

*«Доктор, дайте мне таблеток
от жадности. Да побольше,
побольше!»*

Анекдот.

Речь пойдет сегодня о желании умножать целые числа, в которых количество цифр ограничено только... да ничем не ограничено, то есть о «Длинной арифметике» – так называют раздел олимпиадных задач по программированию, в которых требуется осуществлять арифметические действия с очень большими целыми числами. Многие сталкивались с ситуацией, когда обычный калькулятор не в состоянии точно вычислить, скажем, $20!$ (факториал), и причина банальна – табло калькулятора не в состоянии вместить все цифры ответа. Та же проблема и с калькулятором в компьютере. Даже тип данных `int64`, реализованный в языке Delphi, не может выполнять целочисленные арифметические действия с числами, которые превышают 2^{63} . Но иногда так хочется вычислить, скажем, произведение очень больших целых чисел, и хочется, чтобы ответ был точным, не приближенным, как скажем, это число: 1,414213562373095E45.

Таким образом, умение написать программу для выполнения арифметических действий с очень большими целыми числами применимо не столько на олимпиадах по программированию, сколько пригодится для пытливого математика.

Олимпиадные задачи на длинную арифметику, на наш взгляд, заметно отличаются от остальных задач тем, что ученику можно приготовить несколько домашних заготовок – готовых процедур, которые можно применить на олимпиаде, столкнувшись с задачей длинной арифметики.

Одна из таких задач длинной арифметики предлагается на сайте «Школа программиста»¹.

Постановка задачи.

Требуется вычислить факториал целого числа N . Факториал обозначают как $N!$ и вычисляют по формуле:

$$N! = 1 * 2 * 3 * \dots * (N-1) * N, \text{ причем } 0! = 1.$$

Так же допустимо рекуррентное соотношение: $N! = (N-1)! * N$

Входные данные

В единственной строке входного файла INPUT.TXT записано одно целое неотрицательное число N ($N < 1000$).

Выходные данные

В выходной файл OUTPUT.TXT нужно вывести одно целое число — значение $N!$.

¹ (Факториал, 2007)

Примеры

Таблица 1

| № | INPUT.TXT | OUTPUT.TXT |
|---|-----------|------------|
| 1 | 1 | 1 |
| 2 | 3 | 6 |
| 3 | 5 | 120 |

Идея длинной арифметики проста, как перчатка: длинное целое число будем хранить в памяти компьютера как массив целых чисел – цифр. Операции сложения, вычитания, умножения будем осуществлять в точности так, как делает это человек со своим интеллектом.

Приступаем теперь к реализации описанной идеи.

Таблица 2

| | |
|---|---|
| <pre>uses crt; const n=255; Type TLongInt=array[1..n] of integer; var a,x,b:TlongInt; h,hx:string;</pre> | <p>Константа n ограничивает длину целого числа. Создаем тип – массив для хранения длинного целого числа.</p> <p>Объявляем глобальные переменные: a, x, b – длинные целые числа. h, hx – вспомогательные строки.</p> |
| <pre>Procedure StrToLongInt(h:string; var a:TlongInt); var i,k,code:integer; begin k:=n-length(h); for i:=1 to k do h:='0'+h; for i:=1 to n do val(h[i],a[i],code); end;</pre> | <p>Создаем процедуру, которая на вход в качестве формальных параметров получает строку h, в которой записаны цифры, и возвращает переменную a – длинное целое число.</p> |
| <pre>Procedure LongIntToStr(a:TlongInt; var h:string); var i,j:integer; hh:string; begin i:=1;h:=''; while (a[i]=0)and(i<n) do inc(i); if (i=n)and(a[i]=0) then h:='0' else begin for j:=i to n do begin str(a[j],hh); h:=h+hh end end;</pre> | <p>Напишем обратную процедуру: на вход в качестве формальных параметров она получает длинное целое число, и возвращает строку – соответствующую длинному целому числу.</p> |

3

```
Procedure ShortProduct(a:TlongInt;  
c:integer; var x:TlongInt);
```

Напишем процедуру, которая умеет находить произведение длинного целого числа и обычной цифры. Эта процедура будет использоваться нами как вспомогательная при выполнении умножения двух длинных чисел. В качестве формальных параметров процедура получает длинное целое число и обычное целое число. На выходе процедура запишет результат произведения в переменную x.

```
var i,um,d:integer;  
begin  
  um:=0;  
  for i:=n downto 1 do  
    begin  
      d:=c*a[i]+um;  
      x[i]:=d mod 10;  
      um:=d div 10;  
    end;  
    if um>0 then  
      writeln('Переполнение порядка')  
    end;
```

Переменную um будем использовать, когда что-то потребуется записать в ум (3 пишем, 2 в уме).

```
Procedure LongSumma(a,b:TlongInt;  
var x:TlongInt);  
  var i,um,d:integer;
```

Напишем процедуру для выполнения сложения двух длинных целых чисел. В качестве формальных параметров процедура получает два длинных целых числа, и возвращает результат сложения. Переменную um будем использовать, когда что-то потребуется записать в ум (3 пишем, 2 в уме).

```
begin  
  um:=0;  
  for i:=n downto 1 do  
    begin  
      d:=a[i]+b[i]+um;  
      x[i]:=d mod 10;  
      um:=d div 10;  
    end;  
    if um>0 then  
      writeln('Переполнение порядка')  
    end;
```

Реализуем здесь обычное сложение в столбик. Если при сложении результат не помещается в массив, то происходит переполнение порядка, выдаем сообщение на экран, чтобы сообщить пользователю о такой неприятности.



| | |
|---|--|
| <pre> for i:=n downto k0+1 do begin ShortProduct(a,b[i],x); k:=n-i; </pre> | <p>Выполняем умножение первого множителя на очередную цифру второго множителя.</p> <p>Определяем количество k, насколько нужно сдвинуть все цифры результата этого произведения влево.</p> |
| <pre> Push(x,k); LongSumma(s,x,s); end; end; </pre> | <p>Сдвигаем очередное слагаемое на k цифр. Складываем все промежуточные произведения.</p> |
| <pre> procedure LongDec(var a:TLongInt); var i,j:integer; </pre> | <p>Пишем процедуру, которая в качестве формальных параметров получает длинное целое число, а и изменяет его, возвращает длинное целое, уменьшенное на 1.</p> <p>Объявляем локальные переменные i, j, используемые как индексные переменные.</p> |
| <pre> begin if a[n]>0 then dec(a[n]) </pre> | <p>Если последняя цифра не ноль, то уменьшаем e на 1.</p> |
| <pre> else begin j:=n; while(a[j]=0)and(j>1)do dec(j); </pre> | <p>В противном случае идем справа налево и ищем первую цифру – не ноль.</p> |
| <pre> if a[j]=0 then writeln('Переполнение порядка') </pre> | <p>Если все цифры нули, то вычесть 1 невозможно, возникло переполнение порядка, сообщаем об этой неприятности пользователю.</p> |
| <pre> else begin dec(a[j]); for i:=j+1 to n do a[i]:=9; end end; end; </pre> | <p>Найдена ненулевая цифра, уменьшаем ее на 1, а все нули справа заменяем на 9.</p> |
| <pre> function EqualTo1(a:TLongInt):Boolean; var i:integer; L:Boolean; begin L:=true; for i:=1 to n-1 do L:=L and(a[i]=0); L:=L and(a[n]=1); EqualTo1:=L; end; </pre> | <p>Пишем вспомогательную функцию, которая проверяет, равно ли длинное целое число единице. Будем использовать при вычислении факториала.</p> |



| | |
|---|--|
| <pre> Procedure LongFactorial (a:TLongInt;var s:TLongInt); var i:integer; begin for i:=1 to n-1 do s[i]:=0; s[n]:=1; repeat LongProduct (s, a, s); LongDec (a); until EqualTo1 (a); end; </pre> | <p>Пишем процедуру для вычисления длинного факториала. На вход в качестве формальных параметров процедура получает длинное целое число, и возвращает факториал данного числа. В переменную S записали единицу. Находим произведение длинных целых чисел s и a, затем длинное целое a уменьшаем на 1. И продолжаем процесс, пока число a не станет равно 1.</p> |
| <pre> begin h:='146'; Writeln(h); StrToLongInt (h, a); LongFactorial (a, x); longIntToStr (x, hx); writeln (hx); end. </pre> | <p>Тело программы. Организация ввода и вывода данных с помощью файлов не является предметом настоящей статьи. Ввод данных будем осуществлять простым присвоением. Строку преобразовали в длинное целое число, вычислили факториал длинного целого числа, преобразовали результат вычислений в строку, распечатали результат.</p> |

После успешной работы над программой можно предложить учащимся дописать процедуру для решения чуть более сложной задачи – найти сумму факториалов².

Список таблиц

Таблица 1 2
Таблица 2 2

² (Сумма факториалов, 2007)

Ключевые слова

| | | | |
|---------------------------------|------|---------------------------|---------------|
| Глобальные переменные | 2 | Незначащие нули | 4 |
| Длинная арифметика | 1 | Переполнение порядка..... | 3, 4, 5 |
| Локальные переменные | 4, 5 | Формальные параметры..... | 2, 3, 4, 5, 6 |
| Массив | 2, 3 | | |

Список литературы

1. *Сумма факториалов.* (8 1 2007 г.). Получено 19 11 2014 г., из Школа программиста: http://acmp.ru/index.asp?main=task&id_task=311
2. *Факториал.* (8 1 2007 г.). Получено 21 09 2013 г., из Школа программиста: http://acmp.ru/index.asp?main=task&id_task=18