

```

1      {
      File:      factor.dpr

      Compiler. Borland Delphi 7.0

      Задача. Требуется вычислить факториал целого числа N. Факториал
      обозначают как N! и вычисляют по формуле:
          N! = 1 * 2 * 3 * ... * (N-1) * N,
      причем
          0! = 1.
      Так же допустимо рекуррентное соотношение:
          N! = (N-1)! * N.

      В единственной строке входного файла input.txt записано
      одно целое неотрицательное число N (N < 1000).

      В выходной файл output.txt нужно вывести одно целое
      число - значение N!.

      Решение. Калмыков Вадим (ProCrypt),
      г. Сургут, ЦНИТ "Северная Звезда",
      14:49, 23.01.08

      Source.  http://acm.dvpion.ru

      Public.  www.Best-Listing.ru

      Editor.  MSP, 10.12.09, 17:53, LIST 6.0, (c) MSP-Soft
      }

```

```

22  Program factorial;

23  Uses
24  SysUtils; // модуль, содержащий функцию IntToStr

25  Const
26  Size = 322; // Максимально возможное количество элементов

27  Var
28  i, j : Integer; // счетчики циклов

29  // для отрицательных чисел факториал неопределен, значит
30  // используем тип Word
31  N : Word;
32
33  // строка для хранения длинного числа перед выводом
34  ch : String [8];

35  // массив для хранения цифр
36  F : array [1..Size] of Int64;

```

37
38
39

Begin

{

Алгоритм

Объявим массив из 322 элементов типа Int64. В каждом элементе будем хранить по 8 цифр числа начиная с конца числа, что позволяет сделать массив небольшим и уменьшить количество итераций цикла при вычислении. Хранить меньше цифр недопустимо по скорости выполнения программы. Если число, содержащееся в каком-либо элементе массива превышает 99999999, то происходит перенос целой части в следующий элемент при делении на 100000000, а в текущем сохраняется остаток от деления на это же число.

Последовательно умножаем каждый элемент на число из диапазона от 1 до N и проверяем перенос разряда. При переносе может произойти переполнение следующего разряда, значит, при переносе следует проверять остальные разряды.

В текущем элементе может храниться число длиной менее 8 символов, а следующий разряд может быть отличным от нуля, значит в текущем элементе на место старших разрядов устанавливаем нули.

}

59
60

```
Assign (Input, 'input.txt');  
Assign (Output, 'output.txt');
```

61
62
63

```
// в последний элемент устанавливаем значение 1, чтобы  
// произведение было отлично от нуля.  
F [Size] := 1;
```

64
65

```
// остальные элементы массива инициализируем нулями.
```

66
67

```
for i := 1 to Size - 1 do  
  F [i] := 0;
```

68
69

```
Reset (Input);  
Read (N);
```

70

```
Close (Input);
```

71
72
73

```
// пока считанное число остается больше 1, производим  
// в обратном порядке умножение. Так как при умножении на 1  
// изменений не происходит, то может невыполнять при  
// N = 1 лишние итерации.
```

74
75

```
while (N > 1)
```

76
77

```
do
```

```
begin
```

78
79

```
  // Начинаем со второго, так как может произойти
```

```
  // переполнение, поэтому первый элемент является резервным,
```

```

80         // что позволяет избавиться от ошибки
81         // Runtime Error,
82         // но все равно желательно установить директиву
83         // {$R-}
84         // чтобы избавиться от ошибок при некорректных данных.
85         for i := 2 to Size do
86             begin
87                 F [i] := F [i] * N; // умножаем текущий элемент
88
89                 if (F [i] > 99999999)
90                     then
91                         {
92                             Произшел перенос, значит нужно проверить
93                             остальные элементы
94                         }
95                 for j := Size downto 2 do
96                     begin
97                         if (F [j] > 99999999)
98                             then
99                                 begin
100                                     { Производим перенос разрядов }
101                                     Inc (F [j - 1], F [j] div 100000000);
102                                     F [j] := F [j] mod 100000000;
103                                 end;
104                             end;
105
106                 Dec (N); // уменьшаем N
107             end;
108
109         i := 2; // ищем первый элемент в массиве отличный от нуля.
110         while (F [i] = 0)
111             do
112                 Inc (i);
113
114         ReWrite (Output);
115
116         // печатаем его полностью без ведущих нулей,
117         // так как он первый в числе.
118         Write (F [i]);
119
120         Inc (i);
121
122         for N := i to Size do
123             begin
124                 // переводим для удобства число в строку
125                 ch := IntToStr (F [N]);
126
127                 {
128                     Если длина строки менее 8 символов, то выводим ведущие
129                     нули для каждого элемента

```

```
    }
126     for j := 1 to 8 - Length (ch) do
127         Write (0);

128         // после чего печатаем отличное от нуля значение элемента
129         Write (F [N]);
130     end;

131     // вычисления произведены, закрываем файл для сохранения данных.
132
133     Close (Output);
134 End.
```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-10-task-649.html>

Sergey Mitrofanov, 11.09.14, 12:17

E-mail: infostar@mail.ru

© <http://www.Best-Listing.ru/>, 2006–2014