

ДЛИННЫЕ ЧИСЛА И ДЛИННАЯ АРИФМЕТИКА

В данной работе рассмотрены основные принципы работы с длинными числами и реализованы алгоритмы, построенные на данных принципах.

ВВЕДЕНИЕ

Длинная арифметика — это совокупность математических операций (сложение, вычитание, умножение, деление и др.) над длинными числами, т.е. числами, разрядность которых может превышать длину машинного слова. Целочисленная длинная арифметика разделяется на несколько видов: классическая длинная арифметика (число хранится в виде массива его цифр), длинная арифметика в факторизованном виде (число хранится в виде массива степеней каждого входящего в него простого), длинная арифметика по системе простых модулей (выбирается некоторая система модулей и число хранится в виде вектора из остатков от его деления на каждый из этих модулей). В данной работе рассматривается только классическая классические длинные числа, т.к. для них возможны стандартные математические операции. Цифры в длинном числе записываются в массив в обратном порядке для упрощения алгоритмов. Цифры могут использоваться из той или иной системы счисления, обычно применяются десятичная система счисления и её степени, либо двоичная система счисления.

I. ПРИНЦИП РАБОТЫ И ПРИМЕР ИСПОЛЬЗОВАНИЯ

Алгоритмы сложения, вычитания, умножения и деления схожи с вычислениями “в столбик”, т.е. при превышении основания (отрицательном значении) разница между цифрой и основанием (нулем и цифрой) остается в текущем разряде, а число раз, в которое цифра превышает основание, прибавляется (отнимается) к следующему разряду. Пример сложения длинных чисел 32543412 и 1988967 представлен на рис. 1.

Бульбенков Степан Владимирович, студент 1 курса ФИТиУ, bulbenkovs@mail.ru.

Научный руководитель: Кривоносова Татьяна Михайловна, доцент кафедры вычислительных методов и программирования, krivonosova@bsuir.by.

$$\begin{array}{r}
 \begin{array}{cccc}
 [0] & [1] & [2] & [3] \\
 + 12 & 34 & 54 & 32 \\
 67 & 89 & 98 & 01 \\
 \hline
 79 & 123 & 152 & 33 \\
 & -100 & -100 & +1 \\
 & & +1 & \\
 \hline
 79 & 23 & 53 & 34
 \end{array}
 \end{array}$$

Рис. 1 – пример сложения длинных чисел (основание чисел равно 100)

На основе этих принципов были реализованы функции для работы с длинными числами. Используя реализованные функции, был проведен расчет первых 100 знаков после запятой числа e (Рис. 2).

```

int main()
{
    longint rez{ (long long*)calloc(11, sizeof(long long)), 11, 11 };
    longint tmp{ (long long*)calloc(11, sizeof(long long)), 11, 11 };
    tmp.value[10] = base / 2;
    for (int k = 3; k < 100; del(&tmp, k++)) rez = sum(rez, tmp);
    view(rez);
    free(tmp.value);
    free(rez.value);
}

```

```

Консоль отладки Microsoft Visual Studio
7182818284 5984523536 8287471352 6624977572 4789369995
9574966967 6277248766 3035354759 4571382178 5251664274

```

Рис. 2 – вычисление первых 100 знаков после запятой числа e

II. ВЫВОД

В результате работы реализованы функции для работы с длинными числами и пример использования данных функций. Длинные числа можно использовать для повышения точности расчетов или величины хранимых данных за счет повышения расхода памяти и снижения быстродействия, а также для операций над числами в произвольной системе счисления.

1. https://e-maxx.ru/algo/big_integer.