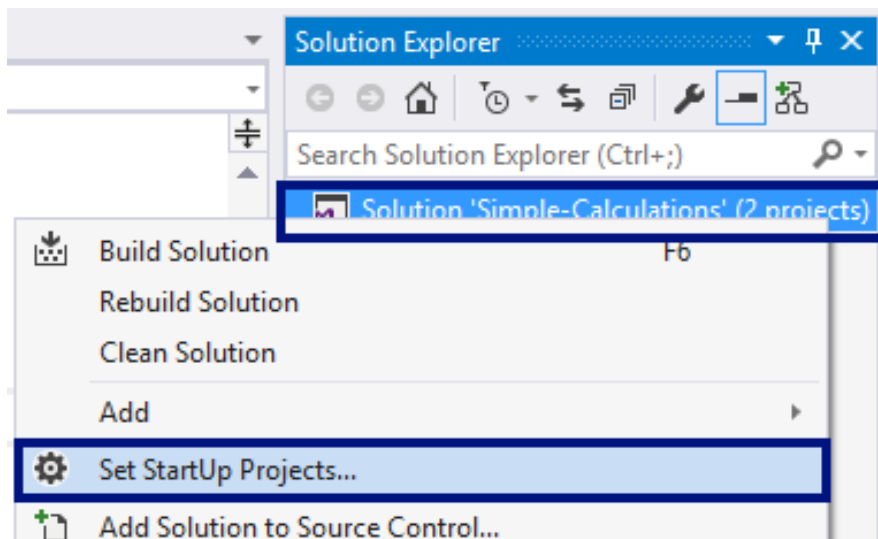
- Първо създайте **нов C# конзолен проект** в решението **"Simple-Conditions"**.
  - Кликнете с мишката върху решението в Solution Explorer и изберете [Add] → [New Project...].
  - Изберете [Visual C#] → [Windows] → [Console Application] и задайте име **"Excellent-or-Not"**.
- Напишете кода** на програмата. Може да си помогнете с примерния код от картинката:

```

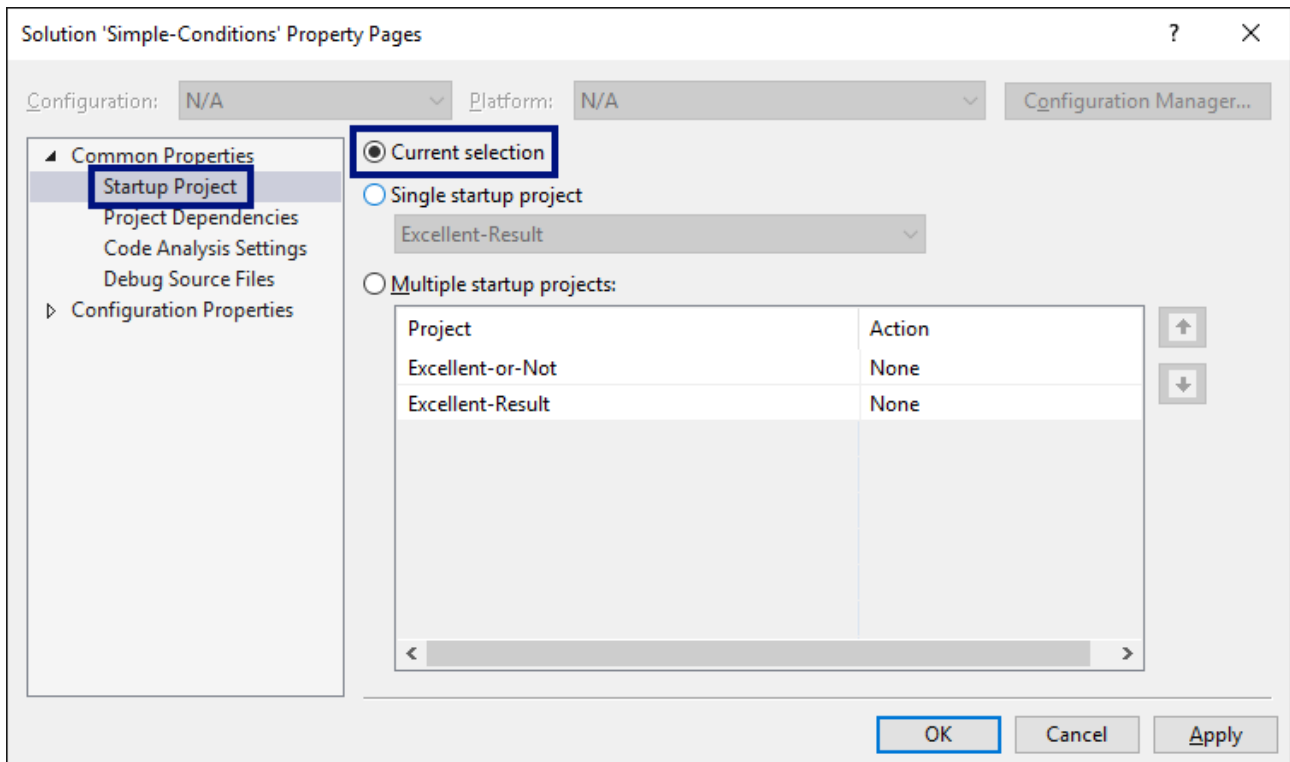
var grade = double.Parse(Console.ReadLine());
if (grade >= 5.50)
{
    Console.WriteLine("Excellent!");
}
else
{
    Console.WriteLine("Not excellent.");
}

```

- 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:

```
C:\WINDOWS\system32\cmd.exe
5.60
Excellent!
Press any key to continue . . .
```

```
C:\WINDOWS\system32\cmd.exe
4.25
Not excellent.
Press any key to continue . . .
```

5. Тествайте в **judge** системата: <https://judge.softuni.bg/Contests/Practice/Index/152#1>. Решението би трябвало да бъде прието като напълно коректно:

Submissions			
<div>⏮ ⏪ 1 ⏩ ⏭ ↻</div>			
Points	Time and memory used	Submission date	
✓✓✓✓ 100 / 100	Memory: 7.83 MB Time: 0.014 s	22:37:59 21.01.2016	<button>Details</button>
<div>⏮ ⏪ 1 ⏩ ⏭ ↻</div>			

## 4. Четно или нечетно

Да се напише програма, която въвежда **цяло число** и печата дали е **четно** или **нечетно**. Примери:

вход	изход
2	even

вход	изход
3	odd

вход	изход
25	even

вход	изход
1024	odd

- Първо добавете **нов C# конзолен проект** в съществуващия solution.
- Напишете кода** на програмата. Проверката за честност 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:

```
C:\WINDOWS\system32\cmd.exe
42
even
Press any key to continue . . .
```

4. Тествайте в **judge** системата: <https://judge.softuni.bg/Contests/Practice/Index/152#2>.

## 5. Намиране на по-голямото число

Да се напише програма, която въвежда **две цели числа** 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

- Първо добавете **нов C# конзолен проект** в съществуващия solution.
- Напишете кода** на програмата. Необходима е единична **if-else** конструкция.
- Стартирайте** програмата с **[Ctrl+F5]** и я тествайте:

```

C:\WINDOWS\system32\cmd.exe
Enter two integers:
5
3
Greater number: 5
Press any key to continue . . .

```

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:

```

static void Main(string[] args)
{
    Console.WriteLine("Enter two integers:");
    var num1 = int.Parse(Console.ReadLine());
    var num2 = int.Parse(Console.ReadLine());
    // Console.WriteLine("Greater number: " + num1);
    // Console.WriteLine("Greater number: " + num2);
}

```

## 6. Изписване на число до 9 с думи

Да се напише програма, която въвежда **цяло число в диапазона [0...9]** 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

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#4>.

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

## 7. 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:

```
C:\WINDOWS\system32\cmd.exe
Enter score: 20
Bonus score: 6
Total score: 26
Press any key to continue . . .
```

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#5>.

**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).

## 8. Сумиране на секунди

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

## 9. 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:



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

Примерен вход и изход:

ВХОД	ИЗХОД
12 km ft	39370.0788 ft

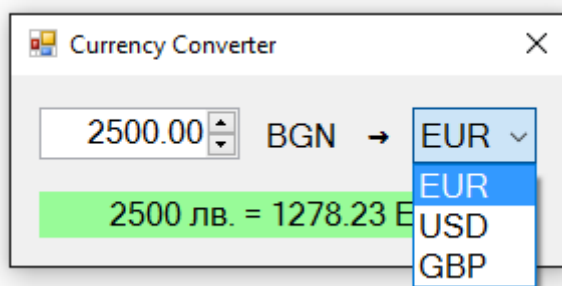
ВХОД	ИЗХОД
150 mi in	9503999.99393599 mi

ВХОД	ИЗХОД
450 yd km	0.41147999937455 yd

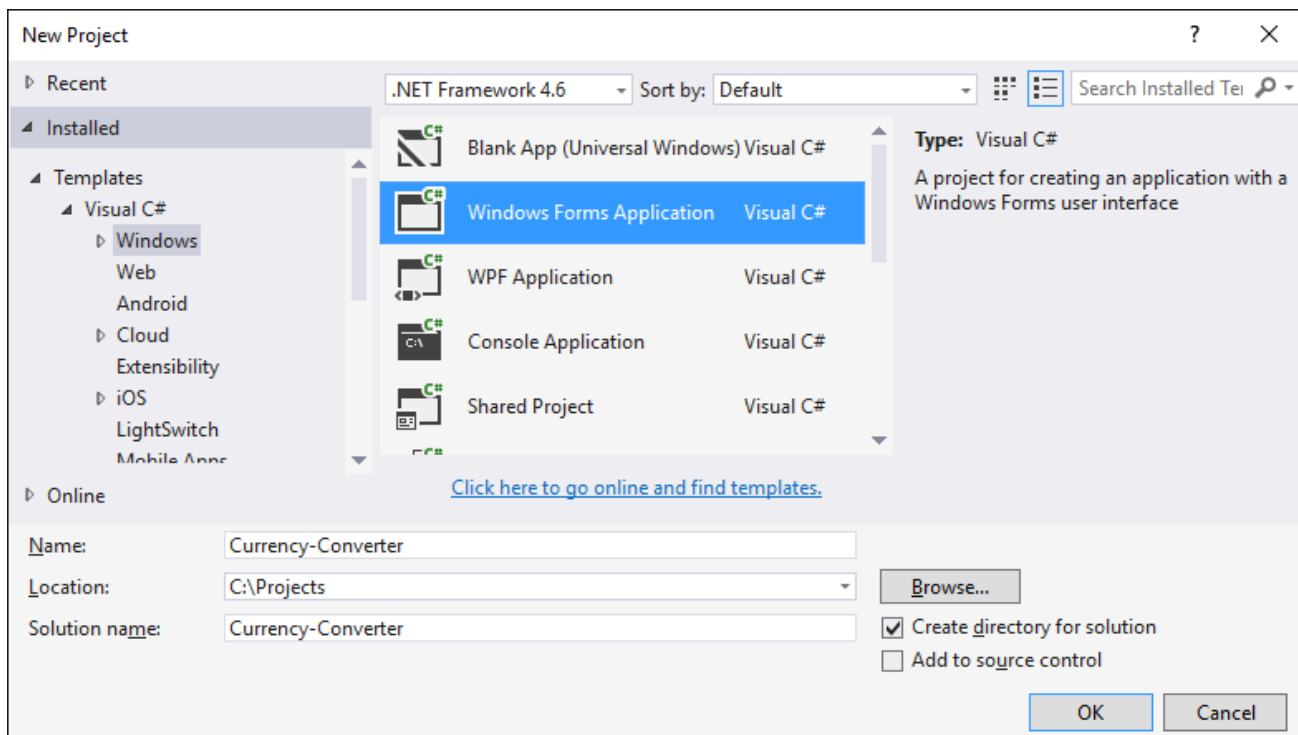
Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#7>.

## 10. Графично приложение: конвертор за валути

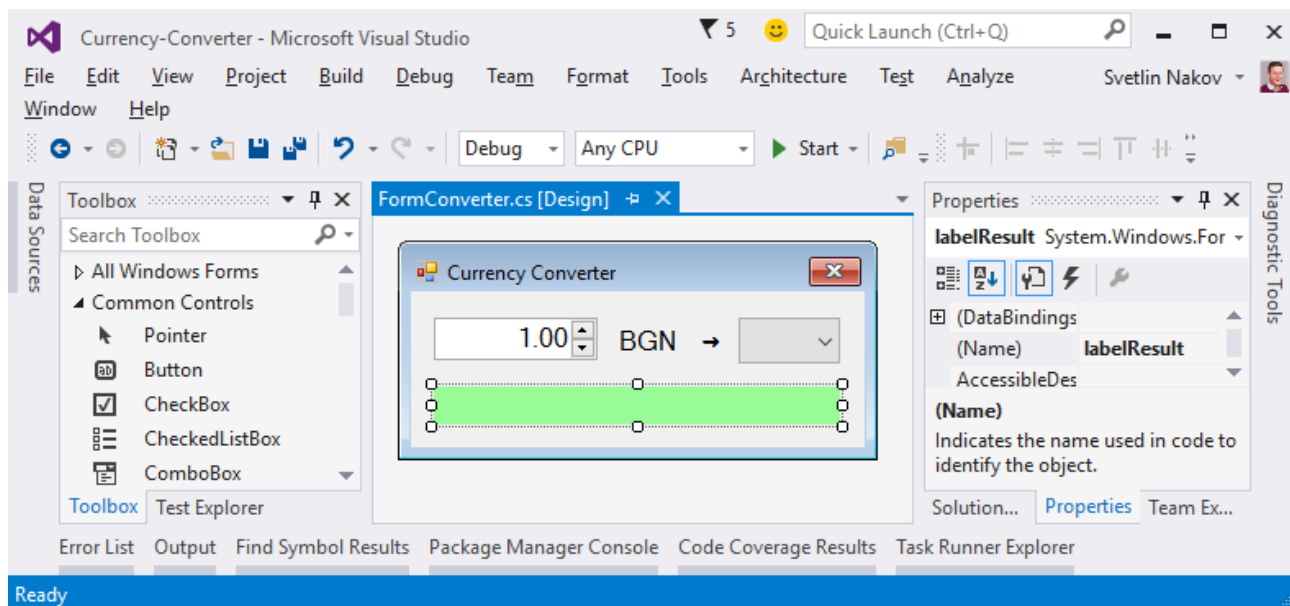
Създайте графично (GUI) приложение за **конвертиране на валути**. Приложението трябва да изглежда приблизително като на картинката по-долу:



1. Създайте нов **Windows Forms Application** с име "Currency-Converter":



2. **Наредете контролите** във формата: една кутийка за въвеждане на число (**NumericUpDown**), един падащ списък с валути (**ComboBox**), текстов блок за резултата (**Label1**) и няколко надписа (**Label1**). Нагласете **размерите** и **свойствата** им, за да изглеждат долу-горе като на картинката:



3. Задайте следните препоръчителни **настройки на контролите**:

За **главната форма (Form)**, която съдържа всички контроли:

- (name) = **FormConverter**
- **Text** = "Currency Converter"
- **Font.Size** = 12
- **MaximizeBox** = False
- **MinimizeBox** = False
- **FormBorderStyle** = FixedSingle

За **полето за въвеждане на число (NumericUpDown)**:

- (name) = **numericUpDownAmount**
- **Value** = 1
- **Minimum** = 0
- **Maximum** = 1000000
- **TextAlign** = Right
- **DecimalPlaces** = 2

За **падащия списък в валутите (ComboBox)**:

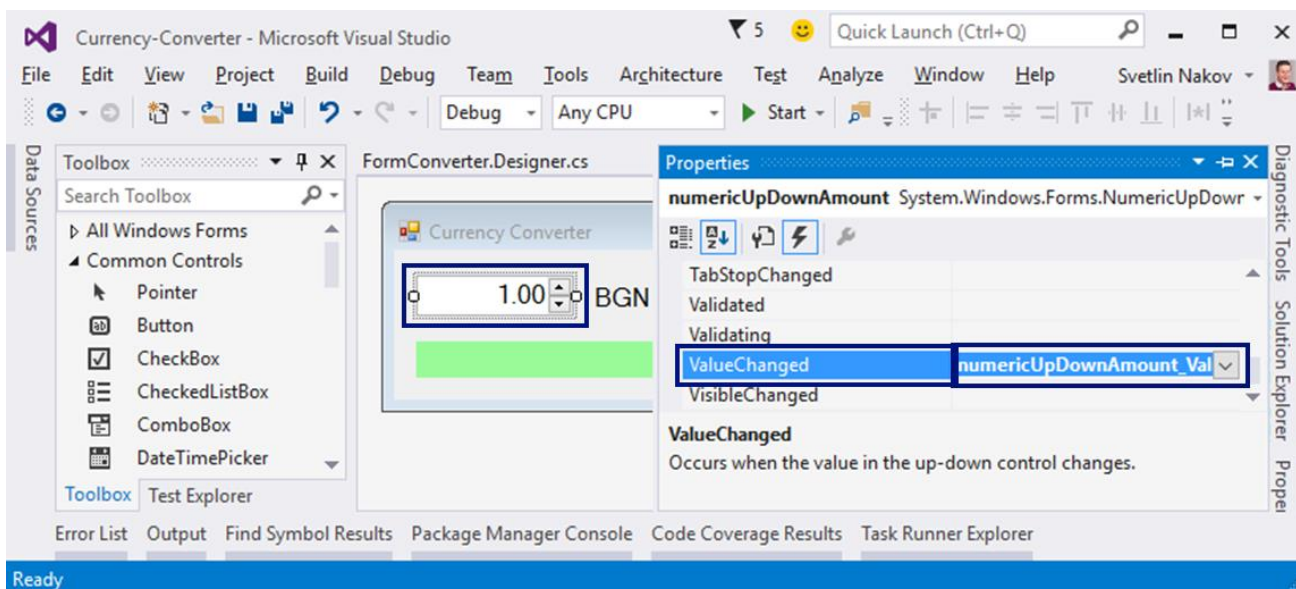
- (name) = **comboBoxCurrency**
- **DropDownStyle** = DropDownList
- **Items** =
  - EUR
  - USD
  - GBP

За **текстовия блок за резултата (Label)**:

- (name) = **labelResult**
- **AutoSize** = False
- **BackColor** = PaleGreen
- **TextAlign** = MiddleCenter
- **Font.Size** = 14
- **Font.Bold** = True

4. Хванете следните **събития**, за да напишете C# кода, който ще се изпълни при настъпването им:

- Събитието **ValueChanged** на контролата за въвеждане на число **numericUpDownAmount**:



- Събитието **Load** на формата **FormConverter**.
- Събитието **SelectedIndexChanged** на падащия списък за избор на валута **comboBoxCurrency**.

5. Напишете следния **C# код** за обработка на събитията:

```
private void FormConverter_Load(object sender, EventArgs e)
{
    this.comboBoxCurrency.SelectedItem = "EUR";
}

private void numericUpDownAmount_ValueChanged(object sender, EventArgs e)
{
    ConvertCurrency();
}

private void comboBoxCurrency_SelectedIndexChanged(object sender, EventArgs e)
{
    ConvertCurrency();
}
```

Задачата на горния код е да избере при стартиране на програмата валута "EUR" и при промяна на стойностите в полето за сума или при смяна на валутата да изчисли резултата, извиквайки **ConvertCurrency()**.

6. Следва да се напише действието **ConvertCurrency()** за конвертиране на въведената сума от лева в избраната валута:

```
private void ConvertCurrency()
{
    var originalAmount = this.numericUpDownAmount.Value;
    var convertedAmount = originalAmount;
    if (this.comboBoxCurrency.SelectedItem.ToString() == "EUR")
    {
        convertedAmount = originalAmount / 1.95583m;
    }
    else if (this.comboBoxCurrency.SelectedItem.ToString() == "USD")
    {
        convertedAmount = originalAmount / 1.80810m;
    }
}
```

```

    }
    else if (this.comboBoxCurrency.SelectedItem.ToString() == "GBP")
    {
        convertedAmount = originalAmount / 2.54990m;
    }
    this.labelResult.Text = originalAmount + " лв. = " +
        Math.Round(convertedAmount, 2) + " " + this.comboBoxCurrency.SelectedItem;
}

```

Горният код взима **сумата** за конвертиране от полето **numericUpDownAmount** и **избраната валута** за резултата от полето **comboBoxCurrency**. След това с **условна конструкция** според избраната валута, сумата се дели на **валутния курс** (който е фиксиран твърдо в сорс кода). Накрая се генерира текстово **съобщение с резултата** (закръглен до 2 цифри след десетичната точка) и се записва в зелената кутийка **labelResult**.

## 11. Познай паролата

Да се напише програма, която **въвежда парола** (един ред с произволен текст) 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.

## 12. Число от 100 до 200

Да се напише програма, която **въвежда цяло число** 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.

## 13. Еднакви думи

Да се напише програма, която **въвежда две думи** 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()**.

## 14. Информация за скоростта

Да се напише програма, която **въвежда скорост** (десетично число) 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

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.

## 15. Лица на фигури

Да се напише програма, която **въвежда 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	вход	изход	вход	изход	вход	изход
square 5	25	rectangle 7 2.5	17.5	circle 6	113.097	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.

## 16. Време + 15 минути

Да се напише програма, която **въвежда час и минути** от 24-часово денонощие 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	вход	изход	вход	изход	вход	изход	вход	изход
1 46	2:01	0 01	0:16	23 59	0:14	11 08	11:23	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 за препълване на минутите и часовете. 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.

Има и по-кратък начин: с изчисляване на **остатък при деление** на 60 (за минутите) и остатък при деление на 24 (за часовете) с **оператора %**.

## 17. Еднакви 3 числа

Да се напише програма, която въвежда **три цели числа** и **печата дали са еднакви** или не. Примери:

вход	изход
5	no
6	
7	

вход	изход
5	yes
5	
5	

вход	изход
5	no
5	
4	

вход	изход
4	no
5	
5	

вход	изход
5	no
4	
4	

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#14>.

**Подсказка:** използвайте двойна проверка с **&&** (логическо „И“).

## 18. \*\* Изписване на число до 100 с думи

Напишете програма, която превръща **цяло число** в диапазона **[0...100]** в **текст на английски език**. Примери:

вход	изход
25	twenty five

вход	изход
12	twelve

вход	изход
99	ninety nine

вход	изход
50	fifty

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#15>.

**Подсказка:** разгледайте няколко случая с **if-else** проверки:

- Число в диапазона **[0...19]** → направете 20 проверки и отпечатайте числото с текст.
- Число в диапазона **[20...99]** → числото **num** отпечатайте на две части (лява и дясна цифра).
  - За **лявата цифра** (**tens = num / 10**) отпечатайте “twenty”, “thirty”, ..., “ninety”.
  - За **дясната цифра** (**ones = num % 10**) отпечатайте “”, “one”, “two”, ..., “nine”.
- Число **100** → отпечатайте „one hundred“.
- Друго** число → отпечатайте „invalid number“.