Циклични алгоритми. Оператор за цикъл с брояч.

1. Цикли

В езика C# има три оператора за цикъл – операторът for, оператора while и оператора while ….. do

2. Операторa for

Синтаксис:

for (<инициализация>;<условие>;<обновяване>)

{<тяло на цикъла>}

Изпълнението на оператора започва с изчисляване на израза<инициализация>. В него обикновенно се задава началната стойност на променливата, която управлява изпълнението на цикъла. След това започва повтарянето на следните три действия:

- Изчислява се логическия израз <условие>. Ако стойността му е false – изпълнението на оператора за цикъл се прекратява.

Ако стойността му е true, се продължава със следващите две стъпки:

- Изпълнява се блокът от оператори, наричан <тяло на цикъла>
- Изчислява се изразът <обновяване>. В него обиновенно се променя стойността на променливата. След това отново се изчслява израза <условие> и т.н.

Трте израза са поставени в кръгли скоби и са разделени един от друг с точка и запетая. Тялото на цикъла е блок от оператори и когато съдържа повече от един оператор трябва да се поставя в къдрави скоби, а когато се състои само от един оператор – тогава скобите не са задължителни.

Операторът for се среща в два варианта.

1-ви вариант:

for([<тип>]<променлива>=<начална стойност>;
<променлива><=<крайна стойност>;
<променлива>=<променлива>+<стъпка>)
{<блок от оператори>}

2-ри вариант:

for([<тип>]<променлива>=<начална стойност>;
<променлива>>=<крайна стойност>;
<променлива>=<променлива>-<стъпка>)
{<блок от оператори>}

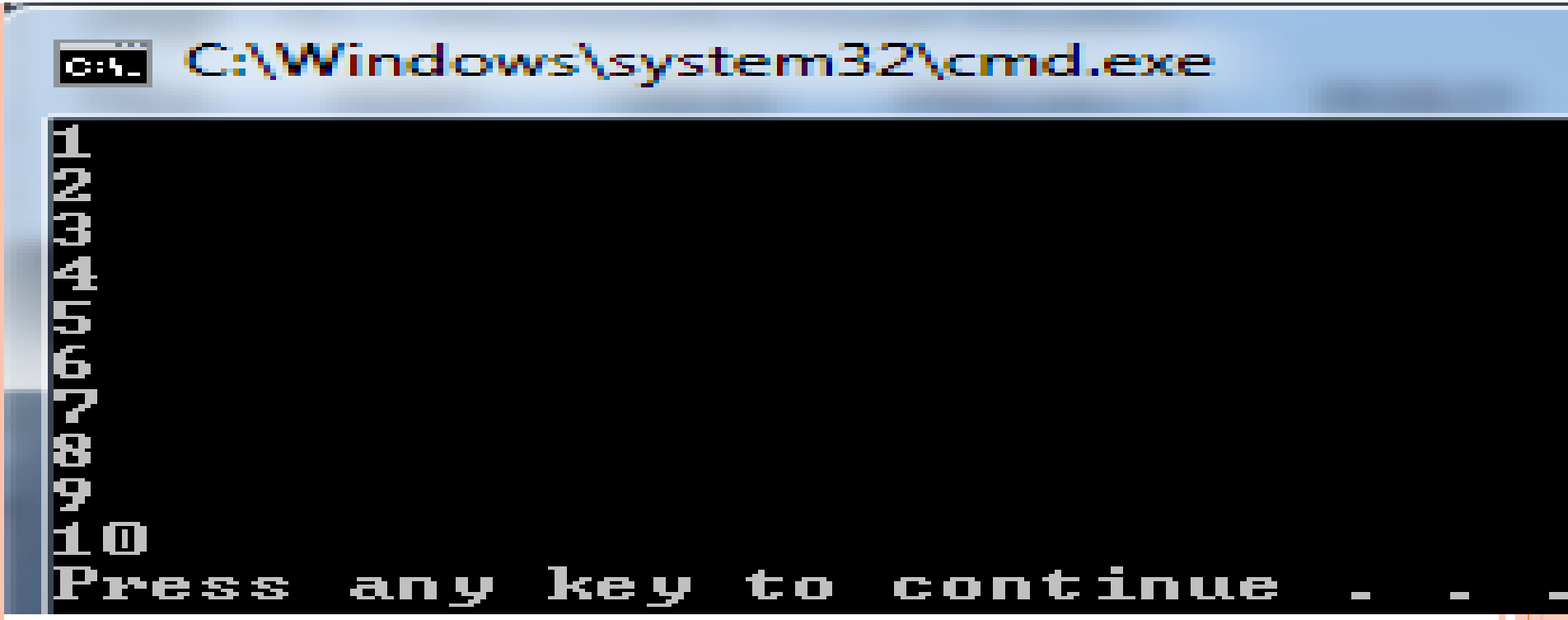
Променливата може да бъде от кой и да е числов тип. Началната стойност е произволна константа, която в първия вариант трябва да е по-малка или равна на крайната стойност, а във втори вариант – по-голяма или равна на крайната стойност. Типът на променливата не трябва да се задава, ако променливата е декларирана извън цикъла. Когато променливата е декларирана в цикъла тя е локална и не е използваема извън цикъла.

Да разгледаме действието на оператора във първия вариант. Изпълнението на цикъла започва с инициализацията на променливата на цикъла, на която се присвоява начална стойност. Проверява се дали е изпълнено условието, т. е. дали стойността на променливата е по-малка или равна на крайната стойност.

Ако е така се изпълнява блокът от оператори. След това променливата се увеличава със стъпката и всичко се повтаря – отново се проверява дали условието е изпълнено и ако е така отново се изпълнява блокът от оператори. Оператора спира действието си, когато променливата стане по-голяма от крайната стойност. Във вторият вариант действието е същото, с разликата, че променливата всеки път намалява със стъпката и оператора спира действието си, когато променливата стане по-малка от крайната стойност. Най-често използваме оператора for, при който началната и крайната стойност са цели числа, а стъпката е 1.

Пример1:

```
static void Main(string[] args)
{
    int i; for(i=1; i<=10; i++)
    { Console.WriteLine("{0}", i); }
}
```

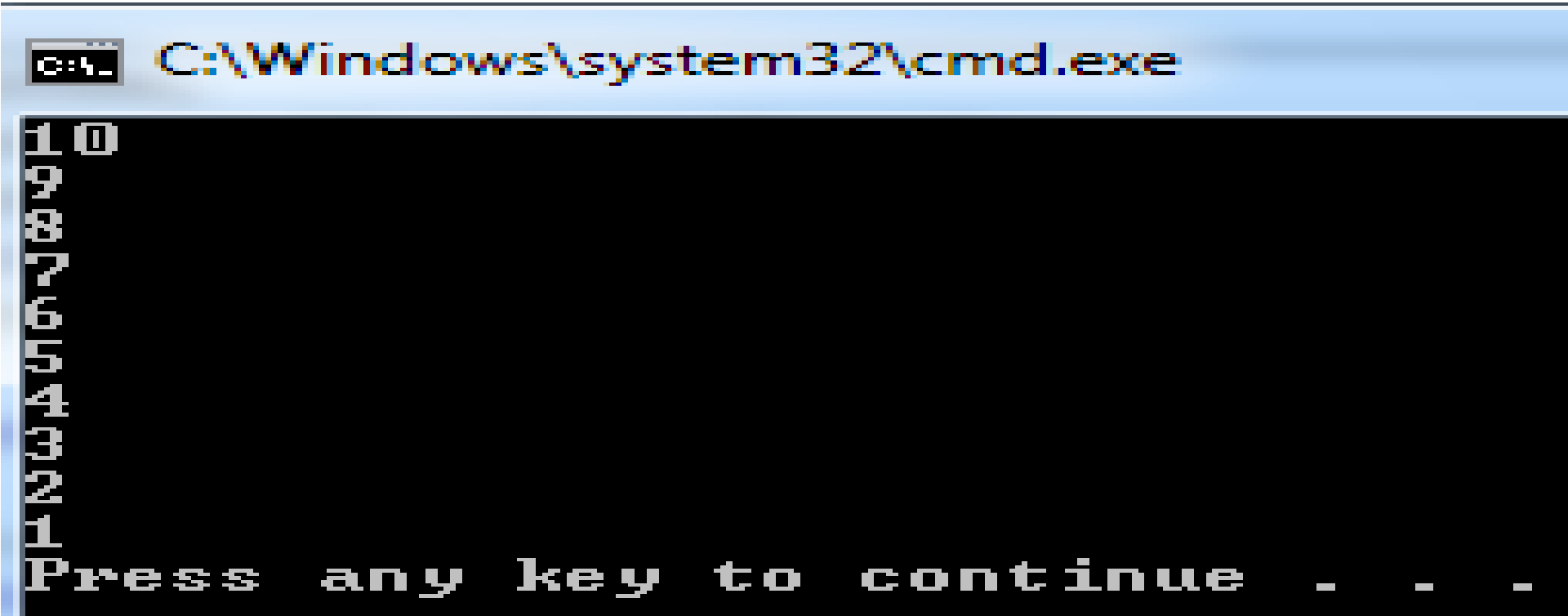


В този пример в тялото на цикъла променливата i ще приеме стойностите 1, 2, 3,.....,10

Когато i стане 10 изразът i++ се пресмята отново в следствие на което i става 11. Сравнението i<=10 вече не е истина, изпълнението на цикъла се прекратява, а стойността на i остава 11.

Пример2:

```
static void Main(string[] args)
{
    int i; for (i = 10; i >= 1; i--)
        Console.WriteLine("{0}", i);
}
```



В тялото на този цикъл променливата i ще приема стойности 10, 9, 8, ……,1. Стойността на i след свършване на цикъла ще бъде 0.

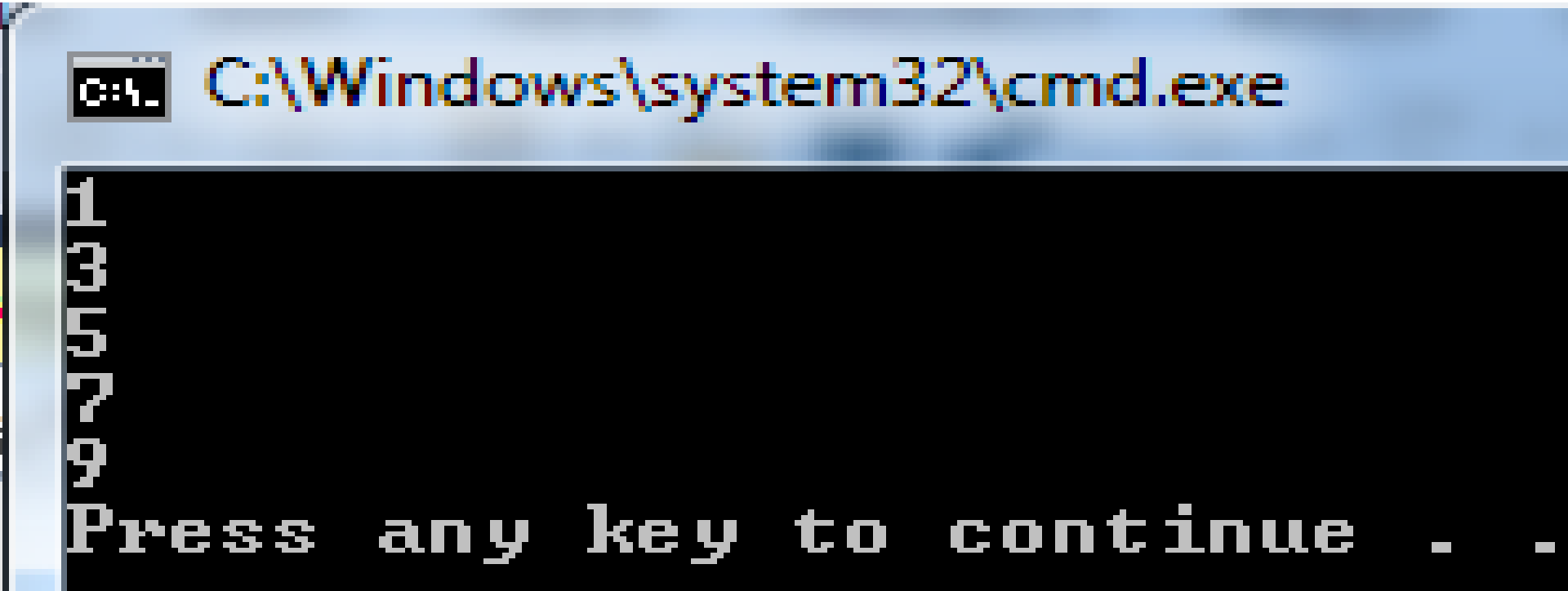
И в двата примера стъпката на цикъла е 1. Ако искаме стъпката да е различна от 1 трябва да променим оператора по следния начин:

for (int i=1; i<=10; i=i+k){<тяло на цикъла>}

Където новата стъпка е к.

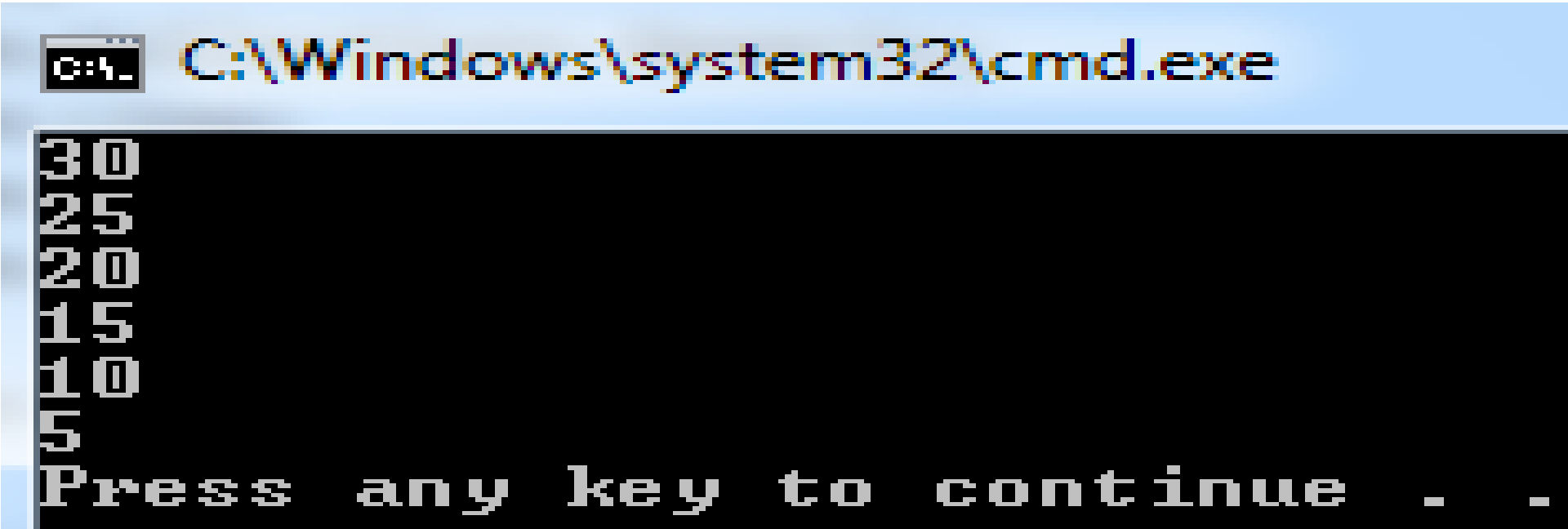
Пример3:

```
static void Main(string[] args)
{for(int i= 1; i <=10; i = i+2)
{ Console.WriteLine("{0}", i); }
}
```



Пример4:

```
static void Main(string[] args)
{for(int i = 30; i >=1; i = i-5)
{ Console.WriteLine("{0}", i); }
}
```

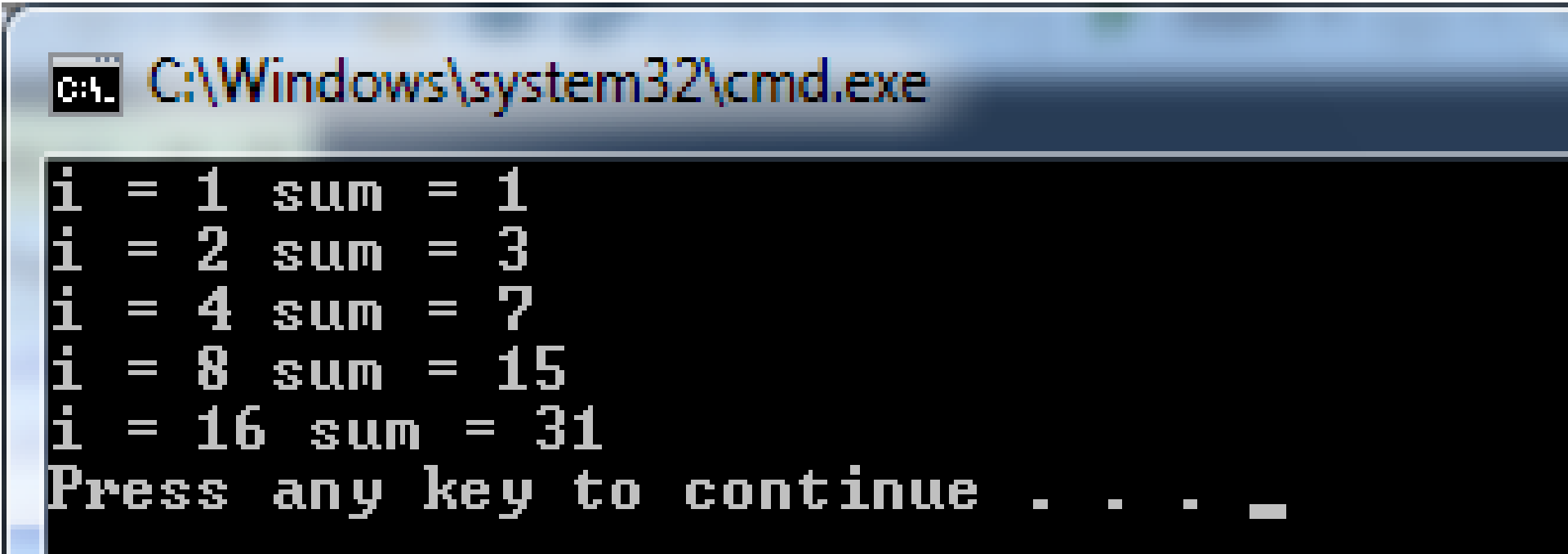


3. Други възможности

Следващият пример е цикъл, в който се инициализират и променят на всяка стъпка две променливи. Правилото е, че на мястото на всеки от изразите може да се напишат няколко израза, разделени със запетая

Пример5:

```
static void Main(string[] args)
{for(int i = 1, sum = 1; i <= 16; i = i*2, sum = sum + i)
{ Console.WriteLine("i = {0} sum = {1}", i, sum); }
}
```



Цикълът който намира сумата на числата от 1 до 10 може да се изпише и без тяло:

```
for(int sum=0, i=1; i<=10; sum=sum+i, i++)
```

Нито един от изразите в оператора не е задължителен – инициализацията може да се направи вън от оператора за цикъл, а проверката на условието и обновяването на променливи в тялото на цикъла. В такъв случай изпълнението на цикъла трябва да се прекрати с оператора break;

```
int i=1, sum=0;
for(; ;){sum=sum+i; i++;
if(i>10) break;}
```

4. Работа с компютър.

Задача: Да се напише функция, която съставя таблица за преминаване от температурната скала на Целзий към температурната скала на Фаренхайт.

Формула за преизчисляване на температурата е f = 9/5.c + 32,където c е температурата по скалата на Целзий, а f съответната и температура по Фаренхайт.

```
static void Main(string[] args)
{
    double c, f;
    Console.WriteLine("C     F");
    for(c=36.0; c<38.0; c =c+0.1)
    { f = (9.0 / 5.0) * c + 32.0;
    Console.Write("{0,4:##.0}", c);
    Console.Write("{0,5:##.0}\n", f);
    }
}
```