Министерство образования и науки Российской Федерации

*Санкт-Петербургский государственный электротехнический университет*

**Центр Новых Информационных Технологий**

**Кафедра МОЭВМ.**

**Отчёт по курсовой работе**

**по теме: «Длинная арифметика»**

Преподаватель: Самойленко В.П.

Выполнил: студент гр.5383

Допира В. Е.

Санкт-Петербург

2015

**Курсовая работа**

**Длинная арифметика**

**1.Формулировка задания**

Многозначное целое число *a* представляется в виде последовательности (массива) целых чисел *a*1, *a*2, …, *an* путем разбиения его цифр на группы по *t* цифр. Каждая группа цифр, обозначенная *ai*, является целым числом, не превосходящим по величине некоторого фиксированного целого числа *m*. Например, число 3 1415 9265 3509 7932 3846 при *t* = 4 и *m* = 32767 может быть представлено в виде массива {0003, 1415, 9265, 3509, 7932, 3846}.

Реализовать следующие операции над многозначными целыми числами: операции: +, ‑.

**2.Пример выполнения задания и его анализ**

**2.1 Пример решения задачи**

Первое число 14518416491578797968542148763864

Операция +

Второе число 766156816856647540020775623498

Ответ = 15284573308435445508562924387362

**2.2 Анализ**

Выбираем переменную t, которая показывает, на сколько цифр делятся числа *t=4.*

Первое число делим на меньшие по 4 разряда, начиная с конца 1451 8416 4915 7879 7968 5421 4876 3864. Аналогично второе число 76 6156 8168 5664 7540 0207 7562 3498.

Получившие числа не должны превышать m. У нас m ограничено типом unsigned int. m≤4 294 967 295.

Далее получившиеся числа складываются, и если их сумма переваливает за разряд, то к последующей сумме прибавляется перенос разряда.

3864+3498=**7362**

4876+7562=12438 (перенос разряда, 1 прибавляется к следующей сумме, а программа фиксирует **2438**)

5421+0207+1=**5629**

7968+7540=1**5508**

7879+5664+1=1**3544**

4915+8168+1=1**3084**

8416+6156+1=1**4573**

1451+76+1=**1528**

Ответ: 1528 4573 3084 3544 5508 5629 2438 7362

**3.Формальная постановка задачи**

**3.1 Исходные данные**

Заданы два длинных числа, представленные двумя строчками, выполняемая операция: сложение или вычитание и число разрядов t, на которое делятся числа.

**3.2. Ограничения на исходные данные**

В строке, содержащей исходные два числа, может быть не более 10000 элементов. Массив, который передается из строки, также может содержать не более 10000 символов. Количество разрядов t должно быть >0 и ≤9, так как числа, которые получаются при делении на разряды, не должны превышать m, ограниченное типом переменных, то есть m≤4 294 967 295. И значит, что t не превышает количество знаков m, то есть меньше 10.

Считаем, что пользователь может выбрать только операции сложения или вычитания.

**3.3. Результирующие данные**

Ответом является сумма или разность, в зависимости от выбранной операции.

**4. Спецификация программы**

**4.1.Входные данные**

**Ограничения**

Количество знаков в числе не должно превышать 10000: strDigital[10000]

Количество разрядов t, на которые будет поделено число, чтобы выполнить операции 0<t≤9. Программа осуществляет только операции сложения и вычитания.

**Место и форма представления входных данных**

Два длинных числа, расположенные в отдельных строках, находятся в файле «in\_data.txt». Если файла в папке с программой нет, то программа выдаст сообщение: «Входной файл не открыт!».

Пользователь вводит с клавиатуры значение количества для разбивки массива t, и также выбирает операцию суммы «+» или разности «-».

**4.2. Выходные данные**

Программа выводит на экран значение получившего числа после выполнения операций.

Также выше продублированы оба числа и выведены массивы, на которые делятся числа, и их количество.

**4.3. Описание сцены диалога**

Описание сцены приведено в табл. П.1.

Таблица П.1

|  |  |
| --- | --- |
| **Алгоритм ведения диалога** | **Сообщения пользователю и его реакция** |
| Вывод запроса | Введите количество для разбивки массива : |
| Ввод tCount | ?<Enter> |
| Вывод запроса | Введите операцию + сумма или - разность : |
| Ввод operation  (варианта операции) | ?<Enter> |
| Вывод результата | Число первое:  Число второе:  Массив первого числа  Массив второго числа  Количество разрядов в зависимости от Т  Массив результата, после выполнения операции  Результат:  Строка |

**5. Структуры данных и алгоритм работы программы**

**5.1. Структуры данных**

Два длинных числа считываются из файла «in\_data.txt» в строки typedef char strDigital[10000]. С помощью функции strlen подсчитывается количество элементов в строке: int len=strlen(strN). Элементы строк записываются в массив typedef unsigned int arrDigital [Max\_n]. Сначала строка делится на typedef short t разрядов с конца, формируя строку для одного элемента массива char s[t-1], затем функция atoi преобразовывает строку s в целочисленный тип и присваиваем значение элементу массива arr.

while (t\*(count+1)<=len) //Цикл по элементам строки

{

for (int i=0;i<t;i++) s[i]=strN[len-t\*(count+1)+i]; //Формирование строки для числа массива

arr[count]=atoi(s);// Заполнение элемента массива числом (преобразование строки в целое)

count++; //Увеличиваем количество элементов массива

}

Если длина вводимой строки для длинного числа не делится без остатка на t, то необходимо выполнить отдельно преобразования для оставшихся символов и дополнительно сформировать старший элемент массива.

if(len%t!=0)

{

int m=len%t;

memset (s,' ',t);

strcat (s,"\0");

for (int i=0;i<m;i++) s[t-m+i]=strN[i] ;

arr[count]= atoi(s);

}

Функция memset (s,' ',t) при последнем нецелом делении строки на t заполняет оставшееся место в памяти пробелами, а функция strcat (s,"\0") меняет их на нули.

Для просмотра программа дополнительно выводит на экран количество получившихся элементов массивов typedef unsigned short arrCount и сами элементы.

**5.2. Алгоритм работы программы**

**Шаг 1:** Проверка наличия файла с данными

**Шаг 2:** Ввод количества для разбивки массива

**Шаг 3:** Ввод операции

**Шаг 4:** Считывание две строки из файла «in\_data.txt»

**Шаг 5:** Вывод данных на экран

**Шаг 6:** Преобразование строк в целочисленный массив

**Шаг 7:** Вызов функции для выполнения операции

**Шаг 8:** Вывод результата

При выполнении шага 6 алгоритм преобразования строк в целочисленный массив следующий:

ЕСЛИ ( длина первого числа больше длины второго)

ТО число1в массив arrA, число2 в массив arrВ

ИНАЧЕ ЕСЛИ (длина первого числа меньше длины второго)

ТО {ЕСЛИ (операция – ) ТО

результат отрицательный flag=1

ВСЕ ЕСЛИ

число1в массив arrB, число2 в массив arrA

}

ИНАЧЕ // если количество цифр равно

{ число1в массив arrA, число2 в массив arrВ

ЕСЛИ (операция – ) ТО {

//проверка чтобы первое число было больше второго

ЦИКЛ по элементам массива

ЕСЛИ (arrA[i]<arrB[i]) ТО

результат отрицательный flag=1

число1в массив arrB, число2 в массив arrA

ВСЕ ЕСЛИ }

ВСЕ ЕСЛИ

}

ВСЕ ЕСЛИ

ВСЕ ЕСЛИ

**Для реализации этих действий используются функции:**

1. Функция записи элементов из строки в массив
2. Функция вывода массивов с разделенными на разряды числами
3. Функция вывода результата
4. Функция суммы
5. Функция разности

**Реализация функции СУММЫ осуществлена по следующему алгоритму:**

АЛГОРИТМ СУММЫ

1. Формирование делителя tD взависимости от Т (разрядность).
2. Перенос r = 0.
3. ЦИКЛ ПОКА счетчик <= counta количества элементов массива а
   1. Сумма a[i]+ b[i] + r.
   2. ЕСЛИ a[i]+ b[i] + r >= tD ТО происходит перенос в следующий разряд

Перенос r = 1. Из a[i] вычитаем tD.

ELSE случай, когда переноса не происходит r = 0.

ВСЕ ЕСЛИ

КОНЕЦ ЦИКЛА

1. ЕСЛИ r > 0 после сложения остался еще перенос, то нужно добавить еще одну цифру

ТО увеличиваем количество элементов массива counta++ и присваиваем ему перенос a[counta] = r.

ВСЕ ЕСЛИ

**Реализация функции РАЗНОСТЬ осуществлена по следующему алгоритму:**

АЛГОРИТМ РАЗНОСТЬ

1. Формирование делителя tD взависимости от Т (разрядность).
2. Заем отсутствует r = 0.
3. ЦИКЛ ПОКА счетчик <= counta количества элементов массива а
   1. ЕСЛИ a[i] < b[i] ТО разность с учетом заема

a[i] = (tD+a[i] )-b[i] - r;

Заем r = 1.

ELSE случай, когда заем не происходит

a[i] -= b[i] + r; заем r = 0.

ВСЕ ЕСЛИ

КОНЕЦ ЦИКЛА

**5.3. Блок-схема**

Ход основной программы представлен на блок-схеме 1.

**5.4. Описание функций**

**Функция записи элементов из строки в массив**

void readArr ( const int t, const strDigital strN, arrDigital &arr, arrCount &count );

Назначение: Преобразовывает строку в массив целых чисел, разделяя стороку по t символов.

Параметры: t- количество разрядов для элемента массива

strN - строка

arr – массив с числами

count – количество элементов в массиве

Внешний эффект: формирование массива и определение количества элементов в массиве.

**Функция вывода массивов с разделенными на разряды числами**

void writeArr (arrDigital &arr, arrCount &count);

Назначение: Вывод на экран массива, состоящих из разделенных на разряды чисел.

Параметры: arr - массив

count – количество элементов в массиве

Внешний эффект: вывод на экран.

**Функция вывода результата**

void writeStr ( arrDigital &arr, arrCount &count,const unsigned short f, const T &tT);

Назначение: Вывод на экран строки с результатом, преобразование массива в строку.

Параметры: arr – массив элементов

count – количество элементов в массиве

f – флаг, наличие отрицательного результата

tT – количество разрядов для элемента массива

Внешний эффект: вывод результата на экран.

**Функция суммы**

void addArr(arrDigital &a, arrCount &counta, arrDigital &b, arrCount &countb, T &tT);

Назначение: Выполнение операции суммы.

Параметры: tT - количество разрядов для элемента массива

а – массив первого числа

counta – количество элементов в массиве а

b – массив второго числа

countb – количество элементов в массиве b

Внешний эффект: массив с результатом при выполнении операции сложения.

**Функция разности**

void subtract(arrDigital &a, arrCount &counta, arrDigital &b, arrCount &countb, T &tT);

Назначение: Выполнение операции разности.

Параметры: tT - количество разрядов для элемента массива

а – массив первого числа

counta – количество элементов в массиве а

b – массив второго числа

countb – количество элементов в массиве b

Внешний эффект: массив с результатом при выполнении операции вычитания.

**6. Текст программы**

#include <iostream>

#include <stdlib.h>

#include <fstream>

#include <string.h>

#include <stdio.h>

using namespace std;

const unsigned short Max\_n = 10000; //Максимальное количество элементов массива

typedef unsigned int arrDigital [Max\_n] ; //Описание ТИПА массива

typedef unsigned short arrCount; //Описание типа для количества элементов массива

typedef char strDigital[10000]; //Тип строки из файла равная длинному числу

typedef short T; // Тип описание количество для разбивки массива

//typedef unsigned short F;//Тип для флага если результат положительный или отрицательный

//Функции записи элементов в массив и вывода на экран

void readArr ( const int t, const strDigital strN, arrDigital &arr,arrCount &count );

void writeArr ( arrDigital &arr,arrCount &count);

void writeStr ( arrDigital &arr, arrCount &count, const unsigned short f, const T &tT);

void addArr(arrDigital &a, arrCount &counta, arrDigital &b, arrCount &countb, T &tT);

void subtract(arrDigital &a, arrCount &counta, arrDigital &b, arrCount &countb, T &tT);

int main ( )

{

setlocale (0,""); // Подключение русского языка

T tCount; // описание количество для разбивки массива

strDigital strNumber1, strNumber2, strRezult; // Строки из файла=числам

arrDigital arrA,arrB={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}; // Массивы

arrCount countA,countB; // Количество элементов в массиве

char operation;

unsigned short flag=0; //Если результат положительный

ifstream infile ("in\_data.txt"); //данные берутся из файла

ofstream outfile ("out\_data.txt"); //данные записываются в файл

if (!infile) {cout << "Входной файл не открыт!" << endl; return 0;

}

cout << "Введите количество для разбивки массива : ";

cin >> tCount;

cout << "Введите операцию + сумма или - разность : ";

cin >> operation;

infile >> strNumber1;

infile >> strNumber2;

cout << "Число первое: " << strNumber1 << endl;

cout << "Число второе: " << strNumber2 << endl;

if(strlen(strNumber1)>strlen(strNumber2))//если длина первого числа больше длины 2

{

readArr(tCount,strNumber1,arrA,countA);

readArr(tCount,strNumber2,arrB,countB);

}

else if(strlen(strNumber1)<strlen(strNumber2))//если длина 1 числа меньше длины 2

{if (operation=='-') flag=1;

readArr(tCount,strNumber2,arrA,countA);

readArr(tCount,strNumber1,arrB,countB);

}

else // если количество цифр равно

{readArr(tCount,strNumber1,arrA,countA);

readArr(tCount,strNumber2,arrB,countB);

if (operation=='-')

// для разности проверяем чтобы первое число было больше втогого

// если не так меняем массивы чтобы aarA содержал большее число

{ for (int i=countA; i>=0; i--)

if (arrA[i]<arrB[i]) {

flag=1;

readArr(tCount,strNumber2,arrA,countA);

readArr(tCount,strNumber1,arrB,countB);

break; // выход из цикла

}

}

}

writeArr(arrA,countA);

writeArr(arrB,countB);

if (operation=='+')

{

addArr(arrA, countA, arrB, countB, tCount);

}

else

{

subtract(arrA, countA, arrB, countB, tCount);

}

writeArr(arrA,countA);

writeStr(arrA,countA,flag, tCount);

infile.close();

outfile.close();

return 0;

}

void readArr ( const int t, const strDigital strN, arrDigital &arr, arrCount &count) //Функция записи элементов в массив

{

count=0; //Количестро элементов в массиве

char s[t-1]; // Строка для одного элемента массива

int len=strlen(strN);

// cout <<"Длина строки = "<< len<< endl;

while (t\*(count+1)<=len) //Цикл по элементам строки

for(int i=0;i<t;i++)s[i]=strN[len-t\*(count+1)+i];//Формирование строки для числа массива

arr[count]= atoi(s);// Заполнение эемента массива числом (преобразование строки в целое)

// cout<<count<<" = "<<arr[count] << endl;

count++; //Увеличиваем количество элементов массива

// Формирование последнего элемента массива если количество элементов в строке не делится нацело на Т

if(len%t!=0)

{

int m=len%t;

memset (s,' ',t);

strcat (s,"\0");

for (int i=0;i<m;i++) s[t-m+i]=strN[i] ;

arr[count]= atoi(s);

// cout<<count<<" = "<<arr[count]<< endl;

}

else count--;

}

void writeArr ( arrDigital &arr, arrCount &count) //Функция вывода массива

{

int n = count;

cout << "WriteArr: " << n << endl;

for (int i=0; i<=n; i++) cout << i<<" = " << arr[i]<< endl;; // Вывод массива

}

void writeStr ( arrDigital &arr, arrCount &count,const unsigned short f, const T &tT) //Функция вывода массива строкой

{

cout << "Результат: " << endl;

if (f==1) cout << '-';

// если старшие элементы = 0, то находим первый не нулевой элемент массива

int k=count;

while (arr[k]==0)

{k--;

}

// Вывод массива строкой

for (int i=k; i>=0; i--) {

cout << arr[i];

// если количество цифр в arr[i] меньше Т (например t=3, arr[i]=45, нужно вывести 045)

cout.fill('0'); // символ заполнения

cout.width(tT); // количество для заполнения 0

}

}

void addArr(arrDigital &a, arrCount &counta, arrDigital &b, arrCount &countb, T &tT)

{

// Формирование делителя взависимости от Т

int tD=1;

for(int i = 0; i < tT; i++) tD\*=10;

cout<<tD<< endl;

int r = 0;

/\*r - обозначает сколько у нас "в уме" при сложение младших цифр в уме у нас 0\*/

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

{

a[i] += b[i] + r;

//сумма очередных цифр и переноса

if (a[i] >= tD)

{

//случай, когда происходит перенос в следующий разряд

r = 1;

a[i] -= tD;

}

else {

//случай, когда переноса не происходит

r = 0;

}

}

//если после сложения остался еще перенос, то нужно добавить еще одну цифру

if (r > 0) {

counta++;

a[counta] = r;

}

}

void subtract(arrDigital &a, arrCount &counta, arrDigital &b, arrCount &countb, T &tT)

{

// Формирование делителя взависимости от Т

int tD=1;

for(int i = 0; i < tT; i++) tD\*=10;

cout<<tD<< endl;

//функция вычитает из числа a число b

//r - обозначает был ли заем единицы из текущего разряда

int r = 0;

//заем из разряда отсутствует

for(int i = 0; i <= counta; i++) {

if (a[i] < b[i])

{//разность очередных цифр с учетом заема

a[i] = (tD+a[i] )-b[i] - r;

r = 1;

}

else {

//случай, когда заем не происходит

a[i] -= b[i] + r;

r = 0;

}

}

}

**7. Тестирование программы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Входные данные** | | | **Ожидаемый результат** | **Полученный результат** |
| **Числа** | **t** | **Операция** |
| 987654321098765432109876543210  12345678901234567890 | 4 | + | 98765432111111  1111011111111100 | 9876543211111111  11011111111100 |
| 123456789012345678901234567890  987654321098765432109876543210 | 7 | - | -86419753208641  9753208641975320 | -86419753208641  9753208641975320 |
| 123456789012345678901234567890  123456789012365432109876543210 | 3 | - | -19753208641975320 | -19753208641975320 |
| 123456789012345678901234567890  987654321098765432109876543210 | 5 | + | 1111111110111111111  011111111100 | 1111111110111111111  011111111100 |
| Нет файла в папке |  |  | Входной файл не открыт! | Входной файл не открыт! |

**8. Выводы по результатам работы**

В ходе данной работы была разработана программа для вычисления сложения и вычитания двух длинных чисел на языке программирования С++. При выполнении курсовой работы использовались знания полученные во время лекций и практических занятий по типам данных, операторам языка С++, организации ввода и вывода данных, работа с файлами, подпрограммами и функциями, передача параметров. При разработке программы использовались массивы символов и целочисленных данных. Мною были изучены основные приемы работы со строками и функции преобразования.

Тема курсовой работы меня заинтересовала, можно было бы в дальнейшем разработать дополнительные функции (например: умножения, деления) для операций длинной арифметики.