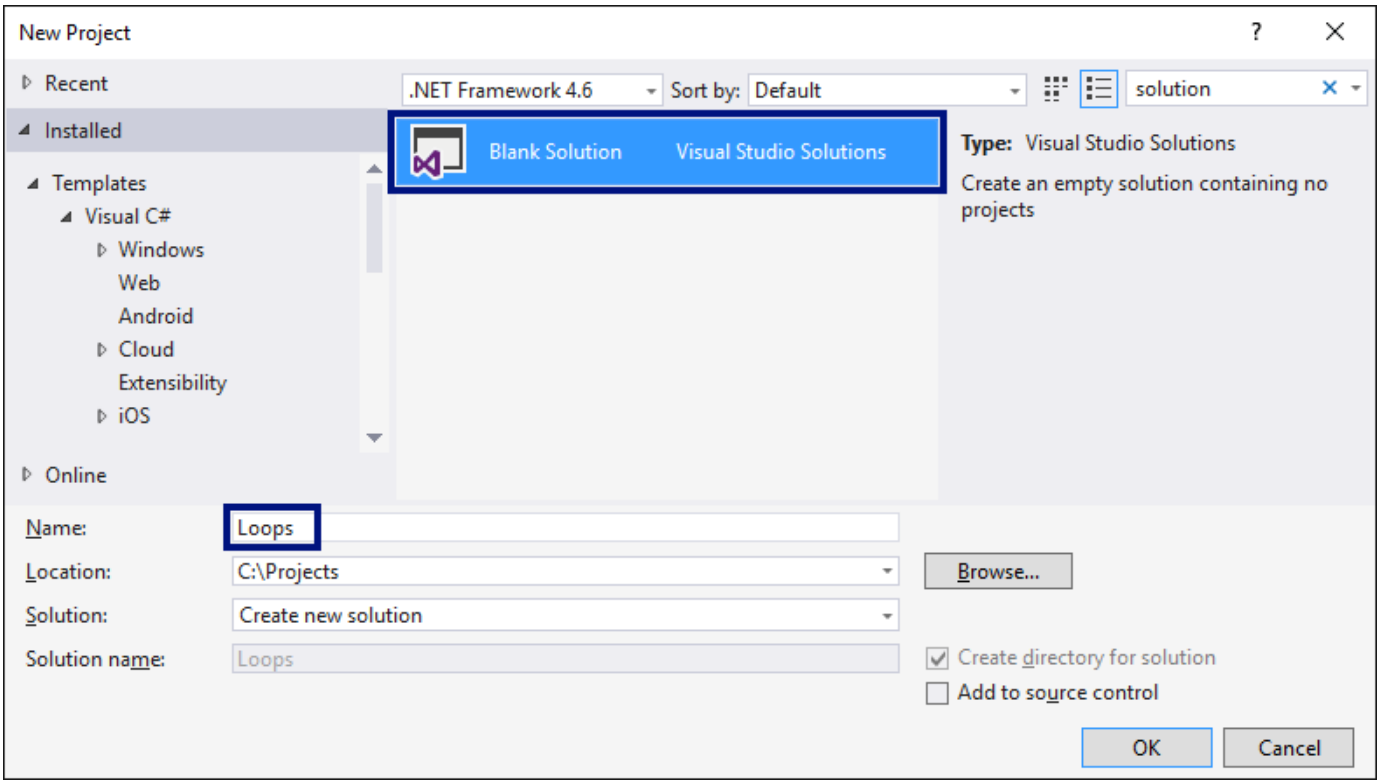
Упражнения: Повторения (цикли)

Задачи за упражнение в клас и за домашно към курса „Основи на програмирането“ @ СофтУни.

0. Празно Visual Studio решение (Blank Solution)

1. Създайте празно решение (**Blank Solution**) във Visual Studio за да организирате кода от задачите за упражнение. Целта на този **blank solution** е да съдържа **по един проект за всяка задача** от упражненията.



2. Задайте **да се стартира по подразбиране текущия проект** (не първият в решението). Кликнете с десен бутон на мишката върху **Solution 'Loops'** → **[Set StartUp Projects…]** → **[Current selection]**.

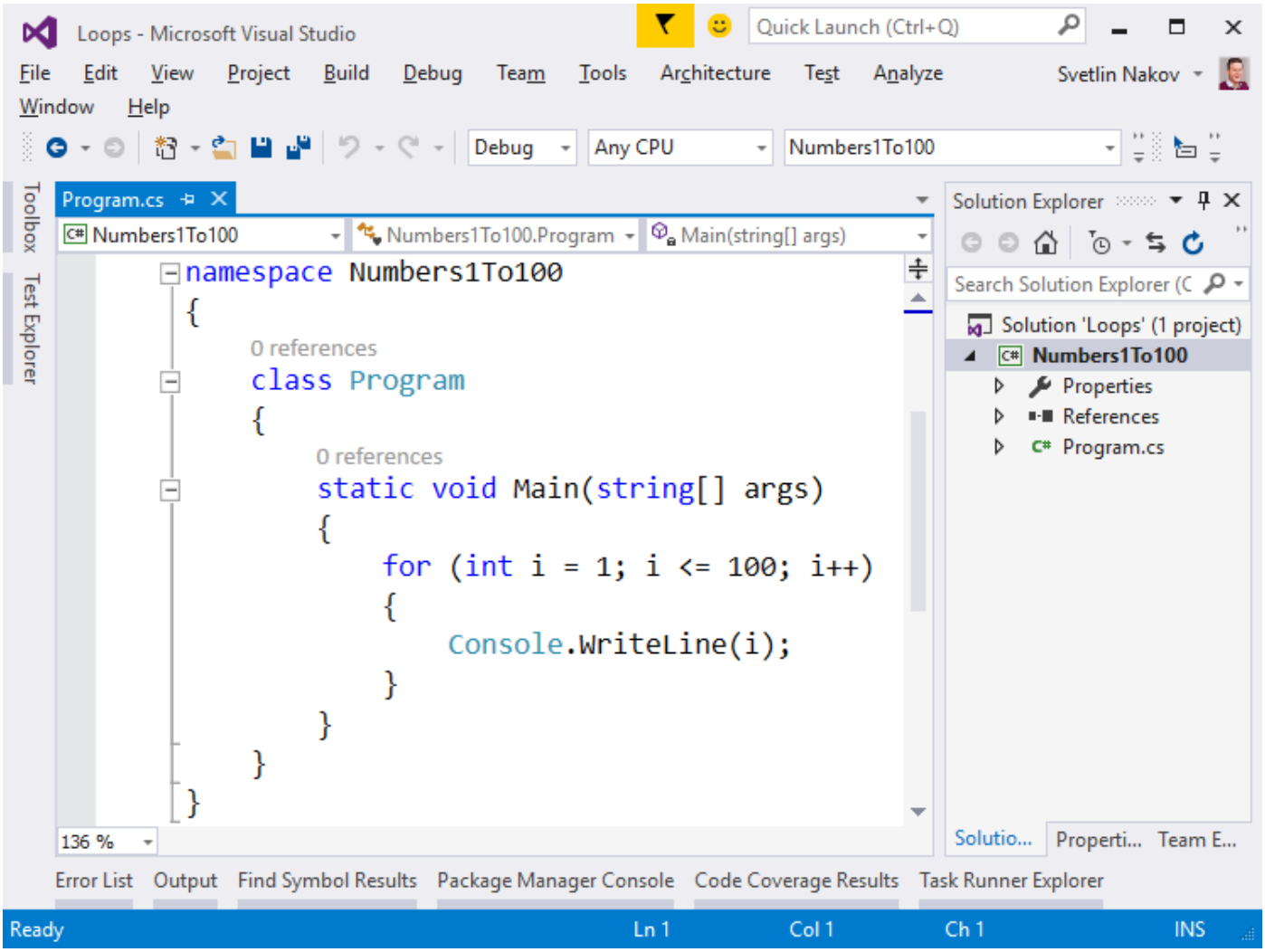
1. Числа от 1 до 100

Напишете програма, която отпечатва числата от 1 до 100, по едно на ред.

вход	изход
(няма)	1 2 3 … 98 99 100

Подсказки:

1. Създайте **нов проект** в съществуващото Visual Studio решение – конзолна C# програма. Задайте подходящо име на проекта, например “**`Numbers1To100`**”.
2. Отидете в тялото на метода **`Main(string[] args)`** и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:



3. **Стартирайте** програмата с [Ctrl+F5] и я **тествайте**:

4. **Тествайте** решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#0. Трябва да получите **100 точки** (напълно коректно решение).

2. Числа до 1000, завършващи на 7

Напишете програма, която отпечатва числата в диапазона **[1…1000]**, които **завършват на 7**.

вход	изход
(няма)	7 17 27 … 997

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#1.

Подсказка: можете да завъртите **for**-цикъл от 1 до 1000 и да проверите всяко число дали завършва на 7. Едно число **num** завършва на 7, когато **(num % 10 == 7)**.

3. Всички латински букви

Напишете програма, която отпечатва всички букви от латинската азбука: **a**, **b**, **c**, …, **z**.

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#2.

Подсказка: можете да завъртите **for**-цикъл от **'a'** до **'z'** (освен числа може да въртите в цикъл и букви).

4. Сумиране на числа

Да се напише програма, която **въвежда n цели числа и ги сумира**.

- От първия ред на входа се въвежда броят числа **n**.
- От следващите **n** реда се въвежда по едно цяло число.

Програмата трябва да прочете числата, да ги сумира и да отпечата сумата им. Примери:

вход	изход
2 10 20	30

вход	изход
3 -10 -20 -30	-60

вход	изход
4 45 -20 7 11	43

вход	изход
1 999	999

вход	изход
0	0

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#3.

Подсказки:

- Първо въведете едно число **n** (броят числа, които предстои да бъдат въведени).
- Инициализирайте **sum = 0** (в началото няма още прочетени числа, и съответно сумата е празна).
- В цикъл **n пъти** прочетете по едно цяло число **num** и го прибавете към сумата (**sum = sum + num**).
- Накрая в **sum** трябва да се е запазила сумата на прочетените числа. Отпечатайте я.

5. Най-голямо число

Напишете програма, която въвежда **n цели числа** (**n** > 0) и намира **най-голямото** измежду тях. Първо се въвежда броят числа **n**, а след това самите **n** числа, по едно на ред. Примери:

вход	изход
2 100 99	100

вход	изход
3 -10 20 -30	20

вход	изход
4 45 -20 7 99	99

вход	изход
1 999	999

вход	изход
2 -1 -2	-1

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#4.

Подсказки:

- Първо въведете едно число **n** (броят числа, които предстои да бъдат въведени).
- Въведете от конзолата първото число. Сложете текущият максимум **max** да е прочетеното число.

- В цикъл **n-1 пъти** прочетете по едно цяло число **num**. Ако прочетеното число **num** е по-голямо от текущият максимум **max**, запомнете **num** в **max**.
- Накрая в **max** трябва да се е запазило най-голямото число. Отпечатайте го.

6. Най-малко число

Напишете програма, която въвежда **n цели числа** (**n** > 0) и намира **най-малкото** измежду тях. Първо се въвежда броят числа **n**, а след това самите **n** числа, по едно на ред. Примери:

вход	изход
2 100 99	99

вход	изход
3 -10 20 -30	-30

вход	изход
4 45 -20 7 99	-20

вход	изход
1 999	999

вход	изход
2 -1 -2	-2

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#5.

Подсказки: задачата е абсолютно аналогична с предходната.

7. Лява и дясна сума

Да се напише програма, която въвежда **2*n** цели числа и проверява дали **сумата на първите n числа** (лява сума) е равна на **сумата на вторите n числа** (дясна сума). При равенство печата "**Yes**" + **сумата**; иначе печата "**No**" + **разликата**. Разликата се изчислява като положително число (по абсолютна стойност). Примери:

вход	изход	коментар
2 10 90 60 40	Yes, sum = 100	10+90 = 60+40 = 100

вход	изход	коментар
2 90 9 50 50	No, diff = 1	90+9 ≠ 50+50 Difference = |99-100| = 1

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#6.

Подсказки:

- Въведете **n**.
- Въведете първите **n** числа (**лявата** половина) и ги сумирайте.
- Въведете още **n** числа (**дясната** половина) и ги сумирайте.
- Изчислете **разликата** между сумите по абсолютна стойност: **Math.Abs(leftSum - rightSum)**.
- Ако разликата е **0**, отпечатайте "**Yes**" + **сумата**; иначе отпечатайте "**No**" + **разликата**.

8. Четна / нечетна сума

Да се напише програма, която въвежда **n** цели числа и проверява дали **сумата от числата на четни позиции** е равна на **сумата на числата на нечетни позиции**. При равенство да се отпечата "**Yes**" + **сумата**; иначе да се отпечата "**No**" + **разликата**. Разликата се изчислява по абсолютна стойност. Примери:

вход	изход	коментар
4 10 50	Yes Sum = 70	10+60 = 50+20 = 70

вход	изход	коментар
4 3 5	No Diff = 1	3+1 ≠ 5-2 Diff = |4-3| = 1

вход	изход	коментар
3 5 8	No Diff = 2	5+1 ≠ 8 Diff = |6-8| = 2

60 20			1 -2			1		

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#7.

Подсказки: Въведете числата едно по едно и изчислете двете **суми** (числа на **четни** позиции и числа на **нечетни** позиции). Както в предходната задача, изчислете абсолютна стойност на разликата и отпечатайте резултата ("**Yes**" + **сумата** при разлика 0 или "**No**" + **разликата** в противен случай).

9. Сумиране на гласните букви

Да се напише програма, която въвежда **текст** (стринг) и изчислява и отпечатва **сумата от стойностите на гласните букви** според таблицата по-долу:

буква	a	e	i	o	u
стойност	1	2	3	4	5

Примери:

вход	изход	коментар
hello	6	e + o = 2 + 4 = 6
hi	3	i = 3
bamboo	9	a + o + o = 1 + 4 + 4 = 9
beer	4	e + e = 2 + 2 = 4

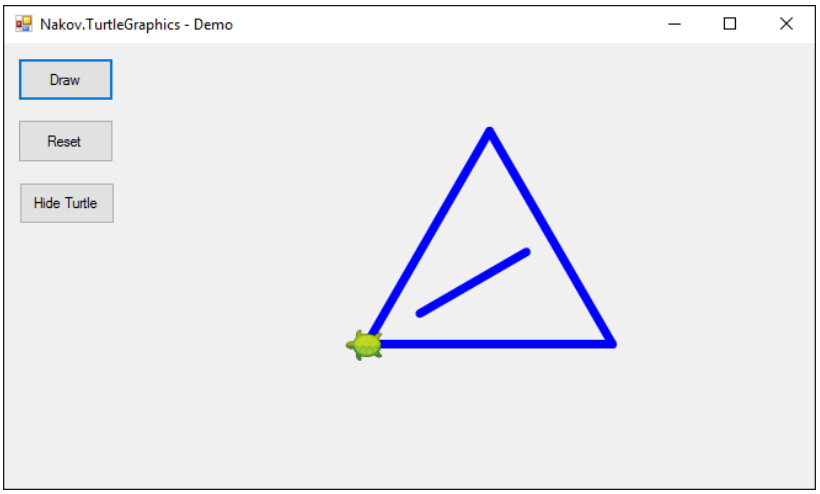
Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#8.

Подсказки:

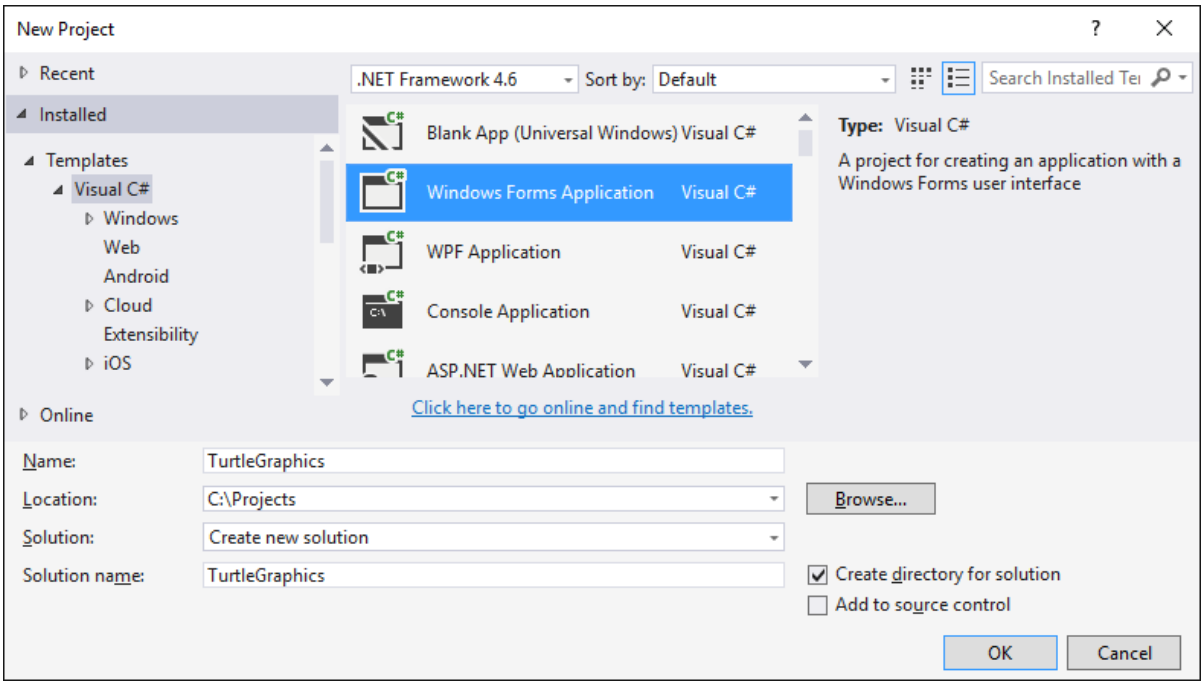
- Прочетете входния текст **`s`**. Нулирайте сумата.
- Завъртете цикъл от **`0`** до **`s.Length-1`** (дължината на текста -1).
- Проверете всяка буква **`s[i]`** дали е гласна и съответно добавете към сумата стойността ѝ.

10. Чертане с костенурка – графично GUI приложение

Целта на следващото упражнение е да си поиграем с една **библиотека за рисуване**, известна като "**графика с костенурка**" (**turtle graphics**). Ще изградим **графично приложение**, в което ще **рисуваме различни фигури**, придвижвайки нашата "**костенурка**" по екрана чрез операции от типа "отиди напред 100 позиции", "завърти се надясно на 30 градуса", "отиди напред още 50 позиции". Приложението ще изглежда приблизително така:



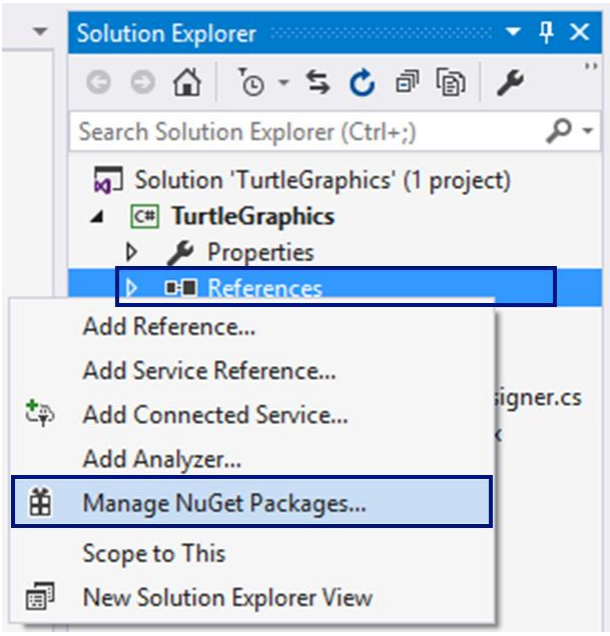
1. Запознайте се с **концепцията за рисуване** “**Turtle Graphics**”. Можете да разгледате следните източници:
   - Дефиниция на понятието “turtle graphics” – http://c2.com/cgi/wiki?TurtleGraphics
   - Статия за “turtle graphics” в Wikipedia – https://en.wikipedia.org/wiki/Turtle_graphics
   - Интерактивен онлайн инструмент за чертане с костенурка – https://blockly-games.appspot.com/turtle
2. Създайте нов **C# Windows Forms проект**:



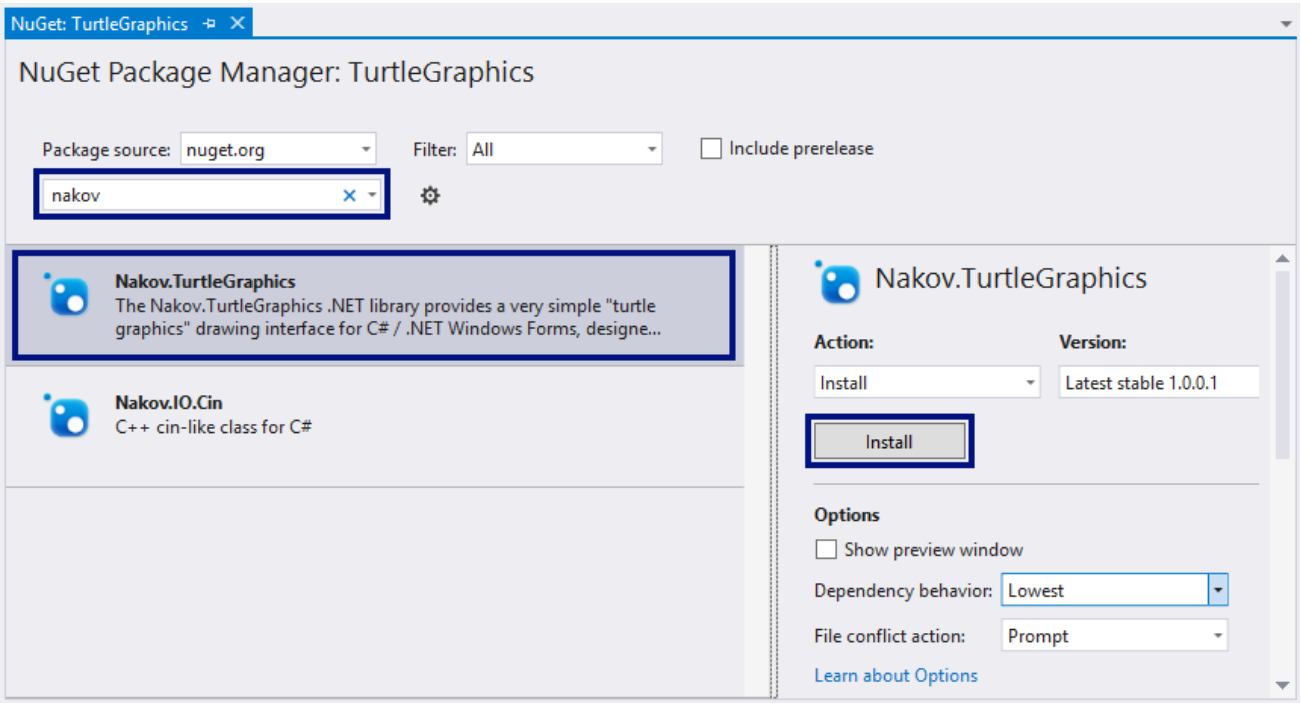
3. Инсталирайте **NuGet** пакета "**`Nakov.TurtleGraphics`**" към вашият Windows Forms проект.

От Visual Studio може да се добавят **външни библиотеки** (пакети) към вашите C# проекти. Те добавят допълнителна функционалност към вашите приложения. Официалното хранилище (repository) за C# библиотеки се поддържа от Microsoft и се нарича **NuGet** (www.nuget.org).

Кликнете върху проекта в **Solution Explorer** и изберете **[Manage NuGet Packages…]**:



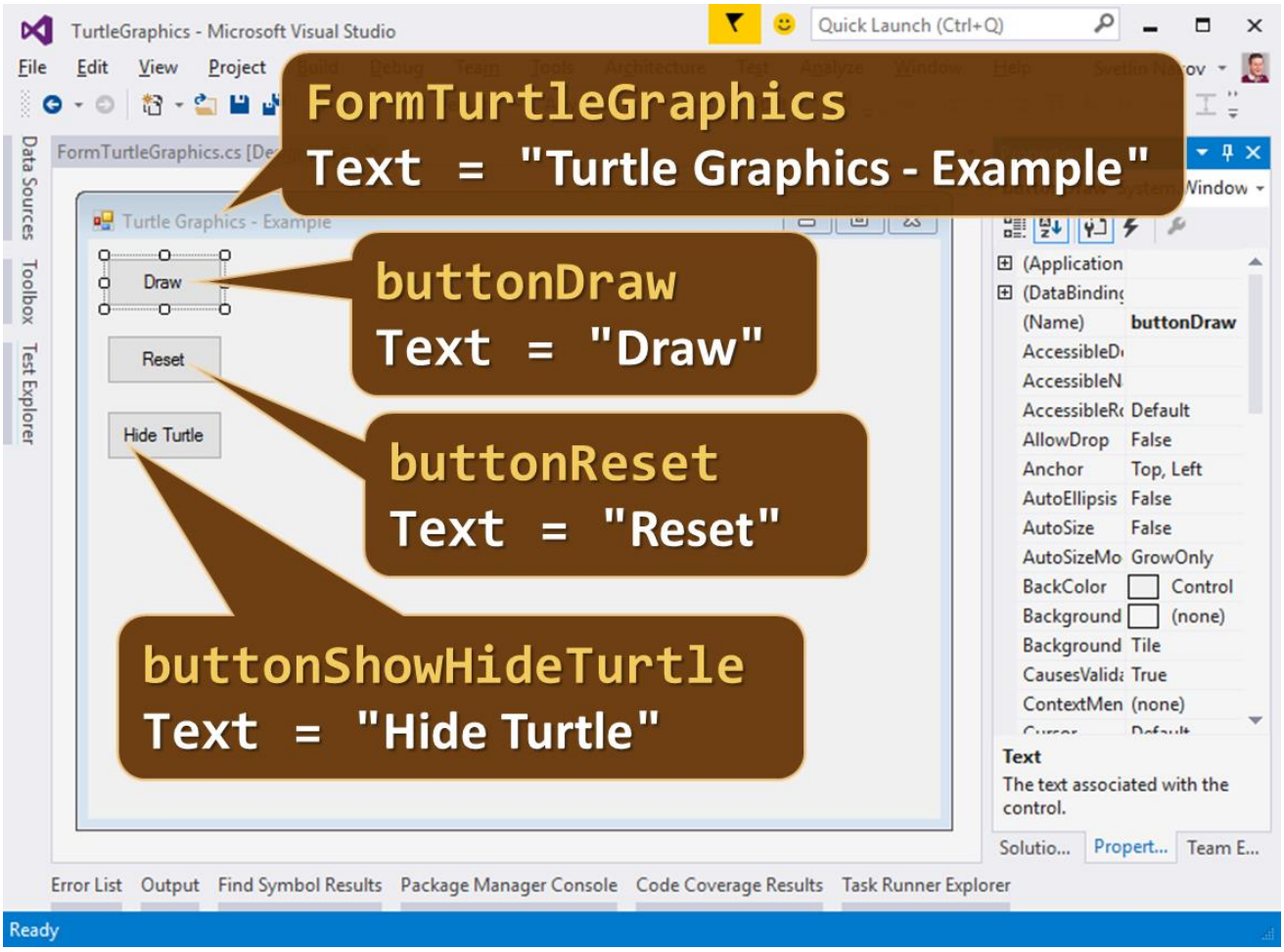
Ще се отвори прозорец за търсене и инсталиране на **NuGet** пакети. Потърсете пакети по ключова дума “**nakov**”. Ще излязат няколко пакета. Изберете пакет “**`Nakov.TurtleGraphics`**”. Натиснете **[Install]** за да го инсталирате към вашия C# проект:



Към вашият C# проект вече в включена външната библиотека "**`Nakov.TurtleGraphics`**". Тя дефинира един клас **`Turtle`**, с който представлява **костенурка за рисуване**. За да го използвате, трябва да добавите в C# кода за вашата форма (**`Form1.cs`**) следния код най-отгоре в началото на файла:

```
using Nakov.TurtleGraphics;
```

4. Сега сложете **три бутона** във формата и нагласете **имената** и **свойствата** им както е посочено по-долу:

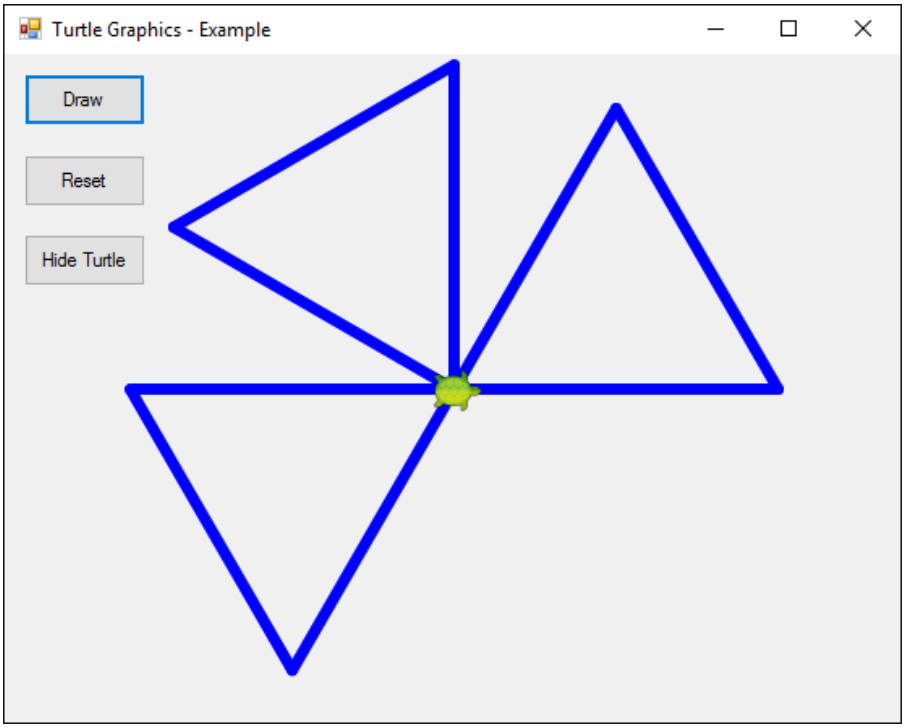


5. Кликнете два пъти върху бутона **[Draw]**, за да напишете кода, който да се изпълни при натискането му. Напишете следния код:

```
private void buttonDraw_Click(object sender, EventArgs e)
{
    Turtle.Rotate(30);
    Turtle.Forward(200);
    Turtle.Rotate(120);
    Turtle.Forward(200);
    Turtle.Rotate(120);
    Turtle.Forward(200);
}
```

Този код мести и върти костенурката, която в началото е в центъра на екрана (в средата на формата), и чертае равностранен триъгълник. Може да го редактирате и да си поиграете с него.

6. **Стартирайте** приложението с **[Ctrl+F5]**. Тествайте го дали работи (натиснете **[Draw]** бутона няколко пъти):



7. Сега можете да напишете **по-сложна програма за костенурката**:

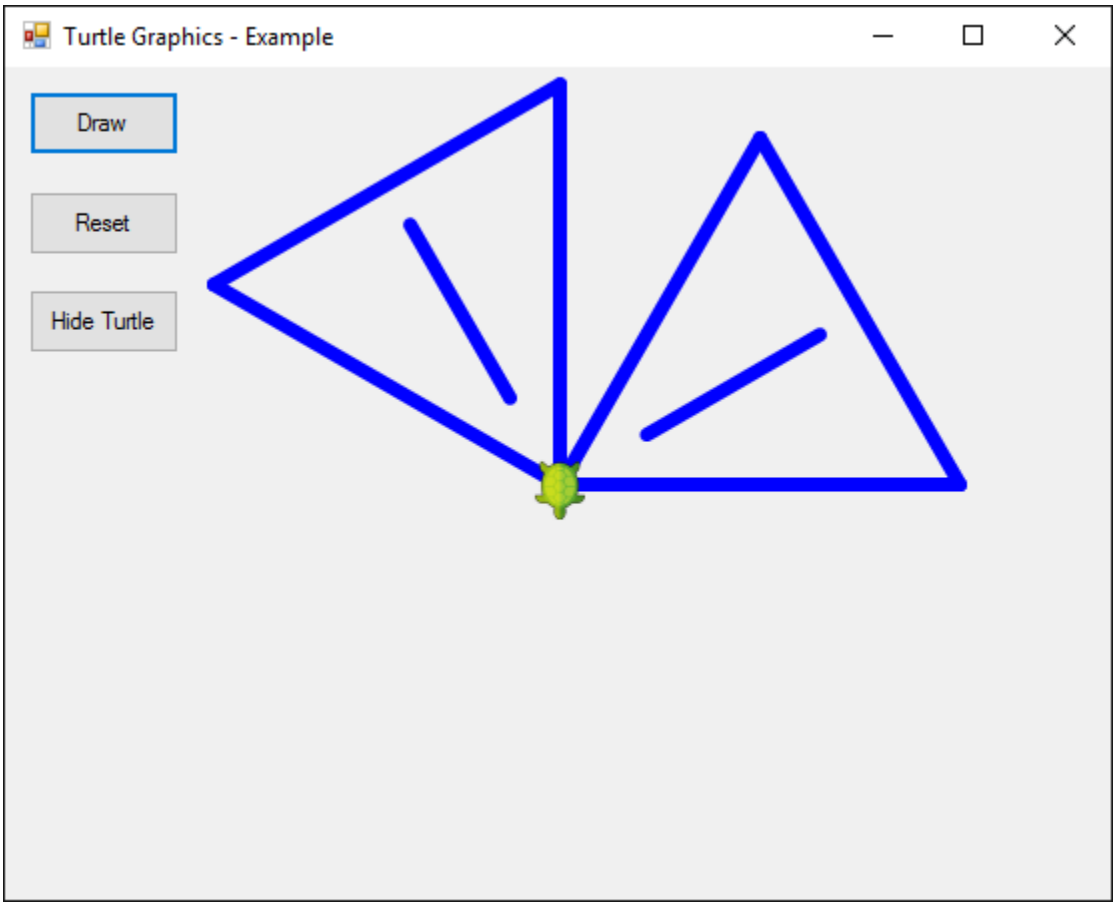
```
// Assign a delay to visualize the drawing process
Turtle.Delay = 200;

// Draw a equilateral triangle
Turtle.Rotate(30);
Turtle.Forward(200);
Turtle.Rotate(120);
Turtle.Forward(200);
Turtle.Rotate(120);
Turtle.Forward(200);

// Draw a line in the triangle
Turtle.Rotate(-30);
Turtle.PenUp();
Turtle.Backward(50);
Turtle.PenDown();
```

```
Turtle.Backward(100);
Turtle.PenUp();
Turtle.Forward(150);
Turtle.PenDown();
Turtle.Rotate(30);
```

8. **Стартирайте** приложението с **[Ctrl+F5]**. Тествайте дали работи новата програма за костенурката:



Вече костенурката чертае по-сложни фигури чрез приятно анимирано движение.

9. Напишете кода и за останалите два бутона. Целта на бутона **[Reset]** е да изтрие графиката и да започне да чертае на чисто:

```
private void buttonReset_Click(object sender, EventArgs e)
{
    Turtle.Reset();
}
```

10. Целта на бутона **[Show / Hide Turtle]** е да показва или скрива костенурката:

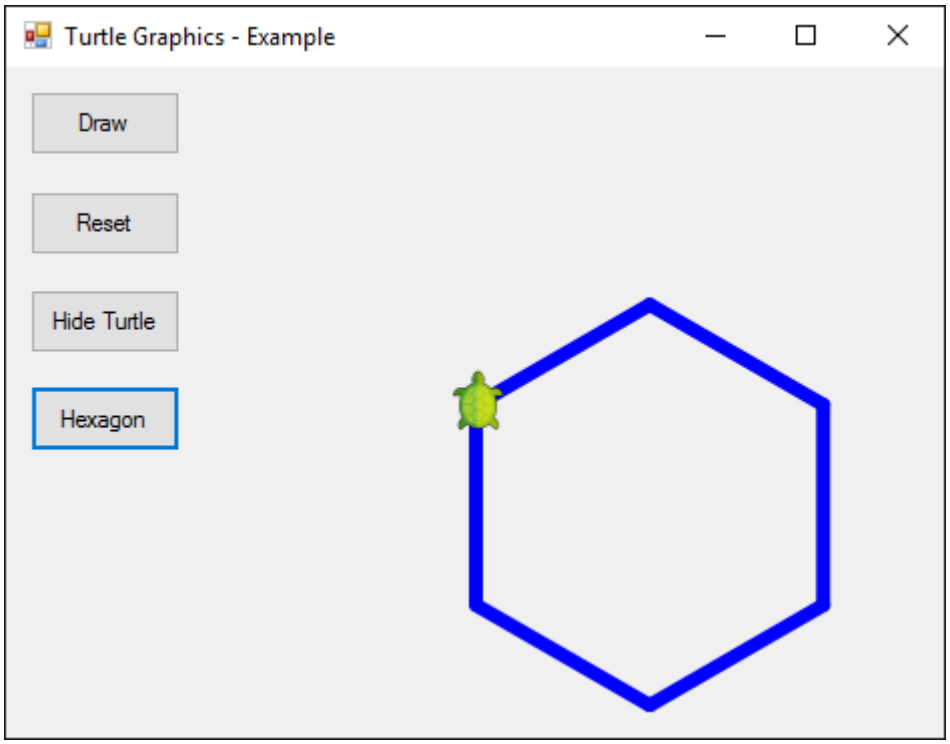
```
private void buttonShowHideTurtle_Click(object sender, EventArgs e)
{
    if (Turtle.ShowTurtle)
    {
        Turtle.ShowTurtle = false;
        this.buttonShowHideTurtle.Text = "Show Turtle";
    }
    else
    {
        Turtle.ShowTurtle = true;
        this.buttonShowHideTurtle.Text = "Hide Turtle";
```

```
    }
}
```

11. **Стартирайте** приложението с **[Ctrl+F5]**. Тествайте дали работят правилно всички бутони.

11. Чертане на шестоъгълник с костенурката

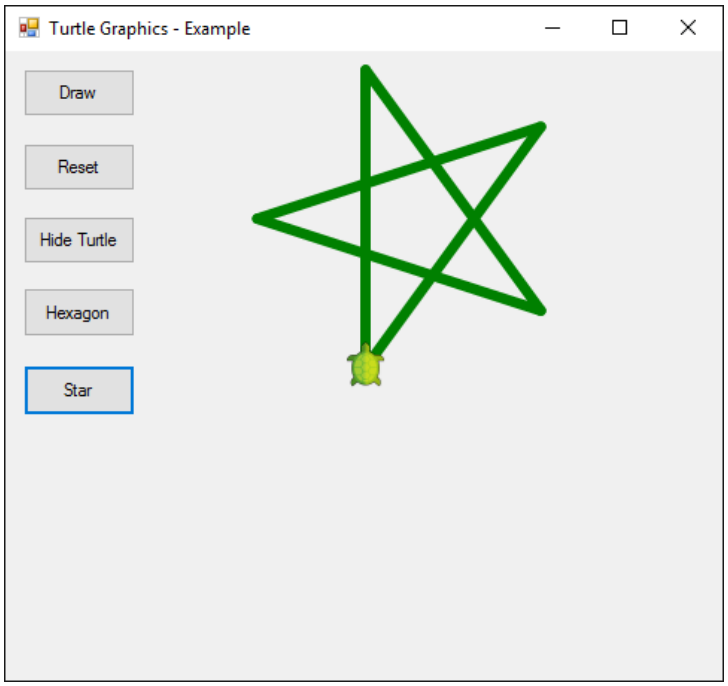
Добавете бутон **[Hexagon]**, който чертае правилен шестоъгълник:



Подсказка: В цикъл повторете 6 пъти следното: ротация на 60 градуса; движение напред 100.

12. Чертане на звезда с костенурката

Добавете бутон **[Star]**, който чертае звезда с 5 върха (**петолъчка**) като на фигурата по-долу:

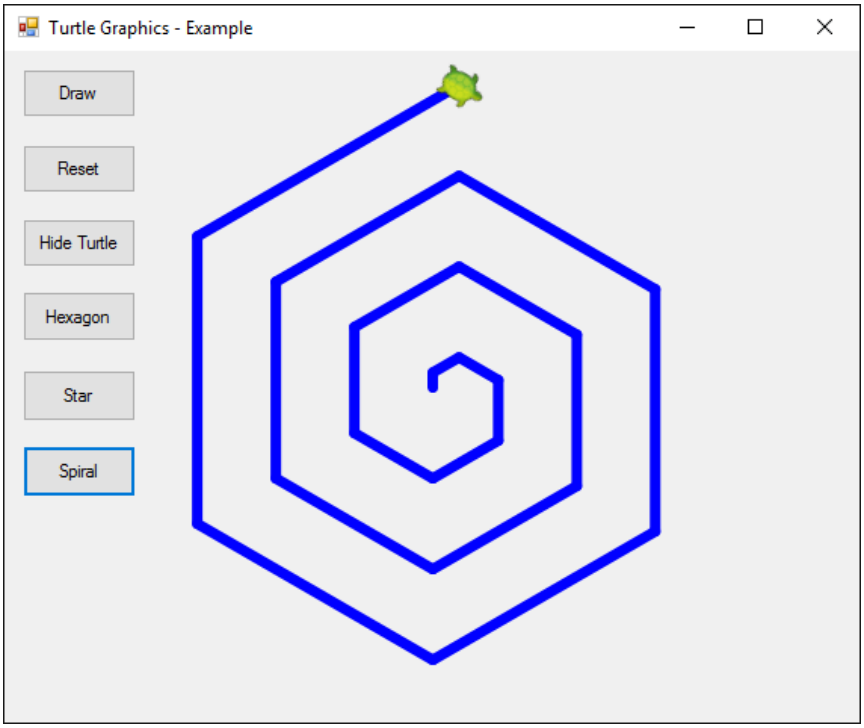


Подсказка:

- Сменете цвета: `Turtle.PenColor = Color.Green`.
- В цикъл повторете 5 пъти следното: движение напред 200, ротация на 144 градуса.

13. Чертане на спирала с костенурката

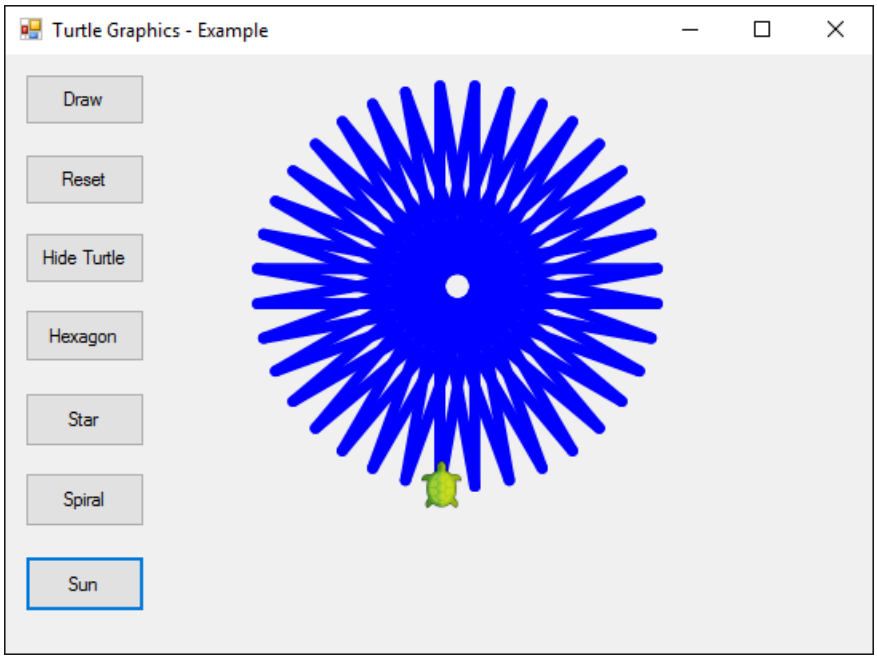
Добавете бутон **[Spiral]**, който чертае спирала с 20 върха като на фигурата по-долу:



Подсказка: Чертайте в цикъл като движите напред и завъртате. С всяка стъпка увеличавайте постепенно дължината на движението напред и завъртайте на 60 градуса.

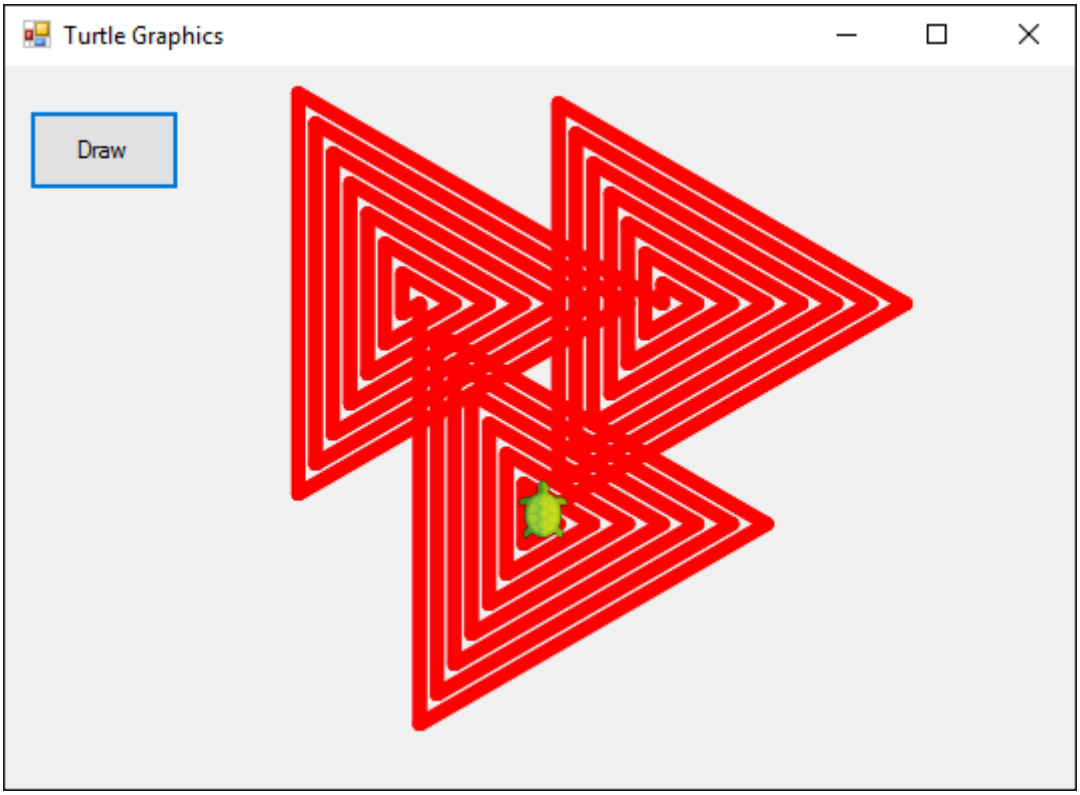
14. Чертане на слънце с костенурката

Добавете бутон **[Sun]**, който чертае слънце с 36 върха като на фигурата по-долу:



15. * Чертане на спирален триъгълник с костенурката

Добавете бутон **[Triangle]**, който три триъгълника с по 22 върха като на фигурата по-долу:



Подсказка: Чертайте в цикъл като движите напред и завъртате. С всяка стъпка увеличавайте с 10 дължината на движението напред и завъртайте на 120 градуса. Повторете 3 пъти за трите триъгълника.

16. * Елемент, равен на сумата на останалите

Да се напише програма, която въвежда **n** цели числа и проверява дали сред тях съществува число, което е равно на сумата на всички останали. Ако има такъв елемент, печата "**Yes**" + **неговата стойност**; иначе печата "**No**" + **разликата между най-големия елемент и сумата на останалите** (по абсолютна стойност). Примери:

вход	изход	коментари
7 3 4 1 1 2 **12** 1	Yes Sum = 12	3 + 4 + 1 + 2 + 1 + 1 = 12
4 **6** 1 2 3	Yes Sum = 6	1 + 2 + 3 = 12
3 1 1	No Diff = 8	|10 - (1 + 1)| = 8

10		
3 5 5 1	No Diff = 1	|5 - (5 + 1)| = 1
3 1 1 1	No Diff = 1	

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#9.

Подсказка: изчислете **сумата** на всички елементи и **най-големият** от тях и проверете търсеното условие.

17. * Четни / нечетни позиции

Напишете програма, която чете **n числа** и пресмята **сумата**, **минимума** и **максимума** на числата на **четни** и **нечетни** позиции (броим от 1). Когато няма минимален / максимален елемент, отпечатайте "**No**". Примери:

вход	изход
6 **2** 3 **5** 4 **2** 1	OddSum=9, OddMin=2, OddMax=5, EvenSum=8, EvenMin=1, EvenMax=4

вход	изход
2 **1.5** -2.5	OddSum=1.5, OddMin=1.5, OddMax=1.5, EvenSum=-2.5, EvenMin=-2.5, EvenMax=-2.5

вход	изход
1 **1**	OddSum=1, OddMin=1, OddMax=1, EvenSum=0, EvenMin=No, EvenMax=No

вход	изход
0	OddSum=0, OddMin=No, OddMax=No, EvenSum=0, EvenMin=No, EvenMax=No

вход	изход
5 **3** -2 **8** 11 **-3**	OddSum=8, OddMin=-3, OddMax=8, EvenSum=9, EvenMin=-2, EvenMax=11

вход	изход
4 1.5 **1.75** 1.5 **1.75**	OddSum=3, OddMin=1.5, OddMax=1.5, EvenSum=3.5, EvenMin=1.75, EvenMax=1.75

вход	изход
1 **-5**	OddSum=-5, OddMin=-5, OddMax=-5, EvenSum=0, EvenMin=No, EvenMax=No

вход	изход
3 **-1** -2 **-3**	OddSum=-4, OddMin=-3, OddMax=-1, EvenSum=-2, EvenMin=-2, EvenMax=-2

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#10.

Подсказки:

- Задача обединява няколко предходни задачи: намиране на **минимум**, намиране на **максимум**, намиране на **сума** и обработка на елементите от **четни и нечетни позиции**. Припомнете си ги.
- Работете с **дробни числа** (не цели). Сумата, минимумът и максимумът също са дробни числа.
- Използвайте **неутрална начална стойност** при намиране на минимум / максимум, например **1000000000.0** и **-1000000000.0**. Ако получите накрая неутралната стойност, печатайте "**No**".

18. * Еднакви двойки

Дадени са 2***n** числа. Първото и второто формират **двойка**, третото и четвъртото също и т.н. Всяка двойка има **стойност** – сумата от съставящите я числа. Напишете програма, която проверява **дали всички двойки имат еднаква стойност** или печата **максималната разлика** между две последователни двойки. Ако всички двойки

имат еднаква стойност, отпечатайте "**Yes, value=…**" + **стойността**. В противен случай отпечатайте "**No, maxdiff=…**" + **максималната разлика**. Примери:

вход	изход	коментари
3 1 2 0 3 4 -1	Yes, value=3	стойности = {3, 3, 3} еднакви стойности
4 1 1 3 1 2 2 0 0	No, maxdiff=4	стойности = {2, 4, 4, 0} разлики = {2, 0, 4} макс. разлика = 4
2 -1 0 0 -1	Yes, value=-1	стойности = {-1, -1} еднакви стойности

вход	изход	коментари
2 1 2 2 2	No, maxdiff=1	стойности = {3, 4} разлики = {1} макс. разлика = 1
1 5 5	Yes, value=10	стойности = {10} една стойност еднакви стойности
2 -1 2 0 -1	No, maxdiff=2	стойности = {1, -1} разлики = {2} макс. разлика = 2

Тествайте решението си в **judge системата**: https://judge.softuni.bg/Contests/Practice/Index/154#11.

Подсказки:

- Прочитайте входните числа **по двойки**. За всяка двойка пресмятайте **сумата**.
- Докато четете входните двойки, за всяка двойка без първата пресмятайте **разликата с предходната**. За целта пазете в отделна променлива сумата на предходната двойка.
- Намерете **най-голямата разлика** между две двойки. Ако е **0**, печатайте "**Yes**" иначе "**No**" + разликата.