



## Модул 1. Увод в програмирането

### Тема 1. Въведение в програмирането

#### Задача 1.1. Конзолна програмка "Hello World!"

Напишете програма, която отпечата на екрана текста Hello World!

##### *Решение*

```
namespace HelloWorld
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

### Тема 2. Пресмятания оператори изрази

#### Задача 2.1. Пресмятане на лице на квадрат

Да се напише конзолна програма, която прочита цяло число „a“, въведено от потребителя, и пресмята лицето на квадрат със страна „a“. Задачата е тривиално лесна: въвеждате число от конзолата, умножавате го само по себе си и печатате получения резултат на конзолата.

##### *Решение*

```
namespace SquareArea
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("a=");
            var a = int.Parse(Console.ReadLine());
            var area = a * a;
            Console.Write("Area = ");
            Console.WriteLine(area);
        }
    }
}
```

#### Задача 2.2. От инчове към сантиметри

Да се напише програма, която чете от конзолата число (не непременно цяло), въведено от потребителя, и преобразува числото от инчове в сантиметри. За целта умножава инчовете по 2.54 (защото 1 инч = 2.54 сантиметра)..

##### *Решение*

```
namespace InchesToCentimeters
```



```
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Inches=");
            var inches = double.Parse(Console.ReadLine());
            var centimeters = inches * 2.54;
            Console.Write("Centimeters = ");
            Console.WriteLine(centimeters);
        }
    }
}
```

### Задача 2.3. Поздрав по име

Да се напише програма, която чете от конзолата име на човек, въведено от потребителя, и отпечатва "Hello, <name>!", където <name> е въведеното преди това име.

#### Решение

```
namespace Greetings
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter your name: ");
            var name = Console.ReadLine();
            Console.WriteLine("Hello, {0}!", name);
        }
    }
}
```

### Задача 2.4. Съединяване на текст и числа

Напишете програма, която прочита от конзолата име, фамилия, възраст и град, въведени от потребителя, и печата съобщение от следния вид: "You are <firstName> <lastName>, a <age>-years old person from <town>".

#### Решение

```
namespace LongText
{
    internal class Program
    {
        static void Main(string[] args)
        {
            var firstName = Console.ReadLine();
            var lastName = Console.ReadLine();
            var age = int.Parse(Console.ReadLine());
            var town = Console.ReadLine();
            Console.WriteLine($"You are {firstName} {lastName}, a {age}-years old
person from {town}.");
        }
    }
}
```



### Задача 2.5. Конвертор от °C към °F

Напишете програма, която чете градуси по скалата на Целзий (°C), въведени от потребителя, и ги преобразува до градуси по скалата на Фаренхайт (°F). Потърсете в Интернет подходяща [формула](#), с която да извършите изчисленията. Примери:

ВХОД	ИЗХОД
25	77

ВХОД	ИЗХОД
0	32

ВХОД	ИЗХОД
-5.5	22.1

ВХОД	ИЗХОД
32.3	90.14

#### Решение

```
namespace CelsiusToFahrenheit
```

```
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.Write("Celsius = ");  
            var Celsius = double.Parse(Console.ReadLine());  
            var Fahrenheit = 1.8 * Celsius + 32;  
            Console.WriteLine("Fahrenheit = {0}", Fahrenheit);  
        }  
    }  
}
```

### Задача 2.6. Конвертор от радиани в градуси

Напишете програма, която чете ъгъл в [радиани](#) (rad), въведен от потребителя, и го преобразува в [градуси](#) (deg). Потърсете в Интернет подходяща формула. Числото  $\pi$  в програми е достъпно чрез Math.PI. Закръглете резултата до най-близкото цяло число използвайки Math.Round(). Примери:

ВХОД	ИЗХОД
3.1416	180

ВХОД	ИЗХОД
6.2832	360

ВХОД	ИЗХОД
0.7854	45

ВХОД	ИЗХОД
0.5236	30

#### Решение

```
namespace RadiansToDegrees
```

```
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.Write("Radians = ");  
            var Radians = double.Parse(Console.ReadLine());  
            var Degrees = (Radians * 180) / Math.PI;  
            Console.WriteLine("Degrees = {0}", Degrees);  
        }  
    }  
}
```



## Задача 2.7. Конвертор от USD към BGN

Напишете програма за конвертиране на щатски долари (USD) в български лева (BGN). Закръглете резултата до 2 цифри след десетичната точка. Използвайте фиксиран курс между долар и лев: 1 USD = 1.79549 BGN.

ВХОД	ИЗХОД
20	35.91 BGN

ВХОД	ИЗХОД
100	179.55 BGN

ВХОД	ИЗХОД
12.5	22.44 BGN

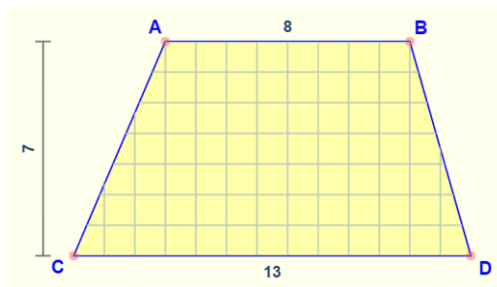
### Решение

```
namespace UsdToBgn
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("USD = ");
            var USD = double.Parse(Console.ReadLine());
            var BGN = USD * 1.79549;
            Console.WriteLine("BGN = {0}", BGN);
        }
    }
}
```

## Задача 2.8. Лице на трапец

Напишете програма, която чете от конзолата три числа b1, b2 и h, въведени от потребителя, и пресмята лицето на трапец с основи b1 и b2 и височина h. Формулата за лице на трапец е:  $(b1 + b2) * h / 2$ .

На фигурата е показан трапец със страни 8 и 13 и височина 7. Той има лице  $(8 + 13) * 7 / 2 = 73.5$ .



### Решение

```
namespace TrapecoidArea
{
    internal class Program
    {
        static void Main(string[] args)
        {
            var b1 = double.Parse(Console.ReadLine());
            var b2 = double.Parse(Console.ReadLine());
            var h = double.Parse(Console.ReadLine());
            var Area = (b1 + b2) * h / 2;
            Console.WriteLine("Trapezoid Area = {0}", Area);
        }
    }
}
```



### Задача 2.9. Периметър и лице на кръг

Напишете програма, която чете от конзолата число  $r$ , въведено от потребителя, и пресмята и отпечатва лицето и периметъра на кръг / окръжност с радиус  $r$ . Закръглете резултата до 2 знака след десетичната точка, използвайки `Math.Round()`.

ВХОД	ИЗХОД
3	Area = 28.27 Perimeter = 18.85
4.5	Area = 63.62 Perimeter = 28.27

За изчисленията можете да използвате следните формули:

- $area = Math.PI * r * r$
- $perimeter = 2 * Math.PI * r$

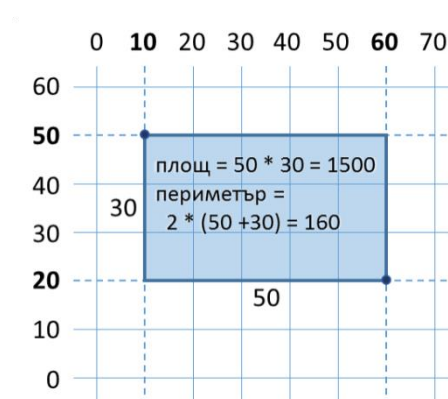
#### Решение

```
namespace CircleAreaAndPerimeter
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Radius = ");
            var r = double.Parse(Console.ReadLine());
            var area = Math.PI * r * r;
            var perimeter = 2 * Math.PI * r;
            Console.WriteLine("Area = {0}", area);
            Console.WriteLine("Perimeter = {0}", perimeter);
        }
    }
}
```

### Задача 2.10. Лице на правоъгълник в равнината

Правоъгълник е зададен с координатите на два от своите срещуположни ъгъла ( $x_1, y_1$ ) – ( $x_2, y_2$ ). Да се пресметнат площта и периметъра му. Входът се въвежда от потребителя. Числата  $x_1, y_1, x_2$  и  $y_2$  са дадени по едно наред. Изходът се извежда на конзолата и трябва да съдържа два реда с по една число на всеки от тях – лицето и периметъра.

ВХОД	ИЗХОД
60	1500
20	160





10 50	
30 40 70 -10	2000 180
600.25 500.75 100.50 -200.5	350449.6875 2402

#### Решение

`namespace RectangleArea`

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var x1 = double.Parse(Console.ReadLine());  
            var y1 = double.Parse(Console.ReadLine());  
            var x2 = double.Parse(Console.ReadLine());  
            var y2 = double.Parse(Console.ReadLine());  
            var h = Math.Max(y1, y2) - Math.Min(y1, y2);  
            var w = Math.Max(x1, x2) - Math.Min(x1, x2);  
            var s = h * w;  
            var p = 2 * (h + w);  
            Console.WriteLine(s);  
            Console.WriteLine(p);  
        }  
    }  
}
```

#### Задача 2.11. Лице на триъгълник

Напишете програма, която чете от конзолата страна и височина на триъгълник, въведени от потребителя, и пресмята неговото лице. Използвайте формулата за лице на триъгълник:  $area = a * h / 2$ . Закръглете резултата до 2 знака след десетичната точка.

ВХОД	ИЗХОД
20 30	Triangle area = 300
15 35	Triangle area = 262.5
7.75 8.45	Triangle area = 32.74



1.23456 4.56789	Triangle area = 2.82
--------------------	----------------------

#### Решение

```
namespace TriangleArea
{
    class Program
    {
        static void Main(string[] args)
        {
            var a = double.Parse(Console.ReadLine());
            var h = double.Parse(Console.ReadLine());
            var s = Math.Round(a * h / 2, 2);
            Console.WriteLine("Triangle area = {0}", s);
        }
    }
}
```

#### Задача 2.12. Междувалутен конвертор

Напишете програма за конвертиране на парична сума от една валута в друга. Трябва да се поддържат следните валути: BGN, USD, EUR, GBP. Използвайте следните фиксирани валутни курсове:

Курс	USD	EUR	GBP
1 BGN	1.79549	1.95583	2.53405

Входът е сума за конвертиране, входна валута, изходна валута, въведени от потребителя. Изходът е едно число – преобразуваната сума по посочените по-горе курсове, закръглен до 2 цифри след десетичната точка. Примери:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
20 USD BGN	35.91 BGN	100 BGN EUR	51.13 EUR	12.35 EUR GBP	9.53 GBP	150.35 USD EUR	138.02 EUR

#### Решение

```
namespace CurrencyConvertor
{
    class Program
    {
        static void Main(string[] args)
        {
            var s = double.Parse(Console.ReadLine());
            var a = Console.ReadLine();
            var b = Console.ReadLine();
            if (a == "BGN" && b == "USD") Console.WriteLine("{0:f2} USD", s /
1.79549);
            if (a == "BGN" && b == "EUR") Console.WriteLine("{0:f2} EUR", s /
1.95583);
        }
    }
}
```



```
2.53405);  
    if (a == "BGN" && b == "GBP") Console.WriteLine("{0:f2} GBP", s /  
1.79549);  
    if (a == "USD" && b == "BGN") Console.WriteLine("{0:f2} BGN", s *  
1.95583);  
    if (a == "GBP" && b == "BGN") Console.WriteLine("{0:f2} BGN", s *  
2.53405);  
    if (a == "USD" && b == "EUR") Console.WriteLine("{0:f2} EUR", (s *  
1.79549) / 1.95583);  
    if (a == "USD" && b == "GBP") Console.WriteLine("{0:f2} GBP", (s *  
1.79549) / 2.53405);  
    if (a == "EUR" && b == "USD") Console.WriteLine("{0:f2} EUR", (s *  
1.95583) / 1.79549);  
    if (a == "EUR" && b == "GBP") Console.WriteLine("{0:f2} GBP", (s *  
1.95583) / 2.53405);  
    if (a == "GBP" && b == "USD") Console.WriteLine("{0:f2} EUR", (s *  
2.53405) / 1.79549);  
    if (a == "GBP" && b == "EUR") Console.WriteLine("{0:f2} GBP", (s *  
2.53405) / 1.95583);  
    }  
}  
}
```

### Задача 2.13. 1000 дни на Земята

Напишете програма, която чете рождена дата във формат "dd-MM-yyyy", въведена от потребителя, и пресмята датата, на която се навършват 1000 дни от тази рождена дата и я отпечатва в същия формат.

ВХОД	ИЗХОД
25-02-1995	20-11-1997
07-11-2003	02-08-2006
30-12-2002	24-09-2005
01-01-2012	26-09-2014
14-06-1980	10-03-1983

*Подсказки: потърсете информация за типа DateTime в C# и по-конкретно разгледайте методите ParseExact(str, format), AddDays(count) и ToString(format). С тяхна помощ може да решите задачата, без да е необходимо да изчислявате дни, месеци и високосни години.*

### Решение

```
namespace ThousandDaysOnEarth  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            string date = Console.ReadLine();
```

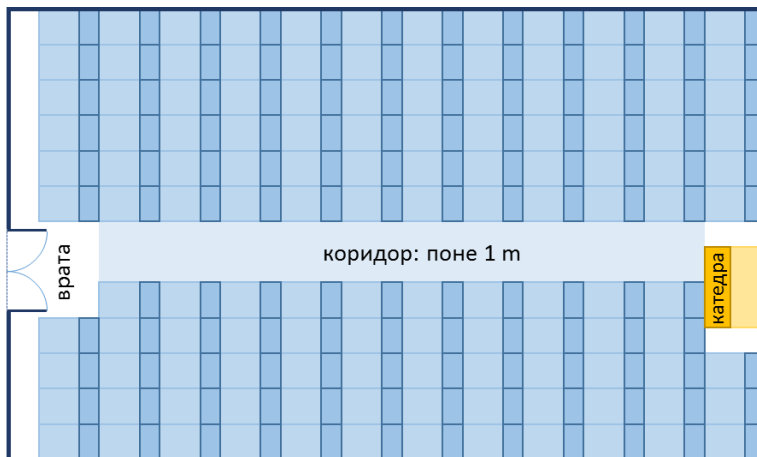




```
DateTime time = DateTime.ParseExact(date, "dd-MM-yyyy",  
    System.Globalization.CultureInfo.InvariantCulture);  
time = time.AddDays(999);  
Console.WriteLine(time.ToString("dd-MM-yyyy"));  
}  
}
```

### Задача 2.14. Учебна зала

Учебна зала има правоъгълен размер  $w$  на  $h$  метра, без колони във вътрешността си. Залата е разделена на две части – лява и дясна, с коридор приблизително по средата. В лявата и в дясната част има редици с бюра. В задната част на залата има голяма входна врата. В предната част на залата има катедра с подиум за преподавателя.



Едно работно място заема 70 на 120 cm (маса с размер 70 на 40 cm + място за стол и преминаване с размер 70 на 80 cm). Коридорът е широк поне 100 cm. Изчислено е, че заради входната врата (която е с отвор 160 cm) се губи точно 1 работно място, а заради катедрата (която е с размер 160 на 120 cm) се губят точно 2 работни места. Напишете програма, която прочита размерите на учебната зала и изчислява броя работни места в нея при описаното разположение (вж. фигурата).

#### Вход

От конзолата се четат 2 числа, по едно на ред:  $h$  (дължина в метри) и  $w$  (широчина в метри), въведени от потребителя.

Ограничения:  $3 \leq h \leq w \leq 100$ .


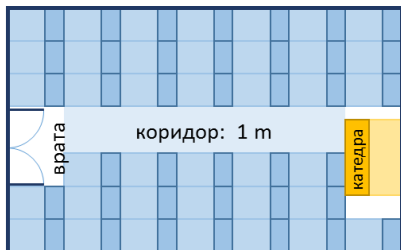
#### Изход

Да се отпечата на конзолата едно цяло число: броят места в учебната зала.

#### Примерен вход и изход

Вход	Изход	Чертеж	Обяснения
------	-------	--------	-----------



15 8.9	129		<p>Залата е дълга 1500 см. В тях могат да бъдат разположени 12 реда (<math>12 * 120 \text{ cm} = 1440 + 60 \text{ cm}</math> остатък).</p> <p>Залата е широка 890 см. От тях 100 см отиват за коридора в средата. В останалите 790 см могат да се разположат по 11 бюра на ред (<math>11 * 70 \text{ cm} = 770 \text{ cm} + 20 \text{ cm}</math> остатък).</p> <p>Брой места = <math>12 * 11 - 3 = 132 - 3 = 129</math> (имаме 12 реда по 11 места = 132 минус 3 места за катедра и входна врата).</p>
8.4 5.2	39		<p>Залата е дълга 840 см. В тях могат да бъдат разположени 7 реда (<math>7 * 120 \text{ cm} = 840</math>, без остатък).</p> <p>Залата е широка 520 см. От тях 100 см отиват за коридора в средата. В останалите 420 см могат да се разположат по 6 бюра на ред (<math>6 * 70 \text{ cm} = 420 \text{ cm}</math>, без остатък).</p> <p>Брой места = <math>7 * 6 - 3 = 42 - 3 = 39</math> (имаме 7 реда по 6 места = 42 минус 3 места за катедра и входна врата).</p>

### Решение

namespace TrainingLab

{

class Program

{

static void Main(string[] args)

{

// Входни данни

var w = double.Parse(Console.ReadLine()) \* 100;

var h = double.Parse(Console.ReadLine()) \* 100;

// Математически сметки

var w1 = (int)w / 120;

var h1 = ((int)h - 100) / 70;

var r = w1 \* h1 - 3;

// Изходни данни

Console.WriteLine(r);

}

}

}



### Задача 2.15. Зеленчукова борса

Градинар продавал реколтата от градината си на зеленчуковата борса. Продава зеленчуци за N лева на килограм и плодове за M лева за килограм. Напишете програма, която да пресмята приходите от реколтата в евро (ако приемем, че едно евро е равно на 1.94лв).

#### Вход

От конзолата се четат 4 числа, по едно на ред, въведени от потребителя:

- Първи ред – Цена за килограм зеленчуци – число с плаваща запетая
- Втори ред – Цена за килограм плодове – число с плаваща запетая
- Трети ред – Общо килограми на зеленчуците – цяло число
- Четвърти ред – Общо килограми на плодовете – цяло число

Ограничения: Всички числа ще са в интервала от 0.00 до 1000.00

#### Изход

Да се отпечата на конзолата едно число с плаваща запетая: приходите от всички плодове и зеленчуци в евро. Резултатът да се форматира до втория знак след запетаята.

#### Примерен вход и изход

Вход	Изход	Обяснения
0.194 19.4 10 10	101.00	Зеленчуците струват – 0.194лв. * 10кг. = 1.94лв. Плодовете струват – 19.4лв. * 10кг. = 194лв. Общо – 195.94лв. = 101евро
1.5 2.5 10 10	20.62	

#### Решение

```
namespace VegetableMarket
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            var VegetablesKgPrice = double.Parse(Console.ReadLine());
```

```
            var FruitsKgPrice = double.Parse(Console.ReadLine());
```

```
            var VegetablesKg = int.Parse(Console.ReadLine());
```

```
            var FruitsKg = int.Parse(Console.ReadLine());
```

```
            var Vegetables = (VegetablesKgPrice * VegetablesKg);
```



```
var Fruits = (FruitsKgPrice * FruitsKg);  
var WholePrice = (Vegetables + Fruits);  
  
Console.WriteLine(WholePrice / 1.94);  
}  
}
```

### Задача 2.16. Ремонт на плочки

На площадката пред жилищен блок трябва да се поставят плочки. Площадката е с форма на квадрат със страна N метра. Плочките са широки „W“ метра и дълги „L“ метра. На площадката има една пейка с ширина M метра и дължина O метра. Под нея не е нужно да се слагат плочки. Всяка плочка се поставя за 0.2 минути.

Напишете програма, която чете от конзолата размерите на площадката, плочките и пейката и пресмята колко плочки са необходими да се покрие площадката и пресмята времето за поставяне на всички плочки.

Пример: площадка с размер 20м. има площ 400кв.м. Пейка широка 1м. и дълга 2м., заема площ 2кв.м. Една плочка е широка 5м. и дълга 4м. има площ = 20кв.м. Площта която трябва да се покрие е  $400 - 2 = 398$  кв.м. Необходими са  $398 / 20 = 19.90$  плочки. Необходимото време е  $19.90 * 0.2 = 3.98$  минути.

#### Вход

От конзолата се четат 5 числа, въведени от потребителя:

- N – дължината на страна от площадката в интервала [1...100]
- W – широчината на една плочка в интервала [0.1...10.00]
- L – дължината на една плочка в интервала [0.1...10.00]
- M – широчината на пейката в интервала [0...10]
- O – дължината на пейката в интервала [0...10]

#### Изход

Да се отпечата на конзолата две числа: броят плочки, необходим за ремонта и времето за поставяне, всяко на нов ред. Резултатът да се форматира до втория знак след запетаята.

#### Примерен вход и изход

Вход	Изход	Обяснения
20 5 4 1 2	19.90 3.98	Обща площ = $20 * 20 = 400$ ; площ на пейката = $1 * 2 = 2$ Площ за покриване = $400 - 2 = 398$ Площ на плочки = $5 * 4 = 20$ Необходими плочки = $398 / 20 = 19.9$ Необходимо време = $19.9 * 0.2 = 3.98$



40	3302.08
0.8	660.42
0.6	
3	
5	

### Решение

```
namespace TilesRepair
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            var N = double.Parse(Console.ReadLine());
```

```
            var W = double.Parse(Console.ReadLine());
```

```
            var L = double.Parse(Console.ReadLine());
```

```
            var M = double.Parse(Console.ReadLine());
```

```
            var O = double.Parse(Console.ReadLine());
```

```
            var Общая_площ = N * N;
```

```
            var Площ_на_пейка = M * O;
```

```
            var Площ_за_покриване = Общая_площ - Площ_на_пейка;
```

```
            var Площ_за_плочки = W * L;
```

```
            var Необходими_плочки = Площ_за_покриване / Площ_за_плочки;
```

```
            var Необходимо_Време = Необходими_плочки * 0.2;
```

```
            Console.WriteLine(Необходими_плочки);
```

```
            Console.WriteLine(Необходимо_Време);
```

```
        }
```

```
    }
```

```
}
```

### Задача 2.17. Парички

Преди време Пешо си е купил биткойни. Сега ще ходи на екскурзия из Европа и ще му трябват евра. Освен биткойни има и китайски юана. Пешо иска да обмени парите си в евро за екскурзията. Напишете програма, която да пресмята колко евра може да купи спрямо следните валутни курсове:

- 1 биткойн = 1168 лева.
- 1 китайски юан = 0.15 долара.
- 1 долар = 1.76 лева.
- 1 евро = 1.95 лева.

Обменното бюро има комисионна от 0 до 5 процента от крайната сума в евро.

### Вход

От конзолата се четат 3 числа, въведени от потребителя:

- На първия ред – броят биткойни. Цяло число в интервала [0...20]



- На втория ред – броят китайски юана. Реално число в интервала [0.00... 50 000.00]
- На третия ред – комисионната. Реално число в интервала [0.00 ... 5.00]

#### Изход

На конзолата да се отпечата 1 число - резултатът от обмяната на валутите. Резултатът да се форматира до втория знак след запетаята.

#### Примерен вход и изход

Вход	Изход	Обяснения
1 5 5	569.67	1 биткойн = 1168 лева 5 юана = 0.75 долара 0.75 долара = 1.32 лева $1168 + 1.32 = 1169.32$ лева = 599.651282051282 евро Комисионна: 5% от 599.651282051282 = 29.9825641025641 Резултат: $599.651282051282 - 29.9825641025641 =$ 569.668717948718 евро

Вход	Изход	Вход	Изход
20 5678 2.4	12442.24	7 50200.12 3	10659.47

#### Решение

`namespace Money`

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var BTC = double.Parse(Console.ReadLine()) * 1168;  
            var CNY = (double.Parse(Console.ReadLine()) * 0.15) * 1.76;  
            var Total = (BTC + CNY) / 1.95;  
            var Commission = (double.Parse(Console.ReadLine()) / 100) * Total;  
            Console.WriteLine($"{Total - Commission:f2}");  
        }  
    }  
}
```

#### Задача 2.18. Дневна печалба

Иван е програмист в американска компания и работи от вкъщи средно  $N$  дни в месеца като изкарва средно по  $M$  долара на ден. В края на годината Иван получава бонус, който е равен на 2.5 месечни заплати. От спечеленото през годината му се удържат 25% данъци. Напишете програма, която да пресмята, колко е чистата средна печалба на Иван на ден в лева, тъй като той харчи изкараното в България. Приема се, че в



годината има точно 365 дни. Курсът на долара спрямо лева ще се чете от конзолата.

### Вход

От конзолата се четат 3 числа, въведени от потребителя:

- На първия ред – работни дни в месеца. Цяло число в интервала [5...30]
- На втория ред – изкарани пари на ден. Реално число в интервала [10.00... 2000.00]
- На третия ред – курсът на долара спрямо лева /1 долар = X лева/. Реално число в интервала [0.99... 1.99]

### Изход

На конзолата да се отпечата 1 число – средната печалба на ден в лева. Резултатът да се форматира до втория знак след запетаята.

### Примерен вход и изход

Вход	Изход	Обяснения
21 75.00 1.59	74.61	1 месечна заплата = $21 * 75 = 1575$ долара Годишен доход = $1575 * 12 + 1575 * 2.5 = 22837.5$ долара Данък = 25% от 22837.5 = 5709.375 Чист годишен доход = 17128.125 долара = 27233.71875 лв Средна печалба на ден = $27233.71875 / 365 = 74.61$ лева

Вход	Изход	Вход	Изход
15 105 1.71	80.24	22 199.99 1.50	196.63

### Решение

```
namespace DailyIncome
{
    class Program
    {
        static void Main(string[] args)
        {
            var Days = double.Parse(Console.ReadLine());
            var Money = double.Parse(Console.ReadLine());
            var UsdBgn = double.Parse(Console.ReadLine());

            var Income = ((Days * Money) * 12) + ((Days * Money) * 2.5);
            var Tax = Income * 0.25;
            var PerDay = ((Income - Tax) * UsdBgn) / 365;

            Console.WriteLine($"{PerDay:f2}");
        }
    }
}
```



```
}  
}  
}
```

## Тема 3. Проверки

### Задача 3.1. 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
6	Excellent!

ВХОД	ИЗХОД
5	(няма изход)

ВХОД	ИЗХОД
5.50	Excellent!

ВХОД	ИЗХОД
5.49	(няма изход)

#### Решение

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

### Задача 3.2. 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
6	Excellent!

ВХОД	ИЗХОД
5	Not excellent.

ВХОД	ИЗХОД
5.50	Excellent!

ВХОД	ИЗХОД
5.49	Not excellent.

#### Решение

```
namespace ExcellentOrNot  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var grade = double.Parse(Console.ReadLine());  
            if (grade >= 5.50)  
            {  
                Console.WriteLine("Excellent!");  
            }  
            else  
            {  
                Console.WriteLine("Not excellent.");  
            }  
        }  
    }  
}
```





```
        Console.WriteLine("Not Excellent.");  
    }  
}  
}
```

### Задача 3.3. Четно или нечетно число

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

ВХОД	ИЗХОД
2	even

ВХОД	ИЗХОД
3	odd

ВХОД	ИЗХОД
25	odd

ВХОД	ИЗХОД
1024	even

*Решение*

```
namespace OddOrEven  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var num = int.Parse(Console.ReadLine());  
            if (num % 2 == 0)  
            {  
                Console.WriteLine("even");  
            }  
            else  
            {  
                Console.WriteLine("odd");  
            }  
        }  
    }  
}
```

### Задача 3.4. 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

*Решение*

```
namespace MaxOfTwo  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Enter two integers:");  
            var a = int.Parse(Console.ReadLine());  
            var b = int.Parse(Console.ReadLine());  
            if (a > b)  
            {
```



```
        Console.WriteLine("Greater number: " + a);  
    }  
    else  
    {  
        Console.WriteLine("Greater number: " + b);  
    }  
}  
}
```

### Задача 3.5. 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!

Подсказка: използвайте if-else конструкцията.

#### Решение

namespace GuessThePassword

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var pass = Console.ReadLine();  
            if (pass == "s3cr3t!P@ssw0rd")  
            {  
                Console.WriteLine("Welcome");  
            }  
            else  
            {  
                Console.WriteLine("Wrong password!");  
            }  
        }  
    }  
}
```

### Задача 3.6. Еднакви гumi

Да се напише програма, която чете две гumi, въведени от потребителя, и проверява дали са еднакви. Да не се прави разлика между главни и малки гumi. 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



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

#### Решение

namespace SameWords

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var word1 = Console.ReadLine().ToLower();  
            var word2 = Console.ReadLine().ToLower();  
            if (word1 == word2)  
            {  
                Console.WriteLine("yes");  
            }  
            else  
            {  
                Console.WriteLine("no");  
            }  
        }  
    }  
}
```

#### Задача 3.7. Време плюс 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

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.

#### Решение

namespace Add15ToTime

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var h = int.Parse(Console.ReadLine());  
            var m = int.Parse(Console.ReadLine());  
            m = m + 15;
```



```
        if (m >= 60)
        {
            h += 1; // h = h + 1;
            m -= 60; // m = m - 60;
        }
        if (h > 23) h -= 24;
        if (m < 10) Console.WriteLine("{0}:0{1}", h, m);
        else Console.WriteLine("{0}:{1}", h, m);
    }
}
```

### Задача 3.8. Три еднакви числа

Да се напише програма, която чете 3 числа и отпечатва дали са еднакви (yes / no).

#### Решение

```
namespace ThreeEqualNumbers
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            var a = double.Parse(Console.ReadLine());
            var b = double.Parse(Console.ReadLine());
            var c = double.Parse(Console.ReadLine());
            if (a == b && b == c)
            {
                Console.WriteLine("yes");
            }
            else
            {
                Console.WriteLine("no");
            }
        }
    }
}
```

### Задача 3.9. 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
5	five

ВХОД	ИЗХОД
1	one

ВХОД	ИЗХОД
9	nine

ВХОД	ИЗХОД
10	number too big

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

#### Решение

```
namespace NumberToWord
```

```
{
    class Program
```



```
{
    static void Main(string[] args)
    {
        var num = int.Parse(Console.ReadLine());
        if (num == 1) Console.WriteLine("one");
        else if (num == 2) Console.WriteLine("two");
        else if (num == 3) Console.WriteLine("three");
        else if (num == 4) Console.WriteLine("four");
        else if (num == 5) Console.WriteLine("five");
        else if (num == 6) Console.WriteLine("six");
        else if (num == 7) Console.WriteLine("seven");
        else if (num == 8) Console.WriteLine("eight");
        else if (num == 9) Console.WriteLine("nine");
        else Console.WriteLine("number too big");
    }
}
```

### Задача 3.10. 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
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:

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

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).

### Решение

namespace BonusPoint



```
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter score: ");
            var num = double.Parse(Console.ReadLine());

            var bonusScore = 0.0;
            if (num <= 100) bonusScore = 5;
            else if (num <= 1000) bonusScore = num * 0.2;
            else bonusScore = num * 0.1;

            if (num % 10 == 5) bonusScore += 2;
            else if (num % 2 == 0) bonusScore += 1;

            Console.WriteLine("Bonus score: {0}", bonusScore);
            Console.WriteLine("Total score: {0}", num + bonusScore);
        }
    }
}
```

### Задача 3.11. 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
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

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.

### Решение

```
namespace SumOfSeconds
{
    class Program
    {
        static void Main(string[] args)
```



```
{
    int a = int.Parse(Console.ReadLine());
    int b = int.Parse(Console.ReadLine());
    int c = int.Parse(Console.ReadLine());
    int s = a + b + c;
    if (s < 60) Console.Write("0:");
    else if (s < 120)
    {
        Console.Write("1:");
        s -= 60;
    }
    else
    {
        Console.Write("2:");
        s -= 120;
    }
    if (s < 10) Console.WriteLine("0" + s);
    else Console.WriteLine(s);
}
}
```

### Задача 3.12. 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
		ВХОД	ИЗХОД		



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

### Решение

```
namespace UnitConverter
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = double.Parse(Console.ReadLine());
            var e1 = Console.ReadLine();
            var e2 = Console.ReadLine();

            // Обявят брой проверки е 56!
            if (e1 == "m" && e2 == "mm") Console.WriteLine("{0} mm", n * 1000);
            if (e1 == "m" && e2 == "cm") Console.WriteLine("{0} cm", n * 100);
            if (e1 == "m" && e2 == "mi") Console.WriteLine("{0} mi", n *
0.000621371192);
            if (e1 == "m" && e2 == "in") Console.WriteLine("{0} in", n *
39.3700787);
            if (e1 == "m" && e2 == "km") Console.WriteLine("{0} km", n * 0.001);
            if (e1 == "m" && e2 == "ft") Console.WriteLine("{0} ft", n * 3.2808399);
            if (e1 == "m" && e2 == "yd") Console.WriteLine("{0} yd", n * 1.0936133);

        }
    }
}
```

### Задача 3.13. Число от 100 до 200

Да се напише програма, която чете цяло число, въведено от потребителя, и проверява дали е по-малко от 100, между 100 и 200 или над 200. Да се отпечата 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

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

### Решение

```
namespace NumberFrom100To200
{
    class Program
    {
        static void Main(string[] args)
        {
            int num = int.Parse(Console.ReadLine());
            if (num < 100)
            {
                Console.WriteLine("Less than 100");
            }
        }
    }
}
```





```
    }  
    else if (num <= 200)  
    {  
        Console.WriteLine("Between 100 and 200");  
    }  
    else  
    {  
        Console.WriteLine("Greater than 200");  
    }  
}  
}
```

### Задача 3.14. 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:

ВХО Д	ИЗХО Д
8	slow

ВХО Д	ИЗХОД
49.5	average

ВХО Д	ИЗХО Д
126	fast

ВХО Д	ИЗХОД
160	ultra fast

ВХОД	ИЗХОД
3500	extremely fast

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

#### Решение

```
namespace SpeedInfo  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var speed = double.Parse(Console.ReadLine());  
            if (speed <= 10) Console.WriteLine("slow");  
            else if (speed <= 50) Console.WriteLine("average");  
            else if (speed <= 150) Console.WriteLine("fast");  
            else if (speed <= 1000) Console.WriteLine("ultra fast");  
            else Console.WriteLine("extremely fast");  
        }  
    }  
}
```

### Задача 3.15. 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

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

*Решение*

`namespace FigureArea`

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var fig = Console.ReadLine();  
            if (fig == "square")  
            {  
                var a = double.Parse(Console.ReadLine());  
                Console.WriteLine(a * a);  
            }  
            else if (fig == "rectangle")  
            {  
                var a = double.Parse(Console.ReadLine());  
                var b = double.Parse(Console.ReadLine());  
                Console.WriteLine(a * b);  
            }  
            else if (fig == "circle")  
            {  
                var r = double.Parse(Console.ReadLine());  
                Console.WriteLine(Math.PI * r * r);  
            }  
            else  
            {  
                var a = double.Parse(Console.ReadLine());  
                var h = double.Parse(Console.ReadLine());  
                Console.WriteLine((a * h)/2.0);  
            }  
        }  
    }  
}
```



### Задача 3.16. 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.

#### Решение

`namespace` NumberToWords

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var num = int.Parse(Console.ReadLine());  
            if (num == 0) Console.WriteLine("zero");  
            else if (num == 1) Console.WriteLine("one");  
            else if (num == 2) Console.WriteLine("two");  
            else if (num == 10) Console.WriteLine("ten");  
            else if (num == 11) Console.WriteLine("eleven");  
            else if (num == 12) Console.WriteLine("twelve");  
            else if (num == 13) Console.WriteLine("thirteen");  
            else if (num == 15) Console.WriteLine("fifteen");  
            else if (num == 19) Console.WriteLine("nineteen");  
            else if (num == 20) Console.WriteLine("twenty");  
            else if (num == 21) Console.WriteLine("twenty one");  
            else if (num == 25) Console.WriteLine("twenty five");  
            else if (num == 27) Console.WriteLine("twenty seven");  
            else if (num == 30) Console.WriteLine("thirty");  
            else if (num == 38) Console.WriteLine("thirty eight");  
            else if (num == 41) Console.WriteLine("forty one");  
            else if (num == 50) Console.WriteLine("fifty");  
            else if (num == 52) Console.WriteLine("fifty two");  
            else if (num == 63) Console.WriteLine("sixty three");  
            else if (num == 74) Console.WriteLine("seventy four");  
            else if (num == 85) Console.WriteLine("eighty five");  
            else if (num == 96) Console.WriteLine("ninety six");  
            else if (num == 99) Console.WriteLine("ninety nine");  
            else if (num == 100) Console.WriteLine("one hundred");  
            else Console.WriteLine("invalid number");  
        }  
    }  
}
```

### Задача 3.17. Обръщение според възраст и пол

Да се напише конзолна програма, която прочита възраст (десетично число) и пол ("m" или "f"), въведени от потребителя, и отпечата 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
12 f	Miss

ВХОД	ИЗХОД
17 m	Mr.

ВХОД	ИЗХОД
25 f	Ms.

ВХОД	ИЗХОД
13.5 m	Master

### Решение

namespace PersonalTitles

```
{
    class Program
    {
        static void Main(string[] args)
        {
            var age = int.Parse(Console.ReadLine());
            var sex = Console.ReadLine();
            if (sex == "f")
            {
                if (age < 16) Console.WriteLine("Miss");
                else Console.WriteLine("Ms.");
            }
            else
            {
                if (age < 16) Console.WriteLine("Master");
                else Console.WriteLine("Mr.");
            }
        }
    }
}
```

### Задача 3.18. 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
coffee Varna 2	0.9

ВХОД	ИЗХОД
peanuts Plovdiv 1	1.5

ВХОД	ИЗХОД
beer Sofia 6	7.2

ВХОД	ИЗХОД
water Plovdiv 3	2.1

ВХОД	ИЗХОД
sweets Sofia 2.23	3.2335



### Решение

```
namespace SmallShop
{
    class Program
    {
        static void Main(string[] args)
        {
            var product = Console.ReadLine().ToLower();
            var town = Console.ReadLine().ToLower();
            var count = double.Parse(Console.ReadLine());
            if (town == "sofia")
            {
                if (product == "coffee") Console.WriteLine(0.5 * count);
                if (product == "water") Console.WriteLine(0.8 * count);
                if (product == "beer") Console.WriteLine(1.2 * count);
                if (product == "sweets") Console.WriteLine(1.45 * count);
                if (product == "peanuts") Console.WriteLine(1.6 * count);
            }
            if (town == "varna")
            {
                if (product == "coffee") Console.WriteLine(0.45 * count);
                if (product == "water") Console.WriteLine(0.7 * count);
                if (product == "beer") Console.WriteLine(1.1 * count);
                if (product == "sweets") Console.WriteLine(1.35 * count);
                if (product == "peanuts") Console.WriteLine(1.55 * count);
            }
            if (town == "plovdiv")
            {
                if (product == "coffee") Console.WriteLine(0.4 * count);
                if (product == "water") Console.WriteLine(0.7 * count);
                if (product == "beer") Console.WriteLine(1.15 * count);
                if (product == "sweets") Console.WriteLine(1.3 * count);
                if (product == "peanuts") Console.WriteLine(1.5 * count);
            }
        }
    }
}
```

### Задача 3.19. Точка в правоъгълник

Напишете програма, която проверява дали точка  $\{x, y\}$  се намира вътре в правоъгълник  $\{x_1, y_1\} - \{x_2, y_2\}$ . Входните данни се четат от конзолата и се състоят от 6 реда, въведени от потребителя: десетичните числа  $x_1, y_1, x_2, y_2, x$  и  $y$  (като се гарантира, че  $x_1 < x_2$  и  $y_1 < y_2$ ). 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
Д		

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	

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 ( $x \geq x1$ )
- Точката е наляво от дясната стена на правоъгълника ( $x \leq x2$ )
- Точката е надолу от горната стена на правоъгълника ( $y \geq y1$ )
- Точката е нагоре от долната стена на правоъгълника ( $y \leq y2$ )

*Решение*

`namespace` InsideRectangle

```
{
    class Program
    {
        static void Main(string[] args)
        {
            var x1 = double.Parse(Console.ReadLine());
            var y1 = double.Parse(Console.ReadLine());
            var x2 = double.Parse(Console.ReadLine());
            var y2 = double.Parse(Console.ReadLine());
            var x = double.Parse(Console.ReadLine());
            var y = double.Parse(Console.ReadLine());
            if (x >= x1 && x <= x2 && y >= y1 && y <= y2)
            {
                Console.WriteLine("Inside");
            }
            else
            {

```



```
        Console.WriteLine("Outside");  
    }  
}  
}
```

### Задача 3.20. 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
banana	fruit

ВХОД	ИЗХОД
apple	fruit

ВХОД	ИЗХОД
tomato	vegetable

ВХОД	ИЗХОД
water	unknown

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

#### Решение

```
namespace FruitVegetable
```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var produkt = Console.ReadLine();  
            switch (produkt)  
            {  
                case "banana":  
                case "apple":  
                case "kiwi":  
                case "cherry":  
                case "lemon":  
                case "grapes": Console.WriteLine("fruit"); break;  
                case "tomato":  
                case "cucumber":  
                case "pepper":  
                case "carrot": Console.WriteLine("vegetable"); break;  
                default: Console.WriteLine("unknown"); break;  
            }  
        }  
    }  
}
```

### Задача 3.21. 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:



ВХО Д	ИЗХОД	ВХО Д	ИЗХОД	ВХО Д	ИЗХОД	ВХО Д	ИЗХОД
75	invalid	150	(няма изход)	220	invalid	199	(няма изход)
-1	invalid	100	(няма изход)	200	(няма изход)	0	(няма изход)

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

### Решение

```
namespace InvalidNumber
{
    class Program
    {
        static void Main(string[] args)
        {
            var a = int.Parse(Console.ReadLine());
            if (a < 100 || a > 200 || a != 0)
                Console.WriteLine("Invalid");
        }
    }
}
```

### Задача 3.22. Точка върху страната на правоъгълник

Напишете програма, която проверява дали точка  $\{x, y\}$  се намира върху някоя от страните на правоъгълник  $\{x_1, y_1\} - \{x_2, y_2\}$ . Входните данни се четат от конзолата и се състоят от 6 реда въведени от потребителя: десетичните числа  $x_1, y_1, x_2, y_2, x$  и  $y$  (като се гарантира, че  $x_1 < x_2$  и  $y_1 < y_2$ ). Да се отпечата **"Border"** (точката лежи на някоя от страните) или **"Inside / Outside"** (в противен случай). Примери:

ВХОД	ИЗХОД	визуализация
2 -3 12 3 8 -1	Inside / Outside	
2 -3 12 3 12 -1	Border	





**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.

### Решение

**namespace** BorderPoint

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            // Въвеждане на входните данни  
            var x1 = double.Parse(Console.ReadLine());  
            var y1 = double.Parse(Console.ReadLine());  
            var x2 = double.Parse(Console.ReadLine());  
            var y2 = double.Parse(Console.ReadLine());  
            var x = double.Parse(Console.ReadLine());  
            var y = double.Parse(Console.ReadLine());  
            // Проверки дали точката лежи на някоя от страните на правоъгълника  
            if ((x == x1 || x == x2) && (y >= y1 && (y <= y2)) ||  
                ((y == y1 || y == y2) && (x >= x1 && (x <= x2)))  
            {  
                Console.WriteLine("Border");  
            }  
            else  
            {  
                Console.WriteLine("Inside / Outside");  
            }  
        }  
    }  
}
```

### Задача 3.23. 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
1	Monday
2	Tuesday
3	Wednesday

Вход	Изход
5	Friday
6	Saturday
7	Sunday



4	Thursday
---	----------

8	Error
---	-------

#### Решение

namespace DayOfWeek

```
{
    class Program
    {
        static void Main(string[] args)
        {
            int day = int.Parse(Console.ReadLine());
            switch (day)
            {
                case 1: Console.WriteLine("Monday"); break;
                case 2: Console.WriteLine("Tuesday"); break;
                case 3: Console.WriteLine("Wednesday"); break;
                case 4: Console.WriteLine("Thursday"); break;
                case 5: Console.WriteLine("Friday"); break;
                case 6: Console.WriteLine("Saturday"); break;
                case 7: Console.WriteLine("Sunday"); break;
                default: Console.WriteLine("Error"); break;
            }
        }
    }
}
```

#### Задача 3.24. 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

#### Решение

namespace AnimalClass

```
{
    class Program
    {
        static void Main(string[] args)
        {
            var animal = Console.ReadLine();
            switch (animal)
            {
                case "dog": Console.WriteLine("mammal"); break;
                case "crocodile":
```



```
        case "tortoise":  
        case "snake": Console.WriteLine("reptile"); break;  
        default: Console.WriteLine("unknown"); break;  
    }  
}  
}
```

### Задача 3.25. 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
Premiere 10 12	1440.00 leva

ВХОД	ИЗХОД
Normal 21 13	2047.50 leva

ВХОД	ИЗХОД
Discount 12 30	1800.00 leva

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

#### Решение

```
namespace Cinema  
{
```

```
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var type = Console.ReadLine();  
            var row = int.Parse(Console.ReadLine());  
            var col = int.Parse(Console.ReadLine());  
            var price = 0.0;  
            switch (type)  
            {  
                case "Premiere": price = 12.00; break;  
                case "Normal": price = 7.50; break;  
                case "Discount": price = 5.00; break;  
            }  
            var total = row * col * price;  
            Console.WriteLine("{0:f2} leva", total);  
        }  
    }  
}
```



```
}  
}
```

### Задача 3.26. Цена за транспорт

Студент трябва да пропътува  $n$  километра. Той има избор измежду три вида транспорт:

- Такси. Начална такса: 0.70 лв. Дневна тарифа: 0.79 лв. / км. Нощна тарифа: 0.90 лв. / км.
- Автобус. Дневна / нощна тарифа: 0.09 лв. / км. Може да се използва за разстояния минимум 20 км.
- Влак. Дневна / нощна тарифа: 0.06 лв. / км. Може да се използва за разстояния минимум 100 км.

Напишете програма, която чете броя километри  $n$  и период от деня (ден или нощ), въведени от потребителя, и изчислява цената на най-евтиния транспорт.

#### Вход

От конзолата се четат два реда:

- Първият ред съдържа числото  $n$  – брой километри – цяло число в интервала [1...5000].
- Вторият ред съдържа думата "day" или "night" – пътуване през деня или през нощта.

#### Изход

Да се отпечата на конзолата най-ниската цена за посочения брой километри. Резултатът да се форматира до втория знак след десетичната запетая.

#### Примерен вход и изход

Вход	Изход	Обяснения
5 day	4.65	Разстоянието е под 20 км → може да се ползва само такси. Началната такса е 0.70 лв. Понеже е през деня, тарифата е 0.79 лв. / км. С такси цената е: $0.70 + 5 * 0.79 = 4.65$ лв.
7 night	7.00	Разстоянието е под 20 км → може да се ползва само такси. Началната такса е 0.70 лв. Понеже е през нощта, тарифата е 0.90 лв. / км. С такси цената е: $0.70 + 7 * 0.90 = 7.00$ лв.
25 day	2.25	Разстоянието е над 20 км → може да се ползва автобус, но не може да се ползва влак. Автобусът е най-евтиния възможен вариант. С автобус цената е: $25 * 0.09 = 2.25$ лв.
180 night	10.80	Разстоянието е над 100 км → може да се ползва влак. Влакът е най-евтиният възможен вариант за пътуване. С влак цената е: $180 * 0.06 = 10.80$ лв.



### Решение

```
namespace TransportPrice
{
    class Program
    {
        static void Main(string[] args)
        {
            var km = long.Parse(Console.ReadLine());
            var t = Console.ReadLine();
            double price = 0;

            if (km < 20 && t == "day") price = 0.7 + km * 0.79;
            else if (km < 20 && t == "night") price = 0.7 + km * 0.9;
            else if (km >= 20 && km < 100) price = km * 0.09;
            else if (km >= 100) price = km * 0.06;

            Console.WriteLine( price);
        }
    }
}
```

### Задача 3.27. Тръби в басейн

Басейн с обем  $V$  има две тръби от които се пълни. Всяка тръба има определен дебит (литрите вода минаващи през една тръба за един час). Работникът пуска тръбите едновременно и излиза за  $N$  часа. Напишете програма, която изкарва състоянието на басейна, в момента, когато работникът се върне.

#### Вход

От конзолата се четат четири реда, въведени от потребителя:

- Първият ред съдържа числото  $V$  – **Обем на басейна в литри** – цяло число в интервала [1...10000].
- Вторият ред съдържа числото  $P1$  – **дебит на първата тръба за час** – цяло число в интервала [1...5000].
- Третият ред съдържа числото  $P2$  – **дебит на втората тръба за час** – цяло число в интервала [1...5000].
- Четвъртият ред съдържа числото  $N$  – **часовете които работникът отсъства** – число с плаваща запетая в интервала [1.0...24.00]

#### Изход

Да се отпечата на конзолата едно от двете възможни състояния:

- До колко се е запълнил басейна и коя тръба с колко процента е допринесла. Всички проценти се свеждат до цяло число (без закръгляне).
  - "The pool is [x]% full. Pipe 1: [y]%. Pipe 2: [z]%."
- Ако басейнът се е препълнил – с колко литра е прелял за даденото време, число с плаваща запетая, форматирано до първия знак след десетичната запетая.
  - "For [x] hours the pool overflows with [y] liters."



\* Имайте предвид, че поради свеждането до цяло число се губят данни и нормално сборът на процентите да е 99%, а не 100%.

#### Примерен вход и изход

Вход	Изход	Обяснения
100 0 100 120 3	The pool is 66% full. Pipe 1: 45%. Pipe 2: 54%.	За 3 часа: Първата тръба е напълнила – 300 л. Втората тръба е напълнила – 360 л. Общо – 660 л. < 1000 л. => 66% са запълнени Първата тръба е допринесла с 45% (300 от 660 л.). Втората тръба е допринесла с 54% (360 от 660 л.).
100 100 100 2.5	For 2.5 hours the pool overflows with 400.0 liters.	За 2.5 часа: Първата тръба е напълнила – 250 л. Втората тръба е напълнила – 250 л. Общо – 500 л. > 100 л. => 400 л. са преляли.

#### Решение

```
namespace PoolPipes
{
    class Program
    {
        static void Main(string[] args)
        {
            var V = int.Parse(Console.ReadLine());
            var P1 = int.Parse(Console.ReadLine());
            var P2 = int.Parse(Console.ReadLine());
            var H = double.Parse(Console.ReadLine());

            var t1 = P1 * H;
            var t2 = P2 * H;
            var all = t1 + t2;
            var allPROCENT = (all / V) * 100;
            var T1PROCENT = (t1 / all) * 100;
            var T2PROCENT = (t2 / all) * 100;

            if (all <= V)
            {
                Console.WriteLine("The pool is {0}% full. Pipe 1: {1}%. Pipe 2: {2}%", Math.Floor(allPROCENT), Math.Floor(T1PROCENT), Math.Floor(T2PROCENT));
            }
            else if (all > V)
            {
                var AL = all - V;
                Console.WriteLine("For {0} hours the pool overflows with {1} liters.", H, AL);
            }
        }
    }
}
```



```
}  
}  
}
```

### Задача 3.28. Поспаливата котка Том

Котката Том обича по цял ден да спи, за негово съжаление стопанинът му си играе с него винаги когато има свободно време. За да се наспи добре, нормата за игра на Том е 30 000 минути в година. Времето за игра на Том зависи от почивните дни на стопанина му:

- Когато е на **работа**, стопанинът му си играе с него **по 63 минути на ден**.
- Когато **почива**, стопанинът му си играе с него **по 127 минути на ден**.

Напишете програма, която чете броя почивни дни, въведени от потребителя, и отпечатва дали Том може да се наспи добре и колко е разликата от нормата за текущата година, като приемем че годината има 365 дни.

Пример: 20 почивни дни -> работните дни са 345 ( $365 - 20 = 245$ ). Реалното време за игра е 24 275 минути ( $345 * 63 + 20 * 127$ ). Разликата от нормата е 5 725 минути ( $30\,000 - 24\,275 = 5\,725$ ) или 95 часа и 25 минути.

#### Вход

Входът се чете от конзолата и се състои от едно число – броят почивни дни – цяло число в интервала [0...365]

#### Изход

На конзолата трябва да се отпечатаат два реда.

- Ако времето за игра на Том **е над нормата** за текущата година:
  - На **първия ред** отпечатайте: **"Tom will run away"**
  - На **втория ред** отпечатайте разликата от нормата във формат:  
**"{H} hours and {M} minutes more for play"**
- Ако времето за игра на Том **е под нормата** за текущата година:
  - На **първия ред** отпечатайте: **"Tom sleeps well"**
  - На **втория ред** отпечатайте разликата от нормата във формат:  
**"{H} hours and {M} minutes less for play"**

#### Примерен вход и изход

вход	изход	коментари
20	Tom sleeps well 95 hours and 25 minutes less for play	Почивни дни: $20 * 127 = 2\,540$ минути игра Работни дни: $365 - 20 = 345 * 63 = 21\,735$ минути игра $30\,000 > 24\,274 \Rightarrow$ остават 5725 мин = 95 часа и 25 мин



113	Tom will run away 3 hours and 47 minutes more for play	Почични дни: $113 * 127 = 14\ 351$ минути Работни дни: $365 - 113 = 252 * 63 =$ $15\ 876$ минути $30\ 000 < 30\ 227 \Rightarrow 227$ мин повече = 3 часа и 47 мин
-----	--	--

### Решение

```
namespace SleepyCatTom
{
    class Program
    {
        static void Main(string[] args)
        {
            int days = int.Parse(Console.ReadLine());

            int playtime = (365 - days) * 63 + days * 127;
            int h = Math.Abs((30000 - playtime) / 60);
            int m = Math.Abs((30000 - playtime) % 60);

            if (playtime < 30000)
            {
                Console.WriteLine("Tom sleeps well");
                Console.WriteLine("{0} hours and {1} minutes less for play", h, m);
            }
            else
            {
                Console.WriteLine("Tom will run away");
                Console.WriteLine("{0} hours and {1} minutes more for play", h, m);
            }
        }
    }
}
```

### Задача 3.29. Реколта

От лозе с площ  $X$  квадратни метри се заделя 40% от реколтата за производство на вино. От 1 кв.м лозе се изкарват  $Y$  килограма грозде. За 1 литър вино са нужни 2,5 кг. грозде. Желаното количество вино за продан е  $Z$  литра.

Напишете програма, която пресмята колко вино може да се произведе и дали това количество е достатъчно. Ако е достатъчно, остатъкът се разделя по равно между работниците на лозето.

#### Вход

Входът се чете от конзолата и се състои от точно 4 реда, въведени от потребителя:

- 1ви ред:  $X$  кв.м е лозето – цяло число в интервала  $[10 \dots 5000]$ ;
- 2ри ред:  $Y$  грозде за един кв.м – реално число в интервала  $[0.00 \dots 10.00]$ ;
- 3ти ред:  $Z$  нужни литри вино – цяло число в интервала  $[10 \dots 600]$ ;
- 4ти ред: брой работници – цяло число в интервала  $[1 \dots 20]$ ;





## Изход

На конзолата трябва да се отпечата следното:

- Ако произведеното вино е по-малко от нужното:
  - "It will be a tough winter! More {недостигащо вино} liters wine needed."
  - Резултатът трябва да е закръглен към по-ниско цяло число
- Ако произведеното вино е повече от нужното:
  - "Good harvest this year! Total wine: {общо вино} liters."
  - Резултатът трябва да е закръглен към по-ниско цяло число
  - "{Оставащо вино} liters left -> {вино за 1 работник} liters per person."
  - И двата резултата трябва да са закръглени към по-високото цяло число

## Примерен вход и изход

ВХОД	ИЗХОД	коментари
650 2 175 3	Good harvest this year! Total wine: 208 liters. 33 liters left -> 11 liters per person.	Общо грозде: $650 * 2 = 1\ 300$ Вино = $40\% * 1\ 300 / 2,5 = 208$ $208 > 175$ $208 - 175 = 33$ л остават -> 11 л на човек
102 0 1.5 425 4	It will be a tough winter! More 180 liters wine needed.	Общо грозде: $1\ 020 * 1.5 = 1\ 530$ Вино = $40\% * 1\ 530 / 2,5 = 244.80$ $244.80 < 425$ $425 - 244.8 = 180.2$ -> 180 л не достигат

## Решение

```
namespace Harvest
{
```

```
    class Program
    {
```

```
        static void Main(string[] args)
        {
```

```
            var x = double.Parse(Console.ReadLine());
            var y = double.Parse(Console.ReadLine());
            var z = double.Parse(Console.ReadLine());
            var workers = int.Parse(Console.ReadLine());
```

```
            var h = (x * y) * 0.4;
            var w = (h / 2.5);
            if(w >= z)
            {
```

```
                var left = w - z;
                Console.WriteLine("Good harvest this year!Total wine: {0} liters.",
```

```
w);
```



```
        Console.WriteLine("{0} liters left -> liters per person.",  
Math.Ceiling(left), left / workers);  
    }  
    else  
    {  
        Console.WriteLine("It will be a tough winter! More {0} liters wine  
needed.", z-w);  
    }  
}  
}
```

### Задача 3.30. Навреме за изпит

Студент трябва да отиде на изпит в определен час (например в 9:30 часа). Той идва в изпитната зала в даден час на пристигане (например 9:40). Счита се, че студентът е дошъл навреме, ако е пристигнал в часа на изпита или до половин час преди това. Ако е пристигнал по-рано повече от 30 минути, той е подранил. Ако е дошъл след часа на изпита, той е закъснял. Напишете програма, която прочита време на изпит и време на пристигане и отпечатва дали студентът е дошъл навреме, дали е подранил или е закъснял и с колко часа или минути е подранил или закъснял.

#### Вход

От конзолата се четат 4 цели числа (по едно на ред), въведени от потребителя:

- Първият ред съдържа **час на изпита** – цяло число от 0 до 23.
- Вторият ред съдържа **минута на изпита** – цяло число от 0 до 59.
- Третият ред съдържа **час на пристигане** – цяло число от 0 до 23.
- Четвъртият ред съдържа **минута на пристигане** – цяло число от 0 до 59.

#### Изход

На първият ред отпечатайте:

- **"Late"**, ако студентът пристига по-късно от часа на изпита.
- **"On time"**, ако студентът пристига точно в часа на изпита или до 30 минути по-рано.
- **"Early"**, ако студентът пристига повече от 30 минути преди часа на изпита.

Ако студентът пристига с поне минута разлика от часа на изпита, отпечатайте на следващия ред:

- **"mm minutes before the start"** за идване по-рано с по-малко от час.
- **"hh:mm hours before the start"** за подраняване с 1 час или повече. Минутите винаги печатайте с 2 цифри, например "1:05".
- **"mm minutes after the start"** за закъснение под час.
- **"hh:mm hours after the start"** за закъснение от 1 час или повече. Минутите винаги печатайте с 2 цифри, например "1:03".

#### Примерен вход и изход



Вход	Изход	Вход	Изход	Вход	Изход
9 30 9 50	Late 20 minutes after the start	9 00 10 30	Late 1:30 hours after the start	10 00 10 00	On time
9 00 8 30	On time 30 minutes before the start	14 00 13 55	On time 5 minutes before the start	11 30 10 55	Early 35 minutes before the start
16 00 15 00	Early 1:00 hours before the start	11 30 8 12	Early 3:18 hours before the start	11 30 12 29	Late 59 minutes after the start

### Решение

```
namespace OnTimeForExam
{
    class Program
    {
        static void Main(string[] args)
        {
            int h1 = int.Parse(Console.ReadLine());
            int m1 = int.Parse(Console.ReadLine());
            int h2 = int.Parse(Console.ReadLine());
            int m2 = int.Parse(Console.ReadLine());

            DateTime dt1 = Convert.ToDateTime(h1 + ":" + m1);
            DateTime dt2 = Convert.ToDateTime(h2 + ":" + m2);
            TimeSpan span = dt2.Subtract(dt1);
            int m = span.Minutes;
            int h = span.Hours;

            if (m < -30) Console.WriteLine("Early");
            else if (m >= -30 && m <= 0) Console.WriteLine("On time");
            else if (m > 0 || h > 0) Console.WriteLine("Late");

            if (h <= -1) Console.WriteLine("{0}:{1} hours before the start",
Math.Abs(h), Math.Abs(m));
            else if (m >= -59 && m <= 0) Console.WriteLine("{0} minutes before the
start", Math.Abs(m));
            else if (h >= 1) Console.WriteLine("{0}:{1} hours after the start",
Math.Abs(h), Math.Abs(m));
            else if (m >= 1 && m <= 59) Console.WriteLine("{0} minutes after the
start", Math.Abs(m));
        }
    }
}
```



### Задача 3.31. Пътешествие

Странно, но повечето хора си планират от рано почивката. Млад програмист разполага с определен бюджет и свободно време в даден сезон. Напишете програма, която да приема на входа бюджета и сезона, а на изхода да изкарва, къде ще почива програмиста и колко ще похарчи.

Бюджета определя дестинацията, а сезона определя колко от бюджета ще изхарчи. Ако е лято ще почива на къмпинг, а зимата в хотел. Ако е в Европа, независимо от сезона ще почива в хотел. Всеки къмпинг или хотел, според дестинацията, има собствена цена която отговаря на даден процент от бюджета:

- При **100лв. или по-малко** – някъде в **България**
  - **Лято** – **30%** от бюджета
  - **Зима** – **70%** от бюджета
- При **1000лв. или по малко** – някъде на **Балканите**
  - **Лято** – **40%** от бюджета
  - **Зима** – **80%** от бюджета
- При **повече от 1000лв.** – някъде из **Европа**
  - При пътуване из Европа, независимо от сезона ще похарчи **90% от бюджета**.

#### Вход

Входът се чете от конзолата и се състои от два реда, въведени от потребителя:

- **Първи ред** – Бюджет, **реално число** в интервала [10.00...5000.00].
- **Втори ред** – Един от двата възможни сезона: „summer” или “winter”

#### Изход

На конзолата трябва да се отпечатаат два реда.

- **Първи ред** – „Somewhere in [дестинация]” измежду “Bulgaria”, “Balkans” и “Europe”
- **Втори ред** – “{Вид почивка} – {Похарчена сума}”
  - Почивката може да е между „Camp” и „Hotel”
  - Сумата трябва да е **закръглена с точност до вторият знак след запетаята**.

#### Примерен вход и изход

ВХОД	ИЗХОД
50 summer	Somewhere in Bulgaria Camp - 15.00
75 winter	Somewhere in Bulgaria Hotel - 52.50
312 summer	Somewhere in Balkans Camp - 124.80
678.53 winter	Somewhere in Balkans Hotel - 542.82



1500 summer	Somewhere in Europe Hotel - 1350.00
----------------	--

### Решение

namespace Trip

{

class Program

{

static void Main(string[] args)

{

var sum = double.Parse(Console.ReadLine());

var type = Console.ReadLine();

if (sum <= 100)

{

Console.WriteLine("Somewhere in Bulgaria");

if (type == "summer") Console.WriteLine("Camp - {0:f2}", sum \* 0.3);

if (type == "winter") Console.WriteLine("Hotel - {0:f2}", sum \* 0.7);

}

else if (sum <= 1000)

{

Console.WriteLine("Somewhere in Balkans");

if (type == "summer") Console.WriteLine("Camp - {0:f2}", sum \* 0.4);

if (type == "winter") Console.WriteLine("Hotel - {0:f2}", sum \* 0.8);

}

else

{

Console.WriteLine("Somewhere in Europe");

Console.WriteLine("Hotel - {0:f2}", sum \* 0.9);

}

}

}

}

### Задача 3.32. Операции между числа

Напишете програма, която чете две цели числа (N1 и N2) и оператор, с който да се извърши дадена математическа операция с тях. Възможните операции са: Събиране(+), Изваждане(-), Умножение(\*), Деление(/) и Модулно деление(%). При събиране, изваждане и умножение на конзолата трябва да се отпечата резултата и дали той е четен или нечетен. При обикновеното деление – резултата. При модулното деление – остатък. Трябва да се има предвид, че делителят може да е равен на 0(нула), а на нула не се дели. В този случай трябва да се отпечата специално съобщение.

### Вход

От конзолата се прочитат 3 реда, въведени от потребителя:

- N1 – цяло число в интервала [0...40 000]
- N2 – цяло число в интервала [0...40 000]
- Оператор – един символ измежду: "+", "-", "\*", "/", "%"



## Изход

Да се отпечата на конзолата един рег:

- Ако операцията е събиране, изваждане или умножение:
  - „{N1} {оператор} {N2} = {резултат} – {even/odd}“
- Ако операцията е деление:
  - „{N1} / {N2} = {резултат}“ – резултатът е форматиран до вторият знак след дес.запетая
- Ако операцията е модулно деление:
  - „{N1} % {N2} = {остатък}“
- В случай на деление с 0(нула):
  - „Cannot divide {N1} by zero“

## Примерен вход и изход

ВХО Д	ИЗХОД	ВХО Д	ИЗХОД	ВХО Д	ИЗХОД
10 12 +	10 + 12 = 22 - even	123 12 /	123 / 12 = 10.25	112 0 /	Cannot divide 112 by zero
10 1 -	10 - 1 = 9 - odd	10 3 %	10 % 3 = 1	10 0 %	Cannot divide 10 by zero
7 3 *	7 * 3 = 21 - odd				

## Решение

namespace Operations

{

class Program

{

static void Main(string[] args)

{

int n1 = int.Parse(Console.ReadLine());

int n2 = int.Parse(Console.ReadLine());

string op = Console.ReadLine();

if (op == "+")

{

Console.Write("{0} + {1} = {2} - ", n1, n2, n1 + n2);

Console.WriteLine((n1 + n2) % 2 == 0 ? "even" : "odd");

}

else if (op == "-")

{

Console.Write("{0} - {1} = {2} - ", n1, n2, n1 - n2);

Console.WriteLine((n1 + n2) % 2 == 0 ? "even" : "odd");

}



```
        else if (op == "*")
        {
            Console.WriteLine("{0} * {1} = {2} - ", n1, n2, n1 * n2);
            Console.WriteLine((n1 + n2) % 2 == 0 ? "even" : "odd");
        }
        else if (op == "/")
        {
            if(n2==0) Console.WriteLine("Cannot divide {0} by zero", n1);
            else Console.WriteLine("{0} / {1} = {2} - ", n1, n2, (double)n1 /
n2);
        }
        else if (op == "%")
        {
            if (n2 == 0) Console.WriteLine("Cannot divide {0} by zero", n1);
            else Console.WriteLine("{0} % {1} = {2} - ", n1, n2, n1 % n2);
        }
    }
}
```

### Задача 3.33. Билети за мач

Когато пуснали билетите за Евро 2016, група запалянковци решили да си закупят. Билетите имат две категории с различни цени:

- VIP – 499.99 лева.
- Normal – 249.99 лева.

Запалянковците имат определен бюджет, а броят на хората в групата определя какъв процент от бюджета трябва да се задели за транспорт:

- От 1 до 4 – 75% от бюджета.
- От 5 до 9 – 60% от бюджета.
- От 10 до 24 – 50% от бюджета.
- От 25 до 49 – 40% от бюджета.
- 50 или повече – 25% от бюджета.

Напишете програма, която да пресмята дали с останалите пари от бюджета могат да си купят билети за избраната категория. И колко пари ще им останат или ще са им нужни.

#### Вход

Програмата чете точно 3 реда , въведени от потребителя:

- На първия ред е бюджетът – реално число в интервала [1 000.00 ... 1 000 000.00]
- На втория ред е категорията – „VIP“ или „Normal“
- На третия ред е броят на хората в групата – цяло число в интервала [1 ... 200]

#### Изход

Да се отпечата на конзолата един ред:



- Ако бюджетът е достатъчен:
  - "Yes! You have {N} leva left." – N са останалите пари на групата
- Ако бюджетът НЕ Е достатъчен:
  - "Not enough money! You need {M} leva." – където M е сумата, която не достига

Сумите трябва да са форматирувани с точност до два знака след десетичната запетая.

#### Примерен вход и изход

Вход	Изход	Вход	Изход
1000 Normal 1	Yes! You have 0.01 leva left.	30000 VIP 49	Not enough money! You need 6499.51 leva.
Обяснения		Обяснения	
1 човек: 75% от бюджета отиват за транспорт Остават: $1000 - 750 = 250$ Категория Normal: билетът струва $249.99 * 1$ $249.99 < 250$ : остават му $250 - 249.99 = 0.01$		49 човек: 40% от бюджета отиват за транспорт Остават: $30000 - 12000 = 18000$ Категория VIP: билета(ът) струва $499.99 * 49$ $24499.510000000002 < 18000$ Не стигат $24499.510000000002 - 18000 = 6499.51$	

#### Решение

namespace Euro2016

```
{
    class Program
    {
        static void Main(string[] args)
        {
            var budget = double.Parse(Console.ReadLine());
            var type = Console.ReadLine();
            var people = int.Parse(Console.ReadLine());

            double tickets = 0;
            if (type == "VIP") tickets = people * 499.99;
            else tickets = people * 249.99;

            double transport = 0;
            if (people < 5) transport = budget * 0.75;
            else if (people < 10) transport = budget * 0.60;
            else if (people < 25) transport = budget * 0.50;
            else if (people < 50) transport = budget * 0.40;
            else transport = budget * 0.25;

            if (budget < (tickets + transport))
            {
                Console.WriteLine($"Not enough money! You need {(tickets + transport) - budget}:f2} leva.");
            }
        }
    }
}
```





```
    }  
    else  
    {  
        Console.WriteLine($"Yes! You have {(budget - (tickets +  
transport))}:f2} leva left.");  
    }  
}  
}
```

### Задача 3.34. Хотелска стая

Хотел предлага 2 вида стаи: студио и апартамент. Напишете програма, която изчислява цената за целия престой за студио и апартамент. Цените зависят от месеца на престоя:

Май и октомври	Юни и септември	Юли и август
Студио – 50 лв./нощувка	Студио – 75.20 лв./нощувка	Студио – 76 лв./нощувка
Апартамент – 65 лв./нощувка	Апартамент – 68.70 лв./нощувка	Апартамент – 77 лв./нощувка

Предлагат се и следните отстъпки:

- За студио, при повече от 7 нощувки през май и октомври : 5% намаление.
- За студио, при повече от 14 нощувки през май и октомври : 30% намаление.
- За студио, при повече от 14 нощувки през юни и септември: 20% намаление.
- За апартамент, при повече от 14 нощувки, без значение от месеца : 10% намаление.

#### Вход

Входът се чете от конзолата и съдържа точно 2 реда, въведени от потребителя:

- На първия ред е месецът – May, June, July, August, September или October
- На втория ред е броят на нощувките – цяло число в интервала [0 ... 200]

#### Изход

Да се отпечата на конзолата 2 реда:

- На първия ред: "Apartment: {цена за целият престой} lv."
- На втория ред: "Studio: {цена за целият престой} lv."

Цената за целия престой форматирана с точност до два знака след десетичната запетая.

#### Примерен вход и изход

Вход	Изход	Обяснения
Д		



May 15	Apartment: 877.50 lv. Studio: 525.00 lv.	През май, при повече от 14 нощувки, намаляваме цената на студиото с 30% (50 – 15 = 35), а на апартамента – с 10% (65 – 6.5 = 58.5). Целият престой в апартамент – 877.50 лв. Целият престой в студио – 525.00 лв.	
Вход	Изход	Вход	Изход
June 14	Apartment: 961.80 lv. Studio: 1052.80 lv.	August 20	Apartment: 1386.00 lv. Studio: 1520.00 lv.

### Решение

namespace HotelRoom

```
{
    class Program
    {
        static void Main(string[] args)
        {
            var m = Console.ReadLine();
            var brn = int.Parse(Console.ReadLine());
            double PS = 0;
            double PA = 0;
            if (m == "May" || m == "October")
            {
                PS *= 50;
                PA *= 65;
                if (brn > 5 && brn <= 14) PS *= 0.95;
                if (brn > 14)
                {
                    PS *= 0.7;
                    PA *= 0.9;
                }
            }
            else if (m == "June" || m == "September")
            {
                PS *= 75.2;
                PA *= 68.7;
                if (brn > 14)
                {
                    PS *= 0.8;
                    PA *= 0.9;
                }
            }
            else
            {
                PS *= 76;
                PA *= 77;
                if (brn > 14) PA *= 0.9;
            }
            Console.WriteLine("Apartment: {0:f2}", PA);
            Console.WriteLine("Studio: {0:f2}", PS);
        }
    }
}
```



```
}  
}  
}
```

## Тема 4. Повторения

### Задача 4.1. 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

#### Решение

```
namespace Numbers1to100  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            for (int i = 1; i <= 100; i++)  
            {  
                Console.WriteLine("i=" + i);  
            }  
        }  
    }  
}
```

### Задача 4.2. 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

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**).

#### Решение

```
namespace NumbersEnding7  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {
```



```
for (int k = 0; k < 1000; k++)  
{  
    if (k % 10 == 7)  
    {  
        Console.Write(k + " ");  
    }  
}
```

### Задача 4.3. Всички латински букви

Напишете програма, която отпечата всички букви от латинската азбука: a, b, c, ..., z.

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

#### Решение

namespace LatinLetters

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            for (var letter = 'a'; letter <= 'z'; letter++)  
            {  
                Console.Write(letter + " ");  
            }  
            Console.WriteLine();  
        }  
    }  
}
```

### Задача 4.4. Сумиране на числа

Да се напише програма, която чете n-на брой цели числа, въведени от потребителя, и ги сумира.

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

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

Примери:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
2	30	3	-60	4	43	1	999	0	0
10		-10		45		999			
20		-20		-20					
		-30		7					



					11						
--	--	--	--	--	----	--	--	--	--	--	--

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.

*Решение*

`namespace SumOfNumbers`

```
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            var sum = 0;
            for (int i = 0; i < n; i++)
            {
                var next = int.Parse(Console.ReadLine());
                sum = sum + next;
            }
            Console.WriteLine(sum);
        }
    }
}
```

### Задача 4.5. Най-голямо число

Напишете програма, която чете  $n$ -на брой цели числа ( $n > 0$ ), въведени от потребителя, 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

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.

#### Решение

```
namespace MaxNumber
```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var n = int.Parse(Console.ReadLine());  
            var max = -1000;  
            for (int i = 0; i < n; i++)  
            {  
                var next = int.Parse(Console.ReadLine());  
                if (next > max) max = next;  
            }  
            Console.WriteLine(max);  
        }  
    }  
}
```

#### Задача 4.6. Най-малко число

Напишете програма, която чете **n**-на брой цели числа ( $n > 0$ ), въведени от потребителя, 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

Подсказки: задачата е абсолютно аналогична с предходната.

#### Решение

```
namespace MinNumber
```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {
```



```
var n = int.Parse(Console.ReadLine());  
var min = 1000;  
for (int i = 0; i < n; i++)  
{  
    var next = int.Parse(Console.ReadLine());  
    if (next < min) min = next;  
}  
Console.WriteLine(min);  
}  
}
```

#### Задача 4.7. Лява и дясна сума

Да се напише програма, която чете  $2 \cdot n$ -на брой цели числа, подадени от потребителя, 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	Вход	изход	коментар
2 10 90 60 40	Yes, sum = 100	10+90 = 60+40 = 100	2 90 9 50 50	No, diff = 1	90+9 ≠ 50+50 Difference =  99-100  = 1

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.

*Решение*

```
namespace LeftRightSum  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var n = int.Parse(Console.ReadLine());  
  
            var leftSum = 0;  
            for (var i = 0; i < n; i++)  
            {  
                leftSum += int.Parse(Console.ReadLine());  
            }  
        }  
    }  
}
```



```
    }

    var rightSum = 0;
    for (var i = 0; i < n; i++)
    {
        rightSum += int.Parse(Console.ReadLine());
    }

    if (leftSum == rightSum)
    {
        Console.WriteLine("Yes, sum = " + leftSum);
    }
    else
    {
        Console.WriteLine("No, diff = " + Math.Abs(rightSum - leftSum));
    }
}
}
```

#### Задача 4.8. Четна / нечетна сума

Да се напише програма, която чете n-на брой цели числа, подадени от потребителя, 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 60 20	Yes Sum = 70	10+60 = 50+20 = 70

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

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

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).

#### Решение

```
namespace EvenOddSums
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
```





```
var oddSum = 0;
var evenSum = 0;
for (var i = 0; i < n; i++)
{
    var element = int.Parse(Console.ReadLine());
    if (i % 2 == 0) oddSum += element;
    else evenSum += element;
}
if (oddSum == evenSum)
{
    Console.WriteLine("Yes, sum = " + oddSum);
}
else
{
    Console.WriteLine("No, diff = " + Math.Abs(oddSum - evenSum));
}
}
```

#### Задача 4.9. Еднакви двойки

Дадени са  $2 \cdot n$ -на брой числа. Първото и второто формират двойка, третото и четвъртото също и т.н. Всяка двойка има стойност – сумата от съставлящите 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={Value}" + стойността. В противен случай отпечатайте "No, maxdiff={Difference}" + максималната разлика.

Примери:

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



0 0					
2 -1 0 0 -1	Yes, value=-1	стойности = {-1, -1} еднакви стойности	2 -1 2 0 -1	No, maxdiff= 2	стойности = {1, - 1} разлики = {2} макс. разлика = 2

### 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 павете в отделна променлива сумата на предходната двойка.
- Намерете най-голямата разлика между две двойки. Ако е 0, печатайте "Yes" иначе "No" + разликата.

### Решение

`namespace EqualCouples`

{

`class Program`

{

`static void Main(string[] args)`

{

`var n = int.Parse(Console.ReadLine());`

`var a = int.Parse(Console.ReadLine());`

`var b = int.Parse(Console.ReadLine());`

`var max = 0;`

`var sum = a + b;`

`var prev = sum;`

`for (int i = 0; i < n-1; i++)`

{

`a = int.Parse(Console.ReadLine());`

`b = int.Parse(Console.ReadLine());`

`sum = a + b;`

`if (max < Math.Abs(sum - prev))`

{

`max = Math.Abs(prev - sum);`

`prev = sum;`

}

}

`if (max == 0) Console.WriteLine("Yes, value=" + prev);`

`else Console.WriteLine("No, maxdiff=" + max);`

}

}

}



### Задача 4.10. Елемент, равен на сумата на останалите

Да се напише програма, която чете n-на брой цели числа, въведени от потребителя, и проверява дали сред тях съществува число, което е равно на сумата на всички останали. Ако има такъв елемент, печата "Yes", "Sum = " + неговата стойност; иначе печата "No", "Diff = " + 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	$1 + 2 + 3 = 6$
3 1 1 10	No Diff = 8	$ 10 - (1 + 1)  = 8$
3 5 5 1	No Diff = 1	$ 5 - (5 + 1)  = 1$
3 1 1 1	No Diff = 1	

**Подсказка:** изчислете сумата на всички елементи и най-големият от тях и проверете търсено

*Решение*

```
namespace SumOfTheOthers
{
    class Program
    {
        static void Main(string[] args)
        {
```



```
var n = int.Parse(Console.ReadLine());
var sum = 0;
var max = -1000000000000000000;
for (var i = 0; i < n; i++)
{
    var next = int.Parse(Console.ReadLine());
    sum += next;
    if (next > max) max = next;
}
if (sum - max == max)
{
    Console.WriteLine("Yes\nSum = " + max);
}
else
{
    Console.WriteLine("No\nDiff = " + Math.Abs(sum - 2 * max));
}
}
```

#### Задача 4.11. Правоъгълник от 10 X 10 звездички

Напишете програма, която чертае на конзолата правоъгълник от 10 x 10 звездички:

ВХОД	ИЗХОД
(няма)	***** ***** ***** ***** ***** ***** ***** ***** ***** *****

#### Решение

```
namespace RectangleOf10x10Stars
{
    class Program
    {
        static void Main(string[] args)
        {
            // v1
            for (int i = 0; i < 10; i++)
            {
                Console.WriteLine(new string('*', 10));
            }

            // v2
            for (int row = 0; row < 10; row++)
```



```
{  
    for (int col = 0; col < 10; col++)  
    {  
        Console.Write("*");  
    }  
    Console.WriteLine();  
}  
}  
}
```

#### Задача 4.12. Правоъгълник от N X N звездички

Напишете програма, която чете цяло положително число  $n$ , въведено от потребителя, и печата на конзолата правоъгълник от  $n * n$  звездички.

Примери:

ВХОД	ИЗХОД
2	** **

ВХОД	ИЗХОД
3	*** *** ***

ВХОД	ИЗХОД
4	**** **** **** ****

Подсказка: отпечатайте  $n$  звездички в цикъл  $n$  пъти, точно както в предната задача.

#### Решение

`namespace RectangleOfNxNStars`

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var n = int.Parse(Console.ReadLine());  
  
            // v1  
            for (int i = 0; i < n; i++)  
            {  
                Console.WriteLine(new string('*', n));  
            }  
  
            // v2  
            for (int row = 0; row < n; row++)  
            {  
                for (int col = 0; col < n; col++)  
                {  
                    Console.Write("*");  
                }  
                Console.WriteLine();  
            }  
        }  
    }  
}
```



}

### Задача 4.13. Квадрат от звездички

Напишете програма, която чете число  $n$ , въведено от потребителя, и чертае квадрат от  $n * n$  звездички. Разликата с предходната задача е, че между всеки две звездички има по един интервал.

Примери:

ВХОД	ИЗХОД
2	* * * *

ВХОД	ИЗХОД
3	* * * * * * * * *

ВХОД	ИЗХОД
4	* * * * * * * * * * * * * * * *

Подсказка: завъртете два вложени цикъла. Може да си помогнете с кода по-долу (част от кода е нарочно замъглен, за да не преписвате механично, а да се замислите малко):

#### Решение

```
namespace SquareOfNxNStars
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            for (int r = 1; r <= n; r++)
            {
                Console.Write("*");
                for (int c = 1; c < n; c++)
                {
                    Console.Write(" *");
                }
                Console.WriteLine();
            }
        }
    }
}
```

### Задача 4.14. Триъгълник от долари

Да се напише програма, която чете число  $n$ , въведено от потребителя, и печата триъгълник от долари като в примерите:

ВХОД	ИЗХОД
2	\$ \$ \$

ВХОД	ИЗХОД
3	\$ \$ \$ \$ \$ \$

ВХОД	ИЗХОД
4	\$ \$ \$ \$ \$ \$ \$ \$ \$ \$

ВХОД	ИЗХОД
5	\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$



Подсказка: завъртете два вложени цикъла: за първия **row = 1 ... n**; за втория **col = 1 ... row**.

*Решение*

```
namespace TriangleOfDollar
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            for (int r = 1; r <= n; r++)
            {
                Console.Write("$");
                for (int c = 1; c < r; c++)
                {
                    Console.Write(" $");
                }
                Console.WriteLine();
            }
        }
    }
}
```

#### Задача 4.15. Квадратна рамка

Напишете програма, която чете цяло положително число **n**, въведено от потребителя, и чертае на конзолата квадратна рамка с размер **n \* n** като в примерите по-долу:

ВХОД	ИЗХОД
3	<pre> + - +   -   + - + </pre>

ВХОД	ИЗХОД
4	<pre> + - - +   - -     - -   + - - + </pre>

ВХОД	ИЗХОД
5	<pre> + - - - +   - - -     - - -     - - -   + - - - + </pre>

ВХОД	ИЗХОД
6	<pre> + - - - - +   - - - -     - - - -     - - - -     - - - -   + - - - - + </pre>

Подсказки:

- Отпечатайте горната част: знак "+", n-2 пъти знак "-", знак "+".
- Отпечатайте средната част: в цикъл n-2 пъти печатайте знак "|", n-2 пъти знак "-", знак "|".
- Отпечатайте долната част: знак "+", n-2 пъти знак "-", знак "+".

*Решение*

```
namespace SquareFrame
{
    class Program
    {
        static void Main(string[] args)
        {
```



```
var n = int.Parse(Console.ReadLine());

// Top
Console.Write("+");
for (int i = 0; i < n - 2; i++) Console.Write(" -");
Console.WriteLine(" +");

// Middle
for (int row = 0; row < n - 2; row++)
{
    Console.Write("|");
    for (int i = 0; i < n - 2; i++) Console.Write(" -");
    Console.WriteLine(" |");
}

// Bottom
Console.Write("+");
for (int i = 0; i < n - 2; i++) Console.Write(" -");
Console.WriteLine(" +");
}
}
```

#### Задача 4.16. Ромбче от звездички

Напишете програма, която чете цяло положително число **n**, въведено от потребителя, и печата ромбче от звездички с размер **n** като в примерите по-долу:

ВХОД	ИЗХОД
1	*

ВХОД	ИЗХОД
2	* * * *

ВХОД	ИЗХОД
3	* * * * * * * * *

ВХОД	ИЗХОД
4	* * * * * * * * * * * * * * * *

#### Подсказки:

- Разделете ромба на горна и долна част и ги печатайте с два отделни цикъла.
- За горната част завъртете цикъл за **row** от **1** до **n**:
  - Отпечатайте **n-row** интервала.
  - Отпечатайте "\*".
  - Отпечатайте **row-1** пъти " \*".
- Долната част отпечатайте аналогично на горната с цикъл от **1** до **n-1**.

#### Решение

`namespace DiamondOfStars`

{





```
class Program
{
    static void Main(string[] args)
    {
        var n = int.Parse(Console.ReadLine());

        // Top
        for (var row = 1; row <= n; row++)
        {
            for (var col = 1; col <= n - row; col++)
                Console.Write(" ");
            Console.Write("*");
            for (var col = 1; col < row; col++)
                Console.Write(" *");
            Console.WriteLine();
        }

        // Bottom
        for (var row = n-1; row >= 1; row--)
        {
            for (var col = 1; col <= n - row; col++)
                Console.Write(" ");
            Console.Write("*");
            for (var col = 1; col < row; col++)
                Console.Write(" *");
            Console.WriteLine();
        }
    }
}
```

#### Задача 4.17. Коледна елха

Напишете програма, която чете число  $n$  ( $1 \leq n \leq 100$ ), въведено от потребителя, и печата коледна елха с размер  $n$  като в примерите по-долу:

ВХОД	ИЗХОД
1	<pre>             *   *</pre>

ВХОД	ИЗХОД
2	<pre>             *   *     **  **</pre>

ВХОД	ИЗХОД
3	<pre>             *   *     **  **     ***  ***</pre>

ВХОД	ИЗХОД
4	<pre>             *   *     **  **     ***  ***     ****  ****</pre>

#### Подсказки:

- В цикъл за  $i$  от 0 до  $n$  печатайте (за лявата част на елхата):
  - $n-i$  интервала;  $n$  звездички; вертикална черта.
- Аналогично довършете дясната част на елхата.

#### Решение

```
namespace ChristmasTree
{
    class Program
```



```
{
    static void Main(string[] args)
    {
        var n = int.Parse(Console.ReadLine());
        for (int i = 0; i <= n; i++)
        {
            var stars = new string('*', i);
            var spaces = new string(' ', n - i);
            Console.Write(spaces);
            Console.Write(stars);
            Console.Write(" | ");
            Console.Write(stars);
            Console.WriteLine(spaces);
        }
    }
}
```

#### Задача 4.18. Слънчеви очила

Напишете програма, която чете цяло число  $n$  ( $3 \leq n \leq 100$ ), въведено от потребителя, и печата слънчеви очила с размер  $5*n \times n$  като в примерите:

ВХОД	ИЗХОД
3	*****      ***** *////*   *////* *****      *****
4	*****      ***** *////////*   *////////* *////////*      *////////* *****      *****
5	*****      ***** *////////*      *////////* *////////*   *////////* *////////*      *////////* *****      *****

#### Подсказки:

- Отпечатайте най-горния ред от очилата:
  - $2*n$  звездички;  $n$  интервала;  $2*n$  звездички
- Отпечатайте средните  $n-2$  реда:
  - звездичка;  $2*n-2$  наклонени черти; звездичка;  $n$  интервала; звездичка;  $2*n-2$  наклонени черти; звездичка
  - когато редът е  $(n-1) / 2 - 1$ , печатайте  $n$  вертикални черти вместо  $n$  интервала
- Отпечатайте най-долния ред от очилата:
  - $2*n$  звездички;  $n$  интервала;  $2*n$  звездички



### Решение

```
namespace SunGlasses
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());

            // Top
            Console.WriteLine( new string('*', 2 * n) + new string(' ', n) + new
string('*', 2 * n));

            // Middle
            for (int i = 0; i < n-2; i++)
            {
                var glass = "*" + new string('/', 2 * n - 2) + "*";
                Console.WriteLine(glass + new string('|', n) + glass);
            }

            // Bottom
            Console.WriteLine(new string('*', 2 * n) + new string(' ', n) + new
string('*', 2 * n));
        }
    }
}
```

### Задача 4.19. Къщичка

Напишете програма, която чете число  $n$  ( $2 \leq n \leq 100$ ), въведено от потребителя, и печата къщичка с размер  $n \times n$ :

ВХОД	ИЗХОД
2	** 

ВХОД	ИЗХОД
3	-*- ***  *

ВХОД	ИЗХОД
4	-**_ ****  **   **

ВХОД	ИЗХОД
5	--*-- -***- *****  ***   ***

ВХОД	ИЗХОД
6	--**-- -****- *****  ****   ****   ****

### Подсказки:

- Отпечатайте в цикъл покрива на къщичката:
  - Той съдържа  $(n + 1) / 2$  реда.
  - На първия си ред съдържа 1 звездичка при нечетно  $n$  или 2 звездички при четно  $n$ .
  - На всеки следващ ред съдържа с 2 звездички повече.
- Отпечатайте в цикъл основата на къщичката:  $n / 2 - 1$  реда.

### Решение

```
namespace House
{
```



```
class Program
{
    static void Main(string[] args)
    {
        var n = int.Parse(Console.ReadLine());
        var stars = 1;
        if (n % 2 == 0) stars++;
        for (int i = 0; i < (n + 1) / 2; i++)
        {
            // Draw the roof
            var padding = (n - stars) / 2;
            Console.Write(new string('-', padding));
            Console.Write(new string('*', stars));
            Console.WriteLine(new string('-', padding));
            stars = stars + 2;
        }
        for (int i = 0; i < n / 2; i++)
        {
            // Draw the house body
            Console.Write("|");
            Console.Write(new string('*', n - 2));
            Console.WriteLine("|");
        }
    }
}
```

#### Задача 4.20. Диамант

Напишете програма, която чете цяло число  $n$  ( $1 \leq n \leq 100$ ), въведено от потребителя, и печата диамант с размер  $n$  като в примерите по-долу:

ВХОД	ИЗХОД
1	*

ВХОД	ИЗХОД
2	**

ВХОД	ИЗХОД
3	-*- *_* _*-

ВХОД	ИЗХОД
4	-**- *_-_* _**-

ВХОД	ИЗХОД
5	--*-- _ *_ _ *_ _ _* _ *_ _ --*--

ВХОД	ИЗХОД
6	--**-- _ *_ _ *_ _ _* _ *_ _ --**--

ВХОД	ИЗХОД
7	---*--- _ *_ _ _ *_ _ *_ _ _* _ *_ _ ---*---

ВХОД	ИЗХОД
8	----**-- _ *_ _ _ *_ _ *_ _ _* _ *_ _ ----**--

ВХОД	ИЗХОД
9	-----*----- _ *_ _ _ *_ _ *_ _ _* _ *_ _ -----*----- _ *_ _ _ *_ _ -----*-----



### Подсказки:

- Всички редове съдържат точно по  $n$  символа.
- Първият ред съдържа отляво и отясно точно  $\text{leftRight} = (n - 1) / 2$  тирета.
- Всеки следващ ред до средния съдържа отляво и отясно с 1 тире по-малко от предходния.
- Всеки следващ ред след средния съдържа отляво и отясно с 1 тире повече от предходния.
- Всеки ред съдържа в средата си (във вътрешността на диаманта)  $\text{mid} = n - 2 * \text{leftRight} - 2$  тирета.
- Всеки ред съдържа 2 звездички, освен когато  $\text{mid}$  е отрицателно (тогава има само 1 звездичка).
- За всеки ред може да се изчислят и отпечатаат неговите 5 съставни части:
  - $\text{leftRight}$  тиренца отляво
  - 1 звездичка
  - $\text{mid}$  тиренца в средата (когато  $\text{mid} \geq 0$ )
  - 1 звездичка (когато  $\text{mid} \geq 0$ )
  - 1 звездичка
  - $\text{leftRight}$  тиренца отляво

### Решение

```
namespace Diamond
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());

            // Draw the top part
            var leftRight = (n - 1) / 2;
            for (int i = 1; i <= (n - 1) / 2; i++)
            {
                Console.Write(new string('-', leftRight));
                Console.Write("*");
                var mid = n - 2 * leftRight - 2;
                if (mid >= 0)
                {
                    Console.Write(new string('-', mid));
                    Console.Write("*");
                }
                Console.WriteLine(new string('-', leftRight));
                leftRight--;
            }

            // Draw the bottom part
            leftRight = 0;
            for (int i = (n - 1) / 2; i >= 0; i--)
            {
                Console.Write(new string('-', leftRight));
                Console.Write("*");
```



```
        var mid = n - 2 * leftRight - 2;  
        if (mid >= 0)  
        {  
            Console.WriteLine(new string('-', mid));  
            Console.WriteLine("*");  
        }  
        Console.WriteLine(new string('-', leftRight));  
        leftRight++;  
    }  
}  
}
```

#### Задача 4.21. Числата от 1 до N през 3

Напишете програма, която чете число **n**, въведено от потребителя, и отпечатва числата от 1 до **n** през 3 (със стъпка 3). Примери:

ВХОД	ИЗХОД
10	1 4 7 10

ВХОД	ИЗХОД
7	1 4 7

ВХОД	ИЗХОД
15	1 4 7 10 13

#### Решение

```
namespace NumbersFrom1toN
```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var n = int.Parse(Console.ReadLine());  
            for (int k = 1; k < n; k=k+3)  
            {  
                Console.WriteLine(k);  
            }  
        }  
    }  
}
```

#### Задача 4.22. Числата от N до 1 в обратен ред

Напишете програма, която чете цяло положително число **n**, въведено от потребителя, и печата числата от **n** до 1 в обратен ред (от най-голямото към най-малкото). Примери:

ВХОД	ИЗХОД
2	2 1

ВХОД	ИЗХОД
3	3 2 1

ВХОД	ИЗХОД
5	5 4 3 2 1



Подсказка: отпечатайте **n** звездички в цикъл **n** пъти, точно както в предната задача.

*Решение*

```
namespace NumbersFromNto1
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            for (int k = n; k > 0; k--)
            {
                Console.WriteLine(k);
            }
        }
    }
}
```

#### Задача 4.23. Числа – степени на 2

Напишете програма, която чете от конзолата цяло число **n**, въведено от потребителя, и печата числата от 1 до  $2^n$ . Примери:

ВХОД	ИЗХОД
3	1 2 4 8

ВХОД	ИЗХОД
4	1 2 4 8 16

ВХОД	ИЗХОД
5	1 2 4 8 16 32

Подсказка: завъртете **for**-цикъл от 0 до **n** и започвайки от **num = 1** на всяка стъпка умножавайте **num** по 2.

*Решение*

```
namespace NumbersPowerOf2
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            var num = 1;
            while (n > -1)
            {
                Console.WriteLine(num);
                num *= 2;
                n--;
            }
        }
    }
}
```



}

#### Задача 4.24. Четни степени на 2

Да се напише програма, която чете число  $n$ , въведено от потребителя, и печата четните степени на  $2 \leq 2^n$ :  $2^0, 2^2, 2^4, 2^8, \dots, 2^n$ . Примери:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
3	1 4	4	1 4 16	5	1 4 16	6	1 4 16 64	7	1 4 16 64

Подсказка: започнете от 1 и в цикъл умножавайте по 4 на всяка стъпка.

#### Решение

```
namespace EvenPowerOf2
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            var num = 1;
            while (n > -1)
            {
                Console.WriteLine(num);
                num *= 4;
                n -= 2;
            }
        }
    }
}
```

#### Задача 4.25. Редица числа $2K+1$

Напишете програма, която чете число  $n$ , подадено от потребителя, и отпечатва всички числа  $\leq n$  от редицата: 1, 3, 7, 15, 31, .... Всяко следващо число се изчислява като предишното число  $\cdot 2 + 1$ . Примери:

ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД	ВХОД	ИЗХОД
3	1 3	8	1 3 7	17	1 3 7 15	31	1 3 7 15 31

Подсказки:

- Започнете от `num = 1`.
- В цикъл докато `num` не стигне `n`, печатайте, го умножавайте по 2 и прибавяйте 1.





### Решение

```
namespace OrderOf2kPlus1
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            var num = 1;
            while (num <= n)
            {
                Console.WriteLine(num);
                num = 2 * num + 1;
            }
        }
    }
}
```

### Задача 4.26. Число в диапазона от 1 до 100

Напишете програма, която чете цяло положително число **n** в диапазона [1...100], въведено от потребителя. При въвеждане на число извън посочения диапазон, да се отпечата съобщение за грешка и потребителят да се подкани да въведе ново число. Примери:

ВХОД / ИЗХОД
Enter a number in the range [1...100]: 35 The number is: 35
Enter a number in the range [1...100]: 105 Invalid number! Enter a number in the range [1...100]: 0 Invalid number! Enter a number in the range [1...100]: -200 Invalid number! Enter a number in the range [1...100]: 77 The number is: 77

### Подсказки:

- Въведете число.
- Повтаряйте в цикъл докато числото е невалидно: отпечатайте грешка и въведете число отново.

### Решение

```
namespace NumberInRangeFrom1To100
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
```



```
var num = int.Parse(Console.ReadLine());  
while (num < 1 || num > 100)  
{  
    Console.WriteLine("Invalid number!");  
    num = int.Parse(Console.ReadLine());  
}  
Console.WriteLine("The number is: {0}", num);  
}  
}
```

#### Задача 4.27. Най-голям общ делител (НОД)

Напишете програма, която чете две цели положителни числа  $a$  и  $b$ , въведени от потребителя, и изчислява и отпечатва най-големият им общ делител (НОД). Примери:

ВХОД	ИЗХОД
24 16	8

ВХОД	ИЗХОД
67 18	1

ВХОД	ИЗХОД
15 9	3

ВХОД	ИЗХОД
100 88	4

ВХОД	ИЗХОД
10 10	10

Подсказка: имплементирайте алгоритъма на Евклид:

<https://bg.wikipedia.org/wiki/алгоритъм-на-Евклид>.

#### Решение

```
namespace GreatestCommonDivider
```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var a = int.Parse(Console.ReadLine());  
            var b = int.Parse(Console.ReadLine());  
            while (b != 0)  
            {  
                var old = b;  
                b = a % b;  
                a = old;  
            }  
            Console.WriteLine("GCD = {0}", a);  
        }  
    }  
}
```

#### Задача 4.28. Факториел

Напишете програма, която чете цяло число  $n$  ( $1 \leq n \leq 12$ ), въведено от потребителя, и изчислява и отпечатва  $n! = 1 * 2 * \dots * n$  ( $n$  факториел).

Примери:

ВХОД	ИЗХОД
5	120

ВХОД	ИЗХОД
6	720

ВХОД	ИЗХОД
10	3628800

ВХОД	ИЗХОД
1	1

ВХОД	ИЗХОД
2	2



**Подсказка:** в цикъл умножете числата от 1 до n.

*Решение*

```
namespace Factoriel
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            var fak = 1;
            var j = 1;
            do
            {
                fak = fak * j;
                j++;
            }
            while (j <= n);
            Console.WriteLine(fak);
        }
    }
}
```

**Задача 4.29. Сумиране на цифрите на число**

Напишете програма, която чете цяло число **num**, въведено от потребителя, и отпечатва **сумата от цифрите му**.

Примери:

Вход	Изход	Коментар
5634	18	6+6+3+4 = 18

Вход	Изход	Коментар
19	10	1+9 = 10

Вход	Изход
5	5

Вход	Изход
17151	15

**Подсказка:** в цикъл докато не стигнете до 0 сумирайте последната цифра на числото (**num % 10**) и го разделяйте след това на 10 (така изтривате последната му цифра).

*Решение*

```
namespace SumDigitsOfNumber
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            var sum = 0;
            do
            {
                sum += (n % 10);
                n /= 10;
            }
            while (n > 0);
        }
    }
}
```



```
        Console.WriteLine(sum);  
    }  
}
```

### Задача 4.30. Проверка за просто число

Напишете програма, която чете цяло число  $n$ , въведено от потребителя, и **проверява дали е просто число** (дали се дели само на себе си и на единица). Да се отпечата "Prime" или "Not prime". Примери:

Вход	Изход
2	Prime

Вход	Изход
3	Prime

Вход	Изход
4	Not Prime

Вход	Изход
5	Prime

Вход	Изход
7	Prime

Вход	Изход
1	Not Prime

Вход	Изход
0	Not Prime

Вход	Изход
-1	Not Prime

Вход	Изход
149	Prime

Подсказки:

- Ако числото е по-малко от 2, значи не е просто.
- Всяко друго число първоначално се приема за просто и се проверява в цикъл дали се дели на числата 2, 3, 4, ..., до корен квадратен от  $n$  (цялата част).

### Решение

```
namespace PrimeNumberCheck
```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var k = int.Parse(Console.ReadLine());  
            var prime = true;  
            for (int j = 2; j < k; j++)  
                if (k % j == 0)  
                    prime = false;  
            if (prime && k > 1) Console.WriteLine("Prime");  
            else Console.WriteLine("Not Prime");  
        }  
    }  
}
```

### Задача 4.31. Въвеждане на четно число (с обработка на грешен вход)

Напишете програма, която **чете четно число**, въведено от потребителя. Ако потребителят въведе **грешно число** (нечетно число или стринг, който не е цяло число), трябва да му излиза **съобщение за грешка** и да **въвежда отново**. Примери:

Вход / Изход
Enter even number: 34



Even number entered: 34

Enter even number: 35

The number is not even.

Enter even number: hello

Invalid number!

Enter even number: 12.85

Invalid number!

Enter even number: 3464232636536513

Invalid number!

Enter even number: 8

Even number entered: 8

### Подсказки:

- В цикъл въвеждайте число и проверявайте дали е четно. При коректно число излезте от цикъла.
- С `try { ... } catch { ... }` конструкцията прихванете грешните числа, които не могат да се обърнат в `int`.

### Решение

```
namespace EvenNumberWithCheck
```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            while (true)  
            {  
                try  
                {  
                    Console.WriteLine("Enter even number:");  
                    var n = int.Parse(Console.ReadLine());  
                    if (n % 2 == 0)  
                    {  
                        Console.WriteLine("Even number entered: " + n);  
                        break;  
                    }  
                    else Console.WriteLine("The number is not even.");  
                }  
                catch  
                {  
                    Console.WriteLine("Invalid number!");  
                }  
            }  
        }  
    }  
}
```



### Задача 4.32. Числа на Фибоначи

Напишете програма, която чете цяло число  $n$ , въведено от потребителя, и пресмята  **$n$ -тото число на Фибоначи**. Нулевото число на Фибоначи е 1, първото е също 1, а всяко следващо е сумата от предходните две.

Примери:

Вход	Изход	Вход	Изход	Вход	Изход	Вход	Изход	Вход	Изход
0	1	1	1	2	2	5	8	10	89

Подсказка:

- При  $n < 2$  отпечатайте 1.
- Започнете от  $f0=1$  и  $f1=1$  и в цикъл сумирайте последните две числа. Записвайте последните две числа след всяка стъпка в  $f0$  и  $f1$ .

### Решение

```
namespace FibonacciNumbers
```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var n = int.Parse(Console.ReadLine());  
            var a = 1;  
            var b = 1;  
            var c = a + b;  
            while (true)  
            {  
                if (n <= 0) break;  
                else n--;  
                a = b;  
                b = c;  
                c = a + b;  
            }  
            Console.WriteLine(a);  
        }  
    }  
}
```

### Задача 4.33. Пирамида от числа

Напишете програма, която чете цяло число  $n$ , въведено от потребителя, и отпечатава **пирамида от числа** като в примерите:

Вход	Изход	Вход	Изход	Вход	Изход	Вход	Изход
7	1 2 3 4 5 6 7	10	1 2 3 4 5 6 7 8 9 10	12	1 2 3 4 5 6 7 8 9 10 11 12	15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15



### Подсказка:

- С два вложени цикъла печатайте пирамида от числа: на първия ред едно число, на втория ред 2 числа, на третия ред 3 числа и т.н.
- В отделен брояч пазете колко числа сте отпечатали до момента (и кое е текущото число). Когато стигнете  $n$ , излезте внимателно от двата вложени цикъла с **break** или **return**.

### Решение

namespace PyramidOfNumbers

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var n = int.Parse(Console.ReadLine());  
            var m = 1;  
            for (int i = 1; i <= n; i++)  
            {  
                for (int j = 1; j <= i; j++)  
                {  
                    Console.Write(m+" ");  
                    m++;  
                    if (m > n) return;  
                }  
                Console.WriteLine();  
            }  
        }  
    }  
}
```

### Задача 4.34. Таблица с числа

Напишете програма, която чете цяло число  $n$ , въведено от потребителя, и отпечатава **таблица (матрица) от числа** като в примерите:

Вход	Изход
2	1 2 2 1

Вход	Изход
3	1 2 3 2 3 2 3 2 1

Вход	Изход
4	1 2 3 4 2 3 4 3 3 4 3 2 4 3 2 1

Вход	Изход
5	1 2 3 4 5 2 3 4 5 4 3 4 5 4 3 4 5 4 3 2 5 4 3 2 1

### Подсказка:

- С два вложени цикъла за row (ред) и col (колона) печатайте число по формулата  $num = row + col + 1$ .
- За долната дясна половина на таблицата ще се получат грешни резултати. Там използвайте формулата  $2*n - num$ .
- Как се сетихме за тези формули? Математическа досетливост: наблюдаваме числата, предполагаме каква е формулата, тестваме и ако не



се получи, измисляме друга формула и пак пробваме. В случая имаме различни формули за горната лява и долната дясна половина от матрицата.

### Решение

```
namespace MatrixOfNumbers
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            for (int row = 0; row < n; row++)
            {
                for (int col = 0; col < n; col++)
                {
                    var num = row + col + 1;
                    if (num > n) num = 2 * n - num;
                    Console.Write(num + " ");
                }
                Console.WriteLine();
            }
        }
    }
}
```

## Тема 5. Подпрограми

### Задача 5.1. Празна касова бележка

Да се напише метод, който печата празна касова бележка:

ВХОД	ИЗХОД
	CASH RECEIPT ----- Charged to _____ Received by _____ ----- © IT KARIERA

### Решение

```
namespace CashReceipt
{
    class Program
    {
        static void PrintHeader()
        {
            Console.WriteLine("CASH RECEIPT");
            Console.WriteLine("-----");
        }

        static void PrintBody()
        {
            Console.WriteLine("Charged to -----");
        }
    }
}
```





```
        Console.WriteLine("Received by_____");  
    }  
  
    static void PrintFooter()  
    {  
        Console.WriteLine("_____");  
        Console.WriteLine("\u00A9 SoftUni");  
    }  
  
    static void Main(string[] args)  
    {  
        PrintHeader();  
        PrintBody();  
        PrintFooter();  
    }  
}
```

### Задача 5.2. Знак на цяло число

Да се създаде метод, който печата знака на цяло число n.

ВХОД	ИЗХОД
2	The number 2 is positive.
-5	The number -5 is negative.
0	The number 0 is zero.

#### Решение

```
namespace WholeNumberSign  
{  
    class Program  
    {  
        static void NumberCheck(int num)  
        {  
            if (num < 0) Console.WriteLine("The number {0} is negative.", num);  
            if (num > 0) Console.WriteLine("The number {0} is positive.", num);  
            if (num == 0) Console.WriteLine("The number 0 is zero.");  
        }  
  
        static void Main(string[] args)  
        {  
            var num = int.Parse(Console.ReadLine());  
            NumberCheck(num);  
        }  
    }  
}
```

### Задача 5.3. Принтиране на триъгълник

Да се създаде метод, който принтира триъгълник, както е показано в примерите:

ВХОД	ИЗХОД
------	-------



3	1 1 2 1 2 3 1 2 1
4	1 1 2 1 2 3 1 2 3 4 1 2 3 1 2 1

#### Решение

```
namespace PrintTriangle
{
    class Program
    {
        static void Print(int n)
        {
            for (int k = 1; k <= n; k++)
            {
                for (int j = 1; j <= k; j++)
                    Console.Write("{0} ", j);
                Console.WriteLine();
            }

            for (int k = n - 1; k > 0; k--)
            {
                for (int j = 1; j <= k; j++)
                    Console.Write("{0} ", j);
                Console.WriteLine();
            }
        }

        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());
            Print(n);
        }
    }
}
```

#### Задача 5.4. Рисуване на запълнен квадрат

Нарисувайте на конзолата запълнен квадрат със страна n, както е показано в примера:

ВХОД	ИЗХОД
4	----- -V\\V\\- -V\\V\\- -V\\V\\-



	-----
--	-------

### Решение

```
namespace DrawFillRectangle
```

```
{  
    class Program  
    {  
        static void Print(int n)  
        {  
            Console.WriteLine(new string('-', n * 2));  
            for (int i = 0; i <= n - 2; i++)  
            {  
                Console.Write("-");  
                for (int j = 0; j <= (n - 2); j++)  
                    Console.Write("\\\\");  
                Console.WriteLine("-");  
            }  
            Console.WriteLine(new string('-', n * 2));  
        }  
  
        static void Main(string[] args)  
        {  
            var n = int.Parse(Console.ReadLine());  
            Print(n);  
        }  
    }  
}
```

### Задача 5.5. Лице на триъгълник

Да се напише метод, който изчислява лицето на триъгълник по дадени основа и височина и връща стойността му.

ВХОД	ИЗХОД
3 4	6

### Решение

```
namespace TriangleArea
```

```
{  
    class Program  
    {  
        static double Area(double a, double ha)  
        {  
            return (a * ha) / 2;  
        }  
  
        static void Main(string[] args)  
        {  
            var a = double.Parse(Console.ReadLine());  
            var ha = double.Parse(Console.ReadLine());  
            var s = Area(a, ha);  
            Console.WriteLine(s);  
        }  
    }  
}
```



}

### Задача 5.6. Степен на число

Да се напише метод, който изчислява и връща резултата от повдигането на число на дадена степен.

ВХОД	ИЗХОД
2 8	256
3 4	81

#### Решение

```
namespace PowerOfNumber
{
    class Program
    {
        static double RaiseToPower(double number, int power)
        {
            double result = 1;
            for (int i = 0; i < power; i++)
                result *= number;
            return result;
        }

        static void Main(string[] args)
        {
            var number = double.Parse(Console.ReadLine());
            var power = int.Parse(Console.ReadLine());
            Console.WriteLine(RaiseToPower(number, power));
        }
    }
}
```

### Задача 5.7. По-голямата от две стойности

Да се създаде метод GetMax(), който връща като резултат по-голямата от двете стойности. Стойностите могат да бъдат **int**, **char** или **string**.

ВХОД	ИЗХОД
int 2 16	16
char a z	z
string aaa bbb	aaa



### Решение

```
namespace MaxOf2Values
{
    class Program
    {
        static int GetMax(int a, int b)
        {
            return a > b ? a : b;
        }

        static char GetMax(char a, char b)
        {
            return a > b ? a : b;
        }

        static string GetMax(string a, string b)
        {
            if (a.CompareTo(b) > 0) return a;
            else return b;
        }

        static void Main(string[] args)
        {
            var type = Console.ReadLine();
            switch (type)
            {
                case "int" :
                {
                    var a = int.Parse(Console.ReadLine());
                    var b = int.Parse(Console.ReadLine());
                    Console.WriteLine(GetMax(a, b));
                    break;
                }
                case "char":
                {
                    var a = char.Parse(Console.ReadLine());
                    var b = char.Parse(Console.ReadLine());
                    Console.WriteLine(GetMax(a, b));
                    break;
                }
                case "string":
                {
                    var a = Console.ReadLine();
                    var b = Console.ReadLine();
                    Console.WriteLine(GetMax(a, b));
                    break;
                }
            }
        }
    }
}
```

### Задача 5.8. Умножение на четна и нечетна сума

Да се напише програма, която умножава сумата от всички четни цифри на число и сумата на всички нечетни цифри на същото число:



- Направете метод `GetMultipleOfEvensAndOdds()`
- Направете метод `GetSumOfEvenDigits()` и `GetSumOfOddDigits()`
- Използвайте `Math.Abs()` за негативните числа

ВХОД	ИЗХОД
-12345	54

*Решение*

`namespace EvenOddSumMultiplication`

```
{
    class Program
    {
        static double GetSumOfEvenDigits(int n)
        {
            double sum = 0;
            while (n > 0)
            {
                if ((n % 10) % 2 == 0) sum += n % 10;
                n /= 10;
            }
            return sum;
        }

        static double GetSumOfOddDigits(int n)
        {
            double sum = 0;
            while (n > 0)
            {
                if ((n % 10) % 2 != 0) sum += n % 10;
                n /= 10;
            }
            return sum;
        }

        static double GetMultipleOfEvensAndOdds(int n)
        {
            return GetSumOfEvenDigits(n) * GetSumOfOddDigits(n);
        }

        static void Main(string[] args)
        {
            var n = Math.Abs(int.Parse(Console.ReadLine()));
            Console.WriteLine(GetMultipleOfEvensAndOdds(n));
        }
    }
}
```

## Тема 6. Подготовка за изпит

### Задача 6.1. Обмен

Покрай нашумелите крипто валути Стефан инвеститорът решава да инвестира част от парите си в биткойн. Като всеки един инвеститор, Стефан иска да знае къде отива всеки един негов цент, но не иска да прави



сметките сам, защото те отнемат време, а за това време той може да прави пари някъде другаде. Поради тази причина Стефан решава да наеме вас за написването на програма, която да пресмята колко общо биткойн е купил с парите си и каква такса е платил за обмена. Това обаче не е всичко. Стефан е наясно, че труда не е безплатен и затова е обещал 10% от общия брой на биткойн в сметката си да бъдат за ваше възнаграждение.

Вашата задача е да напишете програма, която:

- Принтира на конзолата колко общо биткойн има Стефан след като плати таксата за транзакцията и вашето възнаграждение.
- Принтира на конзолата колко долара е таксата за обмен, която Стефан е платил. Като таксата се изчислява по следния начин:
  - $\text{Такса в биткойн} = \frac{\text{броят биткойн който може да бъде закупен} * (\text{сатоши за байт} * 1024)}{100\,000\,000}$
- Принтира на конзолата вашето възнаграждение в биткойн.

### Вход

От конзолата се четат 3 числа:

- На първия ред – щатските долари, които Стефан ще инвестира. Реално число в интервала [1.00...1 000 000.00]
- На втория ред – цената за един биткойн в щатски долари. Реално число в интервала [0.00... 20 000.00]
- На третия ред – сатоши за байт. Цяло число в интервала [1 ... 1000]

### Изход

На конзолата трябва да се отпечата следния изход:

- "Total bitcoin after expenses: {брой биткойни след всички разходи} BTC" – форматира се до петия знак след десетичната запетая.
- "Tax payed: {такса в долари} USD" – форматира се до втория знак след десетичната запетая.
- "Programmer`s payment: {комисионната изплатена на програмиста} BTC" – форматира се до петия знак след десетичната запетая.

### Примерен вход и изход

Вход	Изход	Обяснения
250000 10000 100	Total bitcoin after expenses: 22.47696 BTC Tax payed: 256.00 USD Programmer`s payment: 2.49744 BTC	Биткойн който може да бъде закупен = $250\,000 / 10\,000 = 25$ Такса за покупката на биткойн = $25 * (100 * 1024) / 100\,000\,000 = 0.0256$ Общо закупен биткойн = $25 - 0.0256 = 24.9744$



		<p>Заплата за програмиста =  <math>24.9744 * 10\% = 2.49744</math></p> <p>Такса в долари = <math>0.0256 * 10\ 000 = 256</math></p> <p>Оставащ биткойн след всички разходи: <math>24.9744 - 2.49744 = 22.47696</math></p>
Вход	Изход	
<p>10000 200 500</p>	<p>Total bitcoin after expenses:  44.76960 BTC  Tax paid: 51.20 USD  Programmer`s payment: 4.97440 BTC</p>	<p>Биткойн който може да бъде закупен: 50</p> <p>Такса за покупка: 0.256</p> <p>Общо закупен биткойн: 49.744</p> <p>Заплата за програмиста: 4.9744</p> <p>Такса в долари: 51.2</p> <p>Оставащ биткойн след всички разходи: 44.7696</p>

### Решение

namespace Transaction

```
{
    class Program
    {
        static void Main(string[] args)
        {
            double moneyInUsd = double.Parse(Console.ReadLine());
            double bitcoinPriceInUsd = double.Parse(Console.ReadLine());
            int satoshiPerByte = int.Parse(Console.ReadLine());
            double bitcoinsAvailableToBuy = moneyInUsd / bitcoinPriceInUsd;
            double totalTransactionCostInBtc = bitcoinsAvailableToBuy *
(satoshiPerByte * 1024) / 100000000;
            double bitcoinsBought = bitcoinsAvailableToBuy -
totalTransactionCostInBtc;
            double usdSpentOnTaxes = totalTransactionCostInBtc * bitcoinPriceInUsd;
            double programmersCommission = bitcoinsBought * (10.0 / 100);
            double bitcoinsAfterExpenses = bitcoinsBought - programmersCommission;

            Console.WriteLine($"Total bitcoin after expenses:
{bitcoinsAfterExpenses:F5} BTC");
            Console.WriteLine($"Tax paid: {usdSpentOnTaxes:F2} USD");
            Console.WriteLine($"Programmer`s payment: {programmersCommission:F5}
BTC");
        }
    }
}
```





## Задача 6.2. Алтернативни монети

Стефчо инвеститорът е закупил  $X$  биткойн на цена  $P$ . След известно време цената за един биткойн става  $T$  и той решава, че със спечелените пари от разликата в цената ще инвестира в етереум и нео. Стефчо иска да закупи  $E$  на брой етереум с цена равна на 7.5% от текущата цена за 1 биткойн и  $N$  на брой нео с цена равна на 40% от тази за 1 етереум.

Вашата задача е да напишете програма, която пресмята дали спечелените пари от разликата в цената на биткойн ще бъдат достатъчни, за да може Стефчо да закупи  $E$  на брой етереум и  $N$  на брой нео.

### Вход

Входът се чете от конзолата и се състои от 5 реда:

1.  $X$  - брой биткойн, които Стефчо е купил – реално число в интервала  $[1.00 \dots 10000.00]$ ;
2.  $P$  - началата цена за един биткойн – реално число в интервала  $[1.00 \dots 25000.00]$ ;
3.  $T$  - текущата цена за един биткойн – реално число в интервала  $[1.00 \dots 25000.00]$ ;
4.  $E$  - количество етереум, което Стефчо иска да закупи – реално число в интервала  $[0.00 \dots 10000.00]$ ;
5.  $N$  - количество нео, което Стефчо иска да закупи – реално число в интервала  $[0.00 \dots 10000.00]$ ;

### Изход

На конзолата трябва да се отпечата следното:

- Ако печалбата от разликата в цените е по-малка от нужните пари за инвестицията, принтираме:
  - "Stefcho does not have enough money to make this investment."
  - "He needs {недостигащи пари} more in profits." – като недостигащите пари трябва да бъдат закръглени до втория знак след десетичната запетая.
- Ако печалбата от разликата в цените е по-голяма или равна на нужните пари за инвестицията, принтираме:
  - "Stefcho bought {брой етереум} Ethereum at a price of {цена на етереум}" – броят и цената на закупения етерум трябва да бъдат закръглени до четвъртия знак след десетичната запетая.



- "Stefcho bought {брой нео} Neo at a price of {цена на нео}" – броят и цената на закупеното нео трябва да бъде закръглен до четвъртия знак след десетичната запетая.
- "Stefcho has {оставащи пари след инвестицията} profits left to spend." – оставащите пари трябва да бъдат закръглени до втория знак след десетичната запетая.

### Примерен вход и изход

Вход	изход	коментари
2 5000 10000 2 1	Stefcho bought 2.0000 Ethereum at a price of 750.0000 Stefcho bought 1.0000 Neo at a price of 300.0000 Stefcho has 8200.00 profits left to spend.	Печалба от разлика в цените за 1 биткойн = $2 * 10\,000 - 2 * 5000 = 10\,000$ Цена за 1 етереум = $10\,000 * 7.5\% = 750$ Цена за 1 нео = $750 * 40\% = 300$ Обща цена на инвестицията = $750 * 2 + 300 * 1 = 1850$ $(10\,000 > 1850) \Rightarrow$ Стефан може да направи инвестицията
2.5 6500. 1 7500. 6 3 4	Stefcho does not have enough money to make this investment. He needs 86.46 more in profits.	Печалба = 2501.25 Цена за 1 етереум = 562.545 Цена за 1 нео = 225.018 Цена на инвестицията = 2587.707 $(2501.707 < 2587.707) \Rightarrow$ Стефан няма да може да направи инвестицията

### Решение

namespace Altcoins

```
{
    class Program
    {
        static void Main(string[] args)
        {
            double bitcoinsBought = double.Parse(Console.ReadLine());
            double bitcoinBeginingRate = double.Parse(Console.ReadLine());
            double bitcoinsCurrentRate = double.Parse(Console.ReadLine());
            double desiredEther = double.Parse(Console.ReadLine());
            double desiredNeo = double.Parse(Console.ReadLine());

            double priceOfEther = bitcoinsCurrentRate * 0.075;
            double priceOfNeo = priceOfEther * 0.40;

            double beginingMoney = bitcoinBeginingRate * bitcoinsBought;
```



```
double currentMoney = bitcoinsCurrentRate * bitcoinsBought;

double profit = currentMoney - beginingMoney;

double totalPriceOfEther = desiredEther * priceOfEther;
double totalPriceOfNeo = desiredNeo * priceOfNeo;

double totalPriceInUsd = totalPriceOfEther + totalPriceOfNeo;

if (profit < totalPriceInUsd)
{
    double moneyNeeded = totalPriceInUsd - profit;
    Console.WriteLine("Stefcho does not have enough money to make this
investment.");
    Console.WriteLine($"He needs {moneyNeeded:F2} more in profits.");
}
else
{
    double moneyLeft = profit - totalPriceInUsd;
    Console.WriteLine($"Stefcho bought {desiredEther:F4} Ethereum at a
price of {priceOfEther:F4}");
    Console.WriteLine($"Stefcho bought {desiredNeo:F4} Neo at a price of
{priceOfNeo:F4}");
    Console.WriteLine($"Stefcho has {moneyLeft:F2} profits left to
spend.");
}
}
```

### Задача 6.3. Обменно Бюро

Вдъхновен от интереса на хората към крипто валутите, Стефчо инвеститорът решава да създаде специално обменно бюро, чрез което хората ще могат бързо и лесно да инвестират парите си в етереум(ETH), биткойн (BTC) или рипъл (XRP). Тъй като Стефчо иска да привлече повече хора, той урежда страхотни бонуси за всички, които ще обменят евро за крипто валути в неговото обменно бюро, но все пак всичко си има цена и затова минималният брой монети, които хората могат да обменят е:

- Минимална покупка на XRP -> 80;
- Минимална покупка на BTC -> 0.001;
- Минимална покупка на ETH -> 0.0099;

Обменния курс на валутите е:

- XRP 1 = EUR 0.22;
- BTC 1 = EUR 6400
- ETH 1 = EUR 250

Напишете програма, която:



- Приема стринг, който репрезентира желаната валута, в която потребителя иска да инвестира.
  - Като всичко друго освен ETH, XRP или BTC се счита за невалиден вход и следва да се принтира на конзолата „{въведената валута} is not supported.“.
- Приема число, което репрезентира еврото, което потребителят иска да депозира.
- Проверява дали еврото е повече от 1000:
  - Ако е, добавя бонус 10% към него.
- При избрана валута XRP:
  - Ако монетите са повече от 1000, но по-малко от 2500: Добавяме 5% към тях.
  - Ако монетите са повече или равни на 2500: Добавяме 10% към тях.
- При избрана валута BTC:
  - Ако монетите са повече от 10: Добавяме 2% към тях.

### Вход

- Първи ред – Валута в която потребителят иска да инвестира. – текст.
- Втори ред – Еврото, което потребителят ще инвестира в дадена валута. Реално число в интервала [1.00...1 000 000.00]

### Изход

На конзолата трябва да се отпечата следния изход:

- Ако еврото, което има потребителя:
  - Е достатъчно за да покрие минималния брой монети от дадена валута принтираме - "Successfully purchased {брой закупени монети} {вида на закупените монети}"
    - Броят на закупените монети се закръгля до третия знак след десетичната запетая.
  - Не е достатъчно за да покрие минималния брой монети от дадена валута принтираме – "Insufficient funds".
- Ако въведена валута е невалидна принтираме: „EUR to {въведената валута} is not supported.“

### Примерен вход и изход

Вход	Изход	Обяснения
XRP 2500	Successfully purchased 13750.000 XRP	Евро > 1000 => добавяме 10% към еврото и то става 2750 XRP монети = 2750 / 0.22 = 12 500



		Имаме повече от 2500 XRP монети => $12\,500 + 10\% = 13\,750$ Накрая принтираме информацията от обмена.
Вход	Изход	
Gosho 5000	EUR to Gosho is not supported.	Обменното бюро не работи с монети „Gosho“ и затова на конзолата се принтира „EUR to Gosho is not supported“.
Вход	Изход	
ETH 1	Insufficient funds	Монети които може да закупим с 1 евро = $1 / 250 = 0.004$  $0.004 < 0.0099$ => Тази поръчка за обмен няма да се изпълни, защото еврото което искаме да обменим е твърде малко и не покрива минималните изисквания за обмен в етереум.

### Решение

```
namespace ChangeDesk
{
    class Program
    {
        static void Main(string[] args)
        {
            var toCurrency = Console.ReadLine();
            var euro = double.Parse(Console.ReadLine());

            var minRippleAccount = 80.0;
            var minBtcAccount = 0.001;
            var minEthAccount = 0.0099;

            var xrpToEur = 0.22;
            var btcToEur = 6400.0;
            var ethToEur = 250;

            if (euro > 1000)
            {
```



```
        euro *= 1.10;
    }

    switch (toCurrency)
    {
        case "XRP":
        {
            var rippleCoins = euro / xrpToEur;
            if (rippleCoins < minRippleAccount)
            {
                Console.WriteLine("Insufficient funds");
                break;
            }
            if (rippleCoins > 1000 && rippleCoins < 2500)
            {
                rippleCoins *= 1.05;
            }
            else if (rippleCoins >= 2500)
            {
                rippleCoins *= 1.10;
            }
            Console.WriteLine($"Successfully purchased {rippleCoins:F3}
XRP");
            break;
        }
        case "BTC":
        {
            var bitcoinCoins = euro / btcToEur;
            if (bitcoinCoins < minBtcAccount)
            {
                Console.WriteLine("Insufficient funds");
                break;
            }
            if (bitcoinCoins > 10)
            {
                bitcoinCoins *= 1.02;
            }
            Console.WriteLine($"Successfully purchased {bitcoinCoins:F3}
BTC");
            break;
        }
        case "ETH":
        {
            var ethereumCoins = euro / ethToEur;
            if (ethereumCoins < minEthAccount)
            {
                Console.WriteLine("Insufficient funds");
                break;
            }
            Console.WriteLine($"Successfully purchased
{ethereumCoins:F3} ETH");
            break;
        }
        default: { Console.WriteLine($"EUR to {toCurrency} is not
supported."); break; }
```



```
}  
}  
}  
}
```

#### Задача 6.4. Монетен рейтинг

След създаването на крипто валутата ESTD, Стефан е заинтересован как се справя тя спрямо конкурентните монети. Стефан знае, че ESTD е в една лига с IOTA, NEO и Doge, но тази информация не му е достатъчна. Затова иска да му помогнете като напишете програма тип анкета, която:

- Чете от конзолата число  $N$  – общия брой на участниците в анкетата;
- Следващите  $N$  реда – чете какъв вид крипто валута има всеки един участник и колко монети има от нея.
- Накрая изчислява каква част от целия пазар заема всяка една крипто валута, като за целта са ни дадени следните валутни курсове:
  - DOGE 1 = EUR 0.07
  - IOTA 1 = EUR 1.44
  - NEO 1 = EUR 28.50
  - ESTD 1 = EUR 25.0

#### Вход

Входът се чете от конзолата и съдържа:

- $N$  – цяло число в интервала  $[1...20]$ .
- $N$  – брой реда като в тях:
  1. Прочитаме каква валута притежава конкретния участник в анкетата - текст
  2. Колко монети има от нея – реално число в интервал  $[0.000001...10\ 000\ 000.00]$ .

#### Изход

Изходът се състои от 5 реда:

- "Total votes = {брой на гласове}, Money in market = {обща стойност на монетите на пазара} EUR" – като стойността трябва да бъде закръглена до втория знак след десетичната запетая.
- „DOGE's contribution: {колко процента от пазара е съставен от DOGE монети}%; People who use DOGE: {брой хора който използват DOGE}“
- „IOTA's contribution: {колко процента от пазара е съставен от IOTA монети}%; People who use IOTA: {брой хора който използват IOTA}“
- „NEO's contribution: {колко процента от пазара е съставен от NEO монети}%; People who use NEO: {брой хора който използват NEO}“



- „ESTD's contribution: {колко процента от пазара е съставен от ESTD монети}%; People who use ESTD: {брой хора който използват ESTD}“

### Забележка

Всички проценти трябва да бъдат **закръглени до втория знак** след десетичната запетая.

### Примерен вход и изход

ВХОД	ИЗХОД	Обяснения
4 NEO 3 ESTD 1 DOGE 50 IOTA 30	Total votes = 4, Money in market = 157.20 EUR DOGE's contribution: 2.23%; People who use DOGE: 1 IOTA's contribution: 27.48%; People who use IOTA: 1 NEO's contribution: 54.39%; People who use NEO: 1 ESTD's contribution: 15.90%; People who use ESTD: 1	Обща цена на монетите в евро = $3 \cdot 28.5 + 1 \cdot 25 + 50 \cdot 0.07 + 30 \cdot 1.44 = 157.2$  DOGE => $(3.5/157.2) \cdot 100 = 2.23\%$  IOTA => $(43.19/157.2) \cdot 100 = 27.48\%$  NEO => $(85.5/157.2) \cdot 100 = 54.39\%$  ESTD => $(25/157.2) \cdot 100 = 15.90\%$
ВХОД	ИЗХОД	Обяснения
2 ESTD 100 ESTD 150	Total votes = 2, Money in market = 6250.00 EUR DOGE's contribution: 0.00%; People who use DOGE: 0 IOTA's contribution: 0.00%; People who use IOTA: 0 NEO's contribution: 0.00%; People who use NEO: 0 ESTD's contribution: 100.00%; People who use ESTD: 2	Общо евро - 6250  DOGE => 0%  IOTA => 0%  NEO => 0%  ESTD => $(6250/6250) \cdot 100 = 100.00\%$

### Решение

namespace CoinRatings

```
{
    class Program
    {
        static void Main(string[] args)
        {
            var n = int.Parse(Console.ReadLine());

            var votesForDoge = 0;
            var votesForIOTA = 0;
            var votesForNeo = 0;
            var votesForESTD = 0;
```





```
var dogeToEur = 0.07;
var iotaToEur = 1.44;
var neoToEur = 28.50;
var estdToEur = 25.0;

var coinsInDoge = 0.0;
var coinsInIOTA = 0.0;
var coinsInNeo = 0.0;
var coinsInESTD = 0.0;

for (int i = 0; i < n; i++)
{
    var label = Console.ReadLine();
    var coins = double.Parse(Console.ReadLine());

    switch (label)
    {
        case "DOGE": coinsInDoge += coins; votesForDoge++; break;
        case "IOTA": coinsInIOTA += coins; votesForIOTA++; break;
        case "NEO": coinsInNeo += coins; votesForNeo++; break;
        case "ESTD": coinsInESTD += coins; votesForESTD++; break;
    }
}

var totalDogeInEur = coinsInDoge * dogeToEur;
var totalIotaInEur = coinsInIOTA * iotaToEur;
var totalNeoInEur = coinsInNeo * neoToEur;
var totalEstdInEur = coinsInESTD * estdToEur;

var totalMarketCapInEur = totalDogeInEur + totalIotaInEur +
totalNeoInEur + totalEstdInEur;

var totalVotesForCategories = votesForDoge + votesForESTD + votesForIOTA
+ votesForNeo;

var dogePercentage = totalDogeInEur / totalMarketCapInEur * 100;
var iotaPercentage = totalIotaInEur / totalMarketCapInEur * 100;
var neoPercentage = totalNeoInEur / totalMarketCapInEur * 100;
var estdPercentage = totalEstdInEur / totalMarketCapInEur * 100;

Console.WriteLine($"Total votes = {totalVotesForCategories}, Money in
market = {totalMarketCapInEur:F2} EUR");

Console.WriteLine($"DOGE's contribution: {dogePercentage:F2}%; People
who use DOGE: {votesForDoge}");
Console.WriteLine($"IOTA's contribution: {iotaPercentage:F2}%; People
who use IOTA: {votesForIOTA}");
Console.WriteLine($"NEO's contribution: {neoPercentage:F2}%; People who
use NEO: {votesForNeo}");
Console.WriteLine($"ESTD's contribution: {estdPercentage:F2}%; People
who use ESTD: {votesForESTD}");
}
}
```



## Задача 6.5. Нова монета

Стефан инвеститорът реши да отвори своя собствена крипто валута наречена ESTD(Easier Said Than Done) и има нужда от помощта ви. Помогнете на Стефан с програма, която ще генерира логото на неговата крипто валута в различни размери според въведеното число N.

### Подсказки

Ако числото е четно – средните редове са N – 1.

Ако числото е нечетно – средните редове са N – 2.

### Вход

Входът се чете от конзолата и съдържа само едно число:

- N – цяло число в интервала [2...27].

### Изход

Да се отпечата на конзолата лого, точно както в примерите.

### Примерен вход и изход

ВХОД	ИЗХОД	ВХОД	ИЗХОД
3	<pre>   \\\\/   \\-----//   \-----/   ~~/// ESTD \\~~   /-----\   //-----\\   ///\\ </pre>	4	<pre>   \\\\/   \\-----//   \\-----//   \-----/   ---#####---    ~~/// ESTD \\~~    ---#####---   /-----\   //-----\\   ///-----\\   ///\\ </pre>
ВХОД	ИЗХОД	ВХОД	ИЗХОД
6	<pre>   \\\\/   \\-----//   \\-----//   \\-----//   \\-----//   \-----/   ---#####---   -    ---#####---   -    ~~/// ESTD  \\\\/   ---#####---   -  </pre>	5	<pre>   \\\\/   \\-----//   \\-----//   \\-----//   \-----/   ---#####---   -    ~~/// ESTD  \\\\/   ---#####---   -   /-----\   //-----\\   ///-----\\ </pre>



	<pre>  -----#####-----   /-----\ //-----\\ ///-----\\\ ////-----\\ /////-----\\ ////////\\ </pre>		<pre> ////-----\\ ///\\ </pre>
--	---	--	--------------------------------

## Решения

```
namespace NewCoin
{
    class Program
    {
        static void Main(string[] args)
        {
            int n = int.Parse(Console.ReadLine());

            var innerDashes = 0;
            var dashLeftRight = n;
            var spaces = n * 2;

            for (int i = 0; i < n; i++)
            {
                Console.Write(new string(' ', spaces));
                Console.Write(new string('\\', dashLeftRight));
                Console.Write(new string('-', innerDashes));
                Console.Write(new string('-', innerDashes));
                Console.Write(new string('/', dashLeftRight));
                Console.Write(new string(' ', spaces));
                Console.WriteLine();

                innerDashes += 3;
                dashLeftRight--;
                spaces -= 2;
            }

            for (int i = 1; i < n / 2; i++)
            {
                Console.Write("|");
                Console.Write(new string('-', n - 1));
                Console.Write(new string('#', n * 2));
                Console.Write(new string('#', n * 2));
                Console.Write(new string('-', n - 1));
                Console.WriteLine("|");
            }

            Console.Write("|");
            Console.Write(new string('~', n - 1));
            Console.Write(new string('/', n * 2 - 3));
            Console.Write(" ESTD ");
            Console.Write(new string('\\', n * 2 - 3));
        }
    }
}
```



```
Console.Write(new string('~', n - 1));
Console.WriteLine("|");

for (int i = 1; i < n / 2; i++)
{
    Console.WriteLine("|");
    Console.Write(new string('-', n - 1));
    Console.Write(new string('#', n * 2));
    Console.Write(new string('#', n * 2));
    Console.Write(new string('-', n - 1));
    Console.WriteLine("|");
}

for (int i = 0; i < n; i++)
{
    innerDashes -= 3;
    dashLeftRight++;
    spaces += 2;
    Console.Write(new string(' ', spaces));
    Console.Write(new string('/', dashLeftRight));
    Console.Write(new string('-', innerDashes));
    Console.Write(new string('-', innerDashes));
    Console.Write(new string('\\', dashLeftRight));
    Console.WriteLine();
}
}
```

## Задача 6.6. Генериране на кодове

Стефан, инвеститорът, реши да започне да генерира кодове за съмнителен бизнес с криптовалути. Вашата задача е да направите програма, която генерира кодове спрямо дадена парола.

Комбинациите са структурирани по следния начин:

- Състоят се от общо 6 символа - 3 цифри, 2 букви и една цифра точно в този ред, като:
  - Първи символ – число 0 – 9
  - Втори символ – число 0 – 9
  - Трети символ – число 0 – 9
  - Четвърти символ – буква 'a' – 'z'
  - Пети символ – буква 'a' – 'z'
  - Шести символ – число от 0 – 9
- Стойността на сбора на всички символи трябва да е равна на паролата.

### Вход

Входът се чете от конзолата и съдържа два реда:

- M – цяло число в интервала [190...300] – паролата, според която трябва да се принтират комбинациите.
- N – цяло число в интервала [1...100] – броят кодове, които трябва да се генерират.



### Изход

Да се принтират на конзолата всички комбинации, разделени с интервал, които отговарят на условията.

- Броят на генерираните кодове е в интервала [0... 100]
- Подредбата на кодовете е лексикографска (или възходяща)

### Примерен вход и изход

ВХОД	ИЗХОД	Обяснения
195 4	000aa1 000ab0 000ba0 001aa0	Започваме от самото начало на комбинациите: $0 + 0 + 0 + a(97) + a(97) + 0 = 194$ -> не е равно на M => не принтираме. Продължаваме да правим нови комбинации... $0 + 0 + 0 + a(97) + a(97) + 1 = 195$ -> първо съвпадение => принтираме ... други комбинации които не отговарят на изискванията ... $0 + 0 + 0 + a(97) + b(98) + 0 = 195$ -> второ съвпадение => принтираме ... $0 + 0 + 0 + b(98) + a(97) + 0 = 195$ трето съвпадение => принтираме ... $0 + 0 + 1 + a(97) + a(97) + 0 = 195$ четвърто съвпадение => принтираме След като броя на съвпадения стане N спираме програмата.
ВХОД	ИЗХОД	
196 7	000aa2 000ab1 000ac0 000ba1 000bb0 000ca0 001aa1	

### Решение

namespace GenerateCode

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            long passCode = long.Parse(Console.ReadLine());  
            int codesWanted = int.Parse(Console.ReadLine());  
  
            int count = 0;  
  
            for (int i = 0; i <= 9; i++)  
            {  
                for (int j = 0; j <= 9; j++)  
                {
```



```
for (int k = 0; k <= 9; k++)
{
    for (char l = 'a'; l <= 'z'; l++)
    {
        for (char m = 'a'; m <= 'z'; m++)
        {
            for (int n = 0; n <= 9; n++)
            {
                if (i + j + k + l + m + n == passCode)
                {
                    Console.WriteLine(" " + i + j + k + l + m + n
+ " ");
                    count++;
                }
                if (count == codesWanted)
                {
                    return;
                }
            }
        }
    }
}
}
```

### Задача 6.7. Номера

В автокъщата вече няма достатъчно номера за колите, за това Гошко ви е помолил да генерирате  $n$  на брой нови. Номерата се състоят от 3 части:

- Комбинация от букви в началото
- 4-цифрена комбинация от числа, отговаряща на следното условие:
  - Сборът от всички цифри трябва да е равен на произведението на първата и третата цифра, като от това произведение трябва да се извади броят желани номера.
- Комбинация от букви в края

#### Вход

Входът се състои от 3 реда:

- Комбинацията от букви в началото
- Комбинацията от букви в края
- Броят желани номера  $N$  – цяло число в интервала  $[0... 1000]$

#### Изход

Трябва да се принтират на конзолата всички последователни номера, отговарящи на условията, разделени с интервал.

- Броят на генерираните номера е в интервала  $[0... 1000]$
- Подредбата на номерата е лексикографска (или възходяща)



### Примерен вход и изход

Вход	Изход	Вход	Изход
CA BA 5	CA2070BA CA2081BA CA2092BA CA2180BA CA2191BA	A PK 12	A3081PK A3093PK A3180PK A3192PK A3291PK A3390PK A4062PK A4075PK A4088PK A4161PK A4174PK A4187PK
Обяснения			
$2 + 0 + 7 + 0 = 9$ $(2 * 7) - 5 = 9$ Номерът отговаря на условията			

### Решение

```
using System;
using System.Diagnostics.Metrics;

namespace LicencePlates
{
    class Program
    {
        static void Main(string[] args)
        {
            var firstPart = Console.ReadLine();
            var lastPart = Console.ReadLine();
            var platesWanted = int.Parse(Console.ReadLine());
            var counter = 0;

            for (int i = 0; i < 10; i++)
            {
                for (int j = 0; j < 10; j++)
                {
                    for (int k = 0; k < 10; k++)
                    {
                        for (int l = 0; l < 10; l++)
                        {
                            if (counter < platesWanted)
                            {
                                if (i + j + k + l == (i * k) - platesWanted)
                                {
                                    Console.Write(firstPart + i + j + k + l +
lastPart + " ");
                                    counter++;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```



## Задача 6.8. Препродажба

Вие сте нает от бизнесменът Гошко, притежаващ верига автокъщи. Вашата работа е да напишете програма, която пресмята печалбата от препродаването на даден автомобил.

Освен цената на колата, има допълнителни такси и данъци. (20% от цената на автомобила + 275лв. за разходи при покупката)

Всеки ден, в който автомобилът стои в автокъщата коства пари, за това Гошко решава да ги добавя към цената. (20лв. на ден)

Гошко иска на всяка кола да има печалба 15% от сумата след данъците.

### Вход

Входът се състои от 3 реда:

- Модел на автомобила
- Цена на автомобила – реално число в интервала [1.0... 100000.0]
- Броят дни, в които автомобила е стоял в автокъщата – реално число в интервала [1... 1000]

### Изход

На конзолата трябва да се принтират 2 реда:

- „The { car model } with initial price of { initial price } BGN will sell for {resell value} BGN“ – моделът на колата, каква му е началната цена и за колко може да се продаде
- „Profit: { profit } BGN“ – колко е печалбата

Всички реални числа трябва да са закръглени до втория знак.

### Примерен вход и изход

Вход	Изход
BMW 320xd 7000 12	The BMW 320xd with initial price of 7000.00 BGN will sell for 10252.25 BGN Profit: 1337.25 BGN
Обяснения	
7000лв. + (20% от 7000) + 275 = 8675лв. след данъците. Автомобилът е прекарал 12 дни в автокъщата, т.е. 12 * 20 = 240лв. 8675лв. + 240лв. = 8915лв. Печалба -> 15% от 8915лв. = 1337.25лв.	

Вход	Изход
------	-------





KIA Rio 2000 2	The KIA Rio with initial price of 2000.00 BGN will sell for 3122.25 BGN Profit: 407.25 BGN
----------------------	--

### Решение

```
namespace ResellProfitValue
{
    class Program
    {
        static void Main(string[] args)
        {
            var carModel = Console.ReadLine();
            var initialCarPrice = double.Parse(Console.ReadLine());
            var daysStayedInTheLot = int.Parse(Console.ReadLine());

            // 20% tax + 275 flat tax
            var additionalExpenses = (initialCarPrice * 0.2) + 275;

            // 20 per day
            var carPriceWithExpenses = initialCarPrice + additionalExpenses +
(daysStayedInTheLot * 20);

            // We want 15% profit
            var profit = carPriceWithExpenses * 0.15;

            Console.WriteLine("The {0} with initial price of {1:f2} BGN will sell
for {2:f2} BGN",
                            carModel, initialCarPrice, carPriceWithExpenses +
profit);
            Console.WriteLine("Profit: {0:f2} BGN", profit);
        }
    }
}
```

### Задача 6.9. Отстъпка

Гошко има много познати и роднини, които искат да си купят коли. Следващата ви задача е да направите програма, която проверява дали може да направи отстъпка от цената на колата. За да може да се направи отстъпка, трябва да са спазени следните условия:

- Състоянието на колата трябва да е добро (good).
- VIN номерът трябва да е по-малък от 90000 и да е четно число.
- Печалбата на колата, която е 15% от цената ѝ, трябва да е повече от 400лв, за да „не се мине“ Гошко.

### Вход

Входът се състои от 4 реда:

- Модел на колата
- VIN номер – цяло число в интервала [1000... 1000000]



- Състояние на колата ("good" или "bad")
- Цената на колата – реално числов интервала [1.0... 2000000.0]

### Изход

Вашата задача е да принтирате на конзолата следните редове:

- Ако може да се направи отстъпка:
  - „yes – {car model}“
  - „profit – {profit}“
- Ако не може да се направи отстъпка:
  - „no“
  - Ако състоянието е лошо – „The car is in bad condition“
  - Ако VIN номерът не е валиден – „VIN {vin number} is not valid“
  - Ако печалбата е по-малка от 400 – „Cannot make discount, profit too low – {profit}“
  - Може да се принтират повече от един ред от горепосочените.

Всички реални числа трябва да са закръглени до втория знак.  
Сравняването на текст в задачата трябва да е case-insensitive.



### Примерен вход и изход

Вход	Изход	Вход	Изход
bmw 330xd 31232 good 8000	yes - bmw 330xd profit - 1200.00	mercedes s500 5738922 bad 90000	no The car is in bad condition VIN 5738922 is not valid
Обяснения			
VIN номерът е по-малък от 90000 и е четно число, състоянието е добро и печалбата -> 1200лв. е повече от 400.			

### Решение

```
namespace CarDiscount
{
    class Program
    {
        static void Main(string[] args)
        {
            var carModel = Console.ReadLine();
            var vinNumber = int.Parse(Console.ReadLine());
            var condition = Console.ReadLine().ToLower();
            var priceOfCar = double.Parse(Console.ReadLine());

            double profit = priceOfCar * 0.15;

            bool isGoodCondition = false;
            bool isCorrectVin = false;
            bool canMakeDiscount = false;

            if (condition == "good")
            {
                isGoodCondition = true;
            }
            if (vinNumber < 90000 && vinNumber % 2 == 0)
            {
                isCorrectVin = true;
            }
            if (profit > 400)
            {
                canMakeDiscount = true;
            }

            if (isGoodCondition && isCorrectVin && canMakeDiscount)
            {
                Console.WriteLine("yes - {0}", carModel);
                Console.WriteLine("profit - {0:f2}", profit);
            }
            else
            {
            }
```



```
        Console.WriteLine("no");
        if (!isGoodCondition)
        {
            Console.WriteLine("The car is in bad condition");
        }
        if (!isCorrectVin)
        {
            Console.WriteLine("VIN {0} is not valid", vinNumber);
        }
        if (!canMakeDiscount)
        {
            Console.WriteLine("Cannot make discount, profit too low -
{0:f2}", profit);
        }
    }
}
```

### Задача 6.10. Сезонна продажба

Гошко знае, че цените на колите спадат и се покачват и за това ви възлага задачата да направите програма, която преценява дали си заслужава да се продаде дадена кола сега.

Печалбата варира според състоянието и типа кола:

	SUV	Sedan
perfect	30% от цената	25% от цената
good	20% от цената	15% от цената
bad	10% от цената	10% от цената

През зимата има промоция – безплатни зимни гуми към колата, които костват на Гошко 200лв. от печалбата.

#### Вход

Входът се състои от 6 реда:

- Моделът на колата
- Типът на колата ("sedan" или "suv")
- Сезонът ("winter" или "summer")
- Състоянието на колата ("perfect", "good" или "bad")
- Началната цена на колата – реално число в интервала [0.0... 2000000.0]
- Желаната печалба – реално число в интервала [0.0... 2000000.0]

#### Изход

Ако печалбата е по-голяма или равна на желаната печалба:

- "The profit on {car model} will be good – {profit} BGN"

Ако е печалбата е по-малка от желаната печалба:

- "The car is not worth selling now"



- "Need {X} more profit"

Всички реални числа трябва да се закръглят до втория знак.  
Сравняването на текст в задачата трябва да е case-insensitive.

*Примерен вход и изход*

Вход	Изход	Вход	Изход
bmw 330xd sedan winter perfect 20000 300	The profit on bmw 330xd will be good - 4800.00 BGN	audi q7 suv summer good 9000 2500	The car is not worth selling now Need 700.00 more profit
Обяснения			
Колата е sedan и състоянието и е perfect -> печалбата е 25% от цената = 5000лв. Сезонът е зима -> 5000 - 200 = 4800лв. Печалбата е по-голяма от желаната печалба (4800 > 300)			

*Решение*

```
namespace SeasonSale
{
    class Program
    {
        static void Main(string[] args)
        {
            var carModel = Console.ReadLine();
            var carType = Console.ReadLine().ToLower(); // sedan or suv
            var season = Console.ReadLine().ToLower(); // winter or summer
            var condition = Console.ReadLine().ToLower(); // perfect, good or bad
            var initialPrice = double.Parse(Console.ReadLine());
            var profitWanted = double.Parse(Console.ReadLine());

            var profit = 0.0;

            if (carType == "suv")
            {
                if (condition == "perfect")
                {
                    profit = initialPrice * 0.3; // perfect condition suv
                }
                else if (condition == "good")
                {
                    profit = initialPrice * 0.2; // good condition suv
                }
                else
            }
```



```
        {
            profit = initialPrice * 0.1; // bad condition suv
        }
    }
    else
    {
        if (condition == "perfect")
        {
            profit = initialPrice * 0.25; //perfect condition sedan
        }
        else if (condition == "good")
        {
            profit = initialPrice * 0.15; // good condition sedan
        }
        else
        {
            profit = initialPrice * 0.10; // bad condition sedan
        }
    }
    if (season == "winter")
    {
        profit -= 200; // winter tires
    }
    if (profit >= profitWanted)
    {
        Console.WriteLine("The profit on {0} will be good - {1:2f} BGN",
carModel, profit);
    }
    else
    {
        Console.WriteLine("The car is not worth selling now");
        Console.WriteLine("Need {0:f2} more profit", profitWanted - profit);
    }
}
}
```

### Задача 6.11. Онлайн обяви

Гошко е открил магическото нещо, наречено Интернет и е разбрал че може да се пускат обяви за коли там. Като добър бизнесмен, какъвто е, ви е наел спешно да напишете програма, която да приема n на брой коли и да публикува обяви за тях.

Категориите в сайта за обяви са:

- "sport" – при кола тип "coupe" и вид гориво "gasoline".
  - Ако цената на колата е над 100000лв. категорията става "supersport"
- "ecosport" – при кола тип "coupe" и вид гориво "diesel"
- "executive" – при кола тип "sedan" и вид гориво "gasoline"
  - Ако цената на колата е над 80000лв. категорията става "limousine"



- "economic" – при кола тип "sedan" и вид гориво "diesel"

При правенето на обявата може да се избере опция "vip", при която цената на колата се увеличава с 200лв.

Накрая трябва да се принтира процента на бензиновите и на дизеловите коли, които са въведени.

### Вход

- Цяло число n – броят коли, за които ще се публикуват обяви

За всяка кола:

- Модел на колата
- Тип на колата ("coupe" или "sedan")
- Тип на горивото ("gasoline" или "diesel")
- Статус на обявата ("vip" или "normal")
- Цена на колата – реално число в интервала [0.0... 2000000.0]
- Километри на колата – цяло число в интервала [0... 2000000]

### Изход

За всяка кола:

- Car model – {car model} – моделът на колата
- Category – {category} – категорията на колата
- Type – {type of car} – типът на колата ("coupe" или "sedan")
- Fuel – {type of fuel} – тип на гориво ("gasoline" или "diesel")
- Kilometers – {kilometers} – километрите на колата
- Price – {price} – цената на колата

След принтирането на всички обяви, да се отпечата:

- Gasoline cars: {percentage of gasoline cars} – процент на бензиновите коли
- Diesel cars: {percentage of diesel cars} – процент на дизеловите коли

Всички числа с плаваща запетая трябва да се форматираат до втория знак.

### Примерен вход и изход

Вход	Изход	Вход	Изход
------	-------	------	-------



2 bmw 330xd sedan diesel vip 12000.50 120000 mercedes S320 sedan gasoline normal 15000 100000	Car model - bmw 330xd Category - economic Type - sedan Fuel - diesel Kilometers - 120000 Price - 12200.50 Car model - mercedes S320 Category - executive Type - sedan Fuel - gasoline Kilometers - 100000 Price - 15000.00 Gasoline cars: 50.00% Diesel cars: 50.00%	3 Audi A4 sedan gasoline vip 2012.123 89000 mercedes sls coupe gasoline normal 120000 65291 opel astra coupe diesel vip 2500 200000	Car model - Audi A4 Category - executive Type - sedan Fuel - gasoline Kilometers - 89000 Price - 2212.12 Car model - mercedes Category - supersport Type - coupe Fuel - gasoline Kilometers - 65291 Price - 120000.00 Car model - opel astra Category - ecosport Type - coupe Fuel - diesel Kilometers - 200000 Price - 2700.00 Gasoline cars: 66.67% Diesel cars: 33.33%
Обяснения			
<p>Първата кола е седан, горивото ѝ е дизел и обявата е със стаус "vip" -&gt; категорията е "economic" и цената -&gt; 12000.50 + 200лв. = 12200.50лв.</p> <p>Следващата кола е седан, горивото ѝ е бензин и обявата е нормална -&gt; категорията е "executive" и цената остава същата - 15000лв.</p> <p>Бензиновите коли са <math>(1 / 2) * 100 = 50.00\%</math></p> <p>Дизеловите коли са <math>(1 / 2) * 100 = 50.00\%</math></p>			

### Решение

```
namespace CarAds
{
    class Program
    {
        static void Main(string[] args)
        {
            int n = int.Parse(Console.ReadLine());
            int gasolineCarsCount = 0;
            int dieselCarsCount = 0;

            for (int i = 0; i < n; i++)
```





diesel

```
{
    String carModel = Console.ReadLine().ToLower();
    String carType = Console.ReadLine().ToLower(); // coupe or sedan
    String fuelType = Console.ReadLine().ToLower(); // gasoline or

    String adStatus = Console.ReadLine().ToLower(); // normal or vip
    double price = double.Parse(Console.ReadLine());
    int kilometers = int.Parse(Console.ReadLine());

    String category = "";

    if (carType == "coupe")
    {
        if (fuelType == "gasoline")
        {
            category = "sport";
            if (price > 100000)
            {
                category = "supersport";
            }
            gasolineCarsCount++;
        }
        else
        {
            category = "ecosport";
            dieselCarsCount++;
        }
    }
    else
    {
        if (fuelType == "gasoline")
        {
            category = "executive";
            if (price > 80000)
            {
                category = "limousine";
            }
            gasolineCarsCount++;
        }
        else
        {
            category = "economic";
            dieselCarsCount++;
        }
    }

    if (adStatus == "vip")
    {
        price += 200;
    }

    Console.WriteLine("Car model - {0}", carModel);
    Console.WriteLine("Category - {0}", category);
    Console.WriteLine("Type - {0}", carType);
}
```



```

        Console.WriteLine("Fuel - {0}", fuelType);
        Console.WriteLine("Kilometers - {0}", kilometers);
        Console.WriteLine("Price - {0:f2}", price);
    }

    Console.WriteLine("Gasoline cars: {0:f2}", ((double)gasolineCarsCount /
n) * 100);
    Console.WriteLine("Diesel cars: {0:f2}", ((double)dieselCarsCount / n) *
100);
    }
}

```

### Задача 6.12. Гараж

Гошко иска да си направи гараж, където да държи някои от любимите си коли. Понеже не му се отделят пари да наеме проектант или архитект, дава задачата на вас. Трябва да се принтира гараж с широчина  $2 * n + 2$  и височина  $n + 2$  на конзолата както в примерите по-долу.

Широчината на вратата е винаги една и съща (4 символа). Височината ѝ е 1 или 2 реда, в зависимост колко редаваме имаме в края.

Ако имаме 2 или повече реда след "GARAGE", височината е 2 реда. Ако нямаме – един ред.

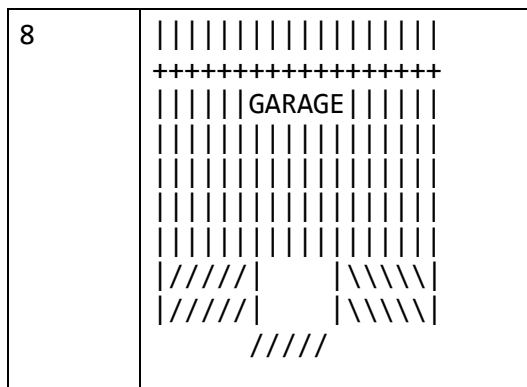
#### Вход

Входът се състои от 1 ред:

- Число  $n$  – цяло число в диапазона [3... 1000]

#### Примерен вход и изход

Вход	Изход	Вход	Изход
3	<pre>         +++++++  GARAGE           ///// </pre>	5	<pre>         +++++++   GARAGE                                 ///// </pre>
Вход	Изход		



### Решение

namespace Garage

{

class Program

{

private static String repeatStr(String str, int count)

{

String result = "";

for (int i = 0; i < count; i++)

{

result += str;

}

return result;

}

static void Main(string[] args)

{

int n = int.Parse(Console.ReadLine());

int width = 2 \* n + 2;

Console.WriteLine(repeatStr("|", width));

Console.WriteLine(repeatStr("+", width));

int sideSpaceCount = width / 2;

Console.WriteLine(repeatStr("|", sideSpaceCount - 3) + "GARAGE"  
+ repeatStr("|", sideSpaceCount - 3));

for (int i = 0; i < n - 2; i++)

{

if (i >= (n - 2) - 2)

{

Console.WriteLine("|"  
+ repeatStr("/", n - 3) + "|"  
+ repeatStr("\\", n - 3) + "|");

}

else

{

Console.WriteLine(repeatStr("|", width));

}

}



```
        Console.WriteLine(repeatStr(" ", n - 2) + "/////");  
    }  
}  
}
```



## Съдържание

Модул 1. Увод в програмирането.....	1
Тема 1. Въведение в програмирането .....	1
Задача 1.1. Конзолна програмка "Hello World!" .....	1
Тема 2. Пресмятания оператори изрази.....	1
Задача 2.1. Пресмятане на лице на квадрат .....	1
Задача 2.2. От инчове към сантиметри .....	1
Задача 2.3. Поздрав по име.....	2
Задача 2.4. Съединяване на текст и числа.....	2
Задача 2.5. Конвертор от °C към °F.....	3
Задача 2.6. Конвертор от радиани в градуси.....	3
Задача 2.7. Конвертор от USD към BGN.....	4
Задача 2.8. Лице на трапец.....	4
Задача 2.9. Периметър и лице на кръг.....	5
Задача 2.10. Лице на правоъгълник в равнината.....	5
Задача 2.11. Лице на триъгълник.....	6
Задача 2.12. Междувалутен конвертор.....	7
Задача 2.13. 1000 дни на Земята .....	8
Задача 2.14. Учебна зала.....	9
Задача 2.15. Зеленчукова борса.....	11
Задача 2.16. Ремонт на плочки.....	12
Задача 2.17. Парички .....	13
Задача 2.18. Дневна печалба.....	14
Тема 3. Проверки.....	16
Задача 3.1. Проверка за отлична оценка .....	16
Задача 3.2. Отлична оценка или не .....	16
Задача 3.3. Четно или нечетно число .....	17
Задача 3.4. Намиране на по-голямото число.....	17
Задача 3.5. Познай паролата .....	18
Задача 3.6. Еднакви думи.....	18
Задача 3.7. Време плюс 15 минути.....	19
Задача 3.8. Три еднакви числа .....	20
Задача 3.9. Изписване на число до 9 с думи.....	20



Задача 3.10. Бонус точки.....	21
Задача 3.11. Сумиране на секунди .....	22
Задача 3.12. Конвертор за мерни единици.....	23
Задача 3.13. Число от 100 до 200.....	24
Задача 3.14. Информация за скоростта .....	25
Задача 3.15. Лица на фигури.....	25
Задача 3.16. Изписване на число до 100 с думи.....	27
Задача 3.17. Обръщение според възраст и пол.....	27
Задача 3.18. Квартално магазинче .....	28
Задача 3.19. Точка в правоъгълник .....	29
Задача 3.20. Плод или зеленчук.....	31
Задача 3.21. Невалидно число .....	31
Задача 3.22. Точка върху страната на правоъгълник .....	32
Задача 3.23. Ден от седмицата.....	33
Задача 3.24. Клас животно .....	34
Задача 3.25. Кино .....	35
Задача 3.26. Цена за транспорт.....	36
Задача 3.27. Тръби в басейн.....	37
Задача 3.28. Поспаливата котка Том.....	39
Задача 3.29. Реколта .....	40
Задача 3.30. Навреме за изпит .....	42
Задача 3.31. Пътешествие.....	44
Задача 3.32. Операции между числа.....	45
Задача 3.33. Билети за мач.....	47
Задача 3.34. Хотелска стая.....	49
Тема 4. Повторения.....	51
Задача 4.1. Числа от 1 до 100 .....	51
Задача 4.2. Числа до 1000, завършващи на 7 .....	51
Задача 4.3. Всички латински букви .....	52
Задача 4.4. Сумиране на числа .....	52
Задача 4.5. Най-голямо число.....	53
Задача 4.6. Най-малко число .....	54
Задача 4.7. Лява и дясна сума.....	55



Задача 4.8. Четна / нечетна сума.....	56
Задача 4.9. Еднакви двойки .....	57
Задача 4.10. Елемент, равен на сумата на останалите.....	59
Задача 4.11. Правоъгълник от 10 X 10 звездички .....	60
Задача 4.12. Правоъгълник от N X N звездички.....	61
Задача 4.13. Квадрат от звездички.....	62
Задача 4.14. Триъгълник от долари .....	62
Задача 4.15. Квадратна рамка .....	63
Задача 4.16. Ромбче от звездички .....	64
Задача 4.17. Коледна елха.....	65
Задача 4.18. Слънчеви очила .....	66
Задача 4.19. Къщичка .....	67
Задача 4.20. Диамант .....	68
Задача 4.21. Числата от 1 до N през 3 .....	70
Задача 4.22. Числата от N до 1 в обратен ред.....	70
Задача 4.23. Числа – степени на 2 .....	71
Задача 4.24. Четни степени на 2 .....	72
Задача 4.25. Редица числа $2K+1$ .....	72
Задача 4.26. Число в диапазона от 1 до 100 .....	73
Задача 4.27. Най-голям общ делител (НОД).....	74
Задача 4.28. Факториел .....	74
Задача 4.29. Сумиране на цифрите на число .....	75
Задача 4.30. Проверка за просто число .....	76
Задача 4.31. Въвеждане на четно число (с обработка на грешен вход) ....	76
Задача 4.32. Числа на Фибоначи .....	78
Задача 4.33. Пирамида от числа.....	78
Задача 4.34. Таблица с числа.....	79
Тема 5. Подпрограми .....	80
Задача 5.1. Празна касова бележка.....	80
Задача 5.2. Знак на цяло число .....	81
Задача 5.4. Рисуване на запълнен квадрат .....	82
Задача 5.5. Лице на триъгълник.....	83
Задача 5.6. Степен на число .....	84



Задача 5.7. По-голямата от две стойности.....	84
Задача 5.8. Умножение на четна и нечетна сума .....	85
Тема 6. Подготовка за изпит.....	86
Задача 6.1. Обмен.....	86
Задача 6.2. Алтернативни монети .....	89
Задача 6.3. Обменно Бюро .....	91
Задача 6.4. Монетен рейтинг.....	95
Задача 6.5. Нова монета.....	98
Задача 6.6. Генериране на кодове.....	100
Задача 6.7. Номера.....	102
Задача 6.8. Препродажба .....	104
Задача 6.9. Отстъпка.....	105
Задача 6.10. Сезонна продажба.....	108
Задача 6.11. Онлайн обяви .....	110
Задача 6.12. Гараж.....	114