## Числата от 1 до N през 3

Напишете програма, която въвежда число n и отпечатва **числата от 1 до** n **през 3** (със стъпка 3).

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

for (int i = 1; i <= n; i=i+3)

{ Console.WriteLine(i); }

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 10 | 1  4  7  10 | 7 | 1  4  7 | 15 | 1  4  7  10  13 |

## Числата от N до 1 в обратен ред

Напишете програма, която въвежда цяло положително число n и печата **числата от** n **до** **1** **в обратен ред**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2 | 2  1 | 3 | 3  2  1 | 5 | 5  4  3  2  1 |

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

for (int i = n; i > 0; i--)

{

Console.WriteLine(i);

}

## Числа от 20 до 2n

Напишете програма, която чете от конзолата цяло число n и **печата числата от 1 до 2n**.

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

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

{

double PowersOfTwo = Math.Pow(2, i);

Console.WriteLine(PowersOfTwo);

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  |
| 3 | 1  2  4  8 | 4 | 1  2  4  8  16 |

## Четни степени на 2

Да се напише програма, която въвежда n и **печата четните степени на 2** **≤** **2n**: **20**, **22**, **24**, **28**, …, **2n**. 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 |

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

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

{

if (i % 2 == 0)

{ Console.WriteLine(Math.Pow(2, i)); }

}

## Редица числа 2k+1

Напишете програма, която въвежда число nи отпечатва **всички числа ≤ n от редицата**: 1, 3, 7, 15, 31, …. Всяко следващо число се изчислява като **предишното число \* 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 |

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

int k = 1;

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

{

if (k <= n)

{

Console.WriteLine(k);

k = 2 \* k + 1;

}

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

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

int k = 1;

while (k <= n)

{

Console.WriteLine(k);

k = 2 \* k + 1;

}

## Число в диапазона [1…100]

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

Console.Write("Enter a number in the range [1...100]: ");

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

while (n < 1 || n > 100)

{

Console.WriteLine("Invalid number!");

Console.Write("Enter a number in the range [1...100]: ");

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

}

Console.WriteLine($"The number is: {n}");

}

|  |
| --- |
| **вход / изход** |
| Еnter a number in the range [1...100]: **0**  Invalid number!  Еnter a number in the range [1...100]: **77**  The number is: 77 |

## Най-голям общ делител (НОД)

Напишете програма, която въвежда две цели положителни числа a и **b** и изчислява и отпечатва **най-големият им общ делител (НОД)**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 24  16 | 8 | 67  18 | 1 | 15  9 | 3 | 100  88 | 4 | 10  10 | 10 |

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

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

while (A % B != 0)

{

int temp = A % B;

A = B;

B = temp;

}

Console.WriteLine(B);

## Факториел

Напишете програма, която въвежда цяло число n (1 ≤ n ≤ 12) и **изчислява и отпечатва** **n!** = 1 \* 2 \* … \* **n** (**n** **факториел**). Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 5 | 120 | 6 | 720 | 10 | 3628800 | 1 | 1 | 2 | 2 |

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

int factoriel = 1;

int count = 1;

while (count <=n )

{

factoriel = factoriel \* count;

count++;

}

Console.WriteLine(factoriel);

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

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

int factoriel = 1; // Важно е да се инициализира от 1!

for (int i = 1; i <= n; i++)

{

factoriel = factoriel \* i;

}

Console.WriteLine(factoriel);

## Сумиране на цифрите на число

Напишете програма, която въвежда цяло число num и отпечатва **сумата от цифрите му**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **коментар** |  | **вход** | **изход** | **коментар** |  | **вход** | **изход** |  | **вход** | **изход** |
| 5634 | 18 | 6+6+3+4 = 18 | 19 | 10 | 1+9 = 10 | 5 | 5 | 17151 | 15 |

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

int sum = 0;

while (n > 0)

{

int k = n % 10;

sum = sum + k;

n = n / 10;

}

Console.WriteLine(sum);

-------------------------- с превръщане в string

string input = Console.ReadLine();

int lenght = input.Length;

int sum = 0;

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

{

sum += int.Parse(input[i].ToString());

}

Console.WriteLine(sum);

## Проверка за просто число

Напишете програма, която въвежда цяло число n 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** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2 | Prime | 3 | Prime | 4 | Not Prime | 5 | Prime | 7 | Prime |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 1 | Not Prime | 0 | Not Prime | -1 | Not Prime | 149 | Prime |

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

n = Math.Abs(n);

int count = 1;

while (true)

{

if (n > count && n % count == 0)

{

Console.WriteLine("Not prime");

break;

}

else if (n == 1 || n == 0)

{

Console.WriteLine("Not prime");

break;

}

else if (n == 2 || n > count)

{

Console.WriteLine("prime");

break;

}

else

{

break;

}

count++;

}

## Въвеждане на четно число (с обработка на грешен вход)

Напишете програма, която **въвежда четно число**. 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:

int n = 0;

while (true)

{

try

{

Console.Write($"Enter even number: ");

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

if (n % 2 == 0)

{

break;

}

Console.WriteLine($"The number is not even.");

}

catch (Exception)

{

Console.WriteLine("Invalid number!");

}

}

Console.WriteLine($"Even number entered: {n}");

|  |
| --- |
| **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 |

## Числа на Фибоначи

Напишете програма, която въвежда цяло число **n** 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 |

12345

Числа Фибоначи:

0 == 1

1 == 1

2= 1+1 =2

3 = 1+2 =3

4 = 2+3 = 5

5 = 3+5 = 8

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

int sum = 0;

int F0 = 1;

int F1 = 1;

if (n == 0 || n == 1)

{

sum = 1;

Console.WriteLine(sum);

}

for (int i = 1; i < n; i++)

{

sum = F0 + F1;

F0 = F1;

F1 = sum;

}

Console.WriteLine(F1);

## Пирамида от числа

Напишете програма, която въвежда цяло число n и отпечатва **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 |

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

int num = 1;

for (int row = 1; row <= n; row++)

{

for (int col = 1; col <= row; col++)

{

Console.Write(num);

Console.Write(" ");

num++;

if (num >n)

{

break;

}

}

Console.WriteLine();

if (num>n)

{

break;

}

}

## Таблица с числа

Напишете програма, която въвежда цяло число n и отпечатва **таблица (матрица) от числа** като в примерите:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2 | 1 2  2 1 | 3 | 1 2 3  2 3 2  3 2 1 | 4 | 1 2 3 4  2 3 4 3  3 4 3 2  4 3 2 1 | 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 |

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

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

{

for (int j = 0; j < n; j++)

{

int num = i + j + 1;

if (num > n)

{

num = 2 \* n - num;

}

Console.Write(num + " ");

}

Console.WriteLine();

}

## Генератор за тъпи пароли

*Шеста задача от изпит на 6 март 2016. Тествайте решението си* [***тук***](https://judge.softuni.bg/Contests/Practice/Index/169#5)*.* Да се напише програма, която въвежда две цели числа *n* и *l* и генерира по азбучен ред всички възможни **“тъпи” пароли**, които се състоят от следните **5 символа**:

* Символ 1: цифра от 1 до *n*.
* Символ 2: цифра от 1 до *n*.
* Символ 3: малка буква измежду първите *l* букви на латинската азбука.
* Символ 4: малка буква измежду първите *l* букви на латинската азбука.
* Символ 5: цифра от 1 до n, по-голяма от първите 2 цифри.

### Вход Входът се чете от конзолата и се състои от две цели числа *n* и *l* в интервала [1…9], по едно на ред.

### Изход На конзолата трябва да се отпечатат всички “тъпи” пароли по азбучен ред, разделени с интервал.

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

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

// паролата се състои от 5 k-елемента (k1,k2,k3,k4,k5)

for (int k1 = 1; k1 <= n; k1++)

{

for (int k2 = 1; k2 <= n; k2++)

{

for (var k3 = 'a'; k3 < 'a' + L; k3++)

{

for (var k4 = 'a'; k4 < 'a' + L; k4++)

{

for (int k5 = Math.Max(k1, k2) + 1; k5 <= n; k5++)

{

Console.Write($"{k1}{k2}{k3}{k4}{k5} "); }

|  |  |
| --- | --- |
| **вход** | **изход** |
| 2  4 | 11aa2 11ab2 11ac2 11ad2 11ba2 11bb2 11bc2 11bd2 11ca2 11cb2 11cc2 11cd2 11da2 11db2 11dc2 11dd2 |
| 3  1 | 11aa2 11aa3 12aa3 21aa3 22aa3 |

## Магически числа

*Шеста задача от междинния изпит на 26 март 2016. Тествайте решението си* [***тук***](https://judge.softuni.bg/Contests/Practice/Index/179#5)*.* Да се напише програма, която въвежда едно цяло **„магическо“** число и изкарва **всички** възможни **6-цифрени числа**, за които **произведението на неговите цифри** е **равно** на **„магическото“ число. Пример:** „Магическо число“ -> 2

* 111112 -> 1 \* 1 \* 1 \* 1 \* 1 \* 2 = 2
* 111121 -> 1 \* 1 \* 1 \* 1 \* 2 \* 1 = 2
* 111211 -> 1 \* 1 \* 1 \* 2 \* 1 \* 1 = 2
* 112111 -> 1 \* 1 \* 2 \* 1 \* 1 \* 1 = 2
* 121111 -> 1 \* 2 \* 1 \* 1 \* 1 \* 1 = 2
* 211111 -> 2 \* 1 \* 1 \* 1 \* 1 \* 1 = 2

### Вход Входът се чете от конзолата и се състои от едно цяло число в интервала [1…600000].

### Изход На конзолата трябва да се отпечатат всички “магически” числа, разделени с интервал.

|  |  |
| --- | --- |
| **вход** | **изход** |
| 2 | 111112 111121 111211 112111 121111 211111 |
| 8 | 111118 111124 111142 111181 111214 111222 111241 111412 111421 111811 112114 112122 112141 112212 112221 112411 114112 114121 114211 118111 121114 121122 121141 121212 121221 121411 122112 122121 122211 124111 141112 141121 141211 142111 181111 211114 211122 211141 211212 211221 211411 212112 212121 212211 214111 221112 221121 221211 222111 241111 411112 411121 411211 412111 421111 811111 |
| 531441 | 999999 |

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

for (int k1 = 1; k1 <= 9; k1++)

{

for (int k2 = 1; k2 <= 9; k2++)

{

for (int k3 = 1; k3 <= 9; k3++)

{

for (int k4 = 1; k4 <= 9; k4++)

{

for (int k5 = 1; k5 < 9; k5++)

{

for (int k6 = 1; k6 < 9; k6++)

{

if (k1 \* k2 \* k3 \* k4 \* k5 \* k6 == n)

{

Console.Write($"{k1}{k2}{k3}{k4}{k5}{k6} ");

}

## Специални числа

Да се напише програма, която **въвежда едно цяло число** N и генерира всички възможни **“специални”** **числа** от **1111** до **9999**. За да бъде **“специалнo”** едно число, то трябва да отговаря на **следното условие**:

* **N да се дели на всяка една от неговите цифри без остатък.**

**Пример:** при **N = 16**, **2418** е специално число:

* **16 / 2** = 8 **без остатък**
* **16 / 4** = 4 **без остатък**
* **16 / 1** = 16 **без остатък**
* **16 / 8** = 2 **без остатък**

### Вход

Входът се чете от конзолата и се състои от **едно** **цяло число** в интервала [**1**…**600000**]

### Изход

На конзолата трябва да се отпечатат **всички “специални” числа**, разделени с **интервал**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **коментари** |
| 3 | 1111 1113 1131 1133 1311 1313 1331 1333 3111 3113 3131 3133 3311 3313 3331 3333 | 3 / 1 = 3 без остатък  3 / 3 = 1 без остатък  3 / 3 = 1 без остатък  3 / 3 = 1 без остатък |
| 11 | 1111 | |

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

for (int i = 1; i <= 9; i++)

{

for (int j = 1; j <= 9; j++)

{

for (int m = 1; m <= 9; m++)

{

for (int k = 1; k <= 9; k++)

{

if (n % i == 0 && n % j == 0 && n % m == 0 && n % k == 0)

{

Console.Write("{0}{1}{2}{3} ", i, j, m, k);

}