**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Савченко Илья Владимирович

Группа: 80-207

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

Вариант 13

1. Постановка задачи  
    Создать класс **Long** для работы с целыми беззнаковыми числами из 64 бит. Число должно быть представлено двумя полями unsigned int. Должны быть реализованы арифметические операции, присутствующие в С++, и сравнения.
2. Описание программы

* Класс **Long** (myLong.h, myLong.cpp)
  + Приватные члены
    - uint32\_t **a**, **b —** два 32 битных числа без знака, две половинки 64 битного числа, где **a** — старший разряд
  + Публичные члены
    - Инициализаторы
    - Функция перевода **Long** в **uint64\_t** и обратно
    - Дружественные перегруженные операторы арифм. действий, сравнений и побитового сдвига и AND
    - Вспомогательные дружественные функции для работы деления и умножения
* Основная программа (**main()**) для демонстрации правильной работы класса
  + Вспомогательная функция **myTest** для вывода сравнения работы **Long** и **uint64\_t**

1. Набор тестов  
   Вводятся два числа (64 битный диапазон, stdin), программа выводит тесты арифм. действий, если результат не совпадает с результатом вычислений встроенных **unsigned long long**, то программа выведет сообщение о несовпадении результатов.

|  |  |
| --- | --- |
| **input** - test\_01.txt ;  7599449623315849900  117656294 | **output:** 7599449623315849900 + 117656294  = 7599449623433506194  7599449623315849900 - 117656294  = 7599449623198193606  7599449623315849900 \* 117656294  = 4473603405704978056  7599449623315849900 / 117656294  = 64590251528  7599449623315849900 % 117656294  = 3532668  A == B 0  A < B 0  A >= B 1 |
| **input** - test\_02.txt ;  5056501857544810777  364856249 | **output:**  5056501857544810777 + 364856249  = 5056501857909667026  5056501857544810777 - 364856249  = 5056501857179954528  5056501857544810777 \* 364856249  = 15996658269424158225  5056501857544810777 / 364856249  = 13858887908  5056501857544810777 % 364856249  = 120473685  A == B 0  A < B 0  A >= B 1 |
| **input** - test\_03.txt ;  5998485669454225695  2269173095170559442 | **output:**  5998485669454225695 + 2269173095170559442  = 8267658764624785137  5998485669454225695 - 2269173095170559442  = 3729312574283666253  5998485669454225695 \* 2269173095170559442  = 9580480057153683054  5998485669454225695 / 2269173095170559442  = 2  5998485669454225695 % 2269173095170559442  = 1460139479113106811  A == B 0  A < B 0  A >= B 1 |
| **input** - test\_04.txt ;  8027736654205395080  3270272109304338446 | **output:** 8027736654205395080 + 3270272109304338446  = 11298008763509733526  8027736654205395080 - 3270272109304338446  = 4757464544901056634  8027736654205395080 \* 3270272109304338446  = 11190118060867290992  8027736654205395080 / 3270272109304338446  = 2  8027736654205395080 % 3270272109304338446  = 1487192435596718188  A == B 0  A < B 0  A >= B 1 |

1. Результаты выполнения тестов  
   Арифметические операции над классом **Long** не отличаются результатами от класса **unsigned long long (uint64\_t)**.
2. Листинг программы

==> myLong.h <==

/\* Савченко Илья Владимирович

\* М8О-208Б-19

\*

\* вар 13

\* Создать класс Long для работы с целыми беззнаковыми числами из 64 бит.

\* Число должно быть представлено двумя полями unsigned int.

\* Должны быть реализованы арифметические операции,

\* присутствующие в С++, и сравнения.

\*/

#ifndef \_MY\_LONG\_

#define \_MY\_LONG\_

#include<cstdint>

class Long{

private:

uint32\_t a, b; // a - старший разряд

public:

Long() : a(0), b(0)

{}

Long(uint64\_t x) // для тестов

{setBy64(x);}

Long(uint32\_t \_a, uint32\_t \_b) : a(\_a), b(\_b)

{}

uint64\_t getUInt64() const{ // для тестов

uint64\_t toRet = a;

toRet = toRet << 32;

toRet += b;

return toRet;

}

void setBy64(uint64\_t x){

b = x;

a = x >> 32;

}

friend Long operator+(const Long &X, const Long &Y);

friend Long operator-(const Long &X, const Long &Y);

friend Long operator\*(const Long &X, const Long &Y);

friend Long operator/(const Long &X, const Long &Y);

friend Long operator%(const Long &X, const Long &Y);

friend Long helpMul(uint32\_t x, uint32\_t y);

friend void divideLong(const Long &X, const Long &Y, Long &Q, Long &R);

Long& operator++();

Long& operator--();

Long& operator+=(const Long &v);

Long& operator-=(const Long &v);

Long operator<<(unsigned int value);

Long operator>>(unsigned int value);

friend Long operator&(const Long &X, const Long &Y);

friend bool operator==(const Long &X, const Long &Y);

friend bool operator!=(const Long &X, const Long &Y);

friend bool operator< (const Long &X, const Long &Y);

friend bool operator> (const Long &X, const Long &Y);

friend bool operator<=(const Long &X, const Long &Y);

friend bool operator>=(const Long &X, const Long &Y);

};

==> myLong.cpp <==

#include"myLong.h"

Long Long::operator<<(unsigned int value){

Long toRet;

if(value == 0){

toRet.a = a;

toRet.b = b;

return toRet;

}

if(value < 32){

toRet.a = (a << value);

toRet.a += (b >> (32-value));

toRet.b = (b << value);

}

else if(value < 64)

toRet.a = (b << (value-32));

return toRet;

}

Long Long::operator>>(unsigned int value){

Long toRet;

if(value == 0){

toRet.a = a;

toRet.b = b;

return toRet;

}

if(value < 32){

toRet.b = (b >> value);

toRet.b += (a << (32-value));

toRet.a = (a >> value);

}

else if(value < 64)

toRet.b = (a >> (value-32));

return toRet;

}

Long operator&(const Long &X, const Long &Y){

Long toRet;

toRet.a = ( X.a & Y.a );

toRet.b = ( X.b & Y.b );

return toRet;

}

Long& Long::operator++(){

++b;

if(b == 0)

++a;

return \*this;

}

Long& Long::operator--(){

if(b == 0)

--a;

--b;

return \*this;

}

bool operator==(const Long &X, const Long &Y)

{return ((X.a == Y.a) && (X.b == Y.b));}

bool operator!=(const Long &X, const Long &Y)

{return !(X==Y);}

bool operator< (const Long &X, const Long &Y){

if(X.a == Y.a)

return (X.b < Y.b);

return (X.a < Y.a);

}

bool operator> (const Long &X, const Long &Y)

{return (Y<X);}

bool operator<=(const Long &X, const Long &Y)

{return !(Y<X);}

bool operator>=(const Long &X, const Long &Y)

{return !(X<Y);}

Long operator+(const Long &X, const Long &Y){

Long toRet;

toRet.a = X.a + Y.a;

if(X.b > UINT32\_MAX - Y.b)

++toRet.a;

toRet.b = X.b + Y.b;

return toRet;

}

Long operator-(const Long &X, const Long &Y){

Long toRet = X;

if(toRet.b < Y.b)

--toRet.a;

toRet.b -= Y.b;

toRet.a -= Y.a;

return toRet;

}

Long& Long::operator+=(const Long &v){

a += v.a;

if(b > UINT32\_MAX - v.b)

++a;

b += v.b;

return \*this;

}

Long& Long::operator-=(const Long &v){

a -= v.a;

if(b < v.b)

--a;

b -= v.b;

return \*this;

}

Long helpMul(uint32\_t x, uint32\_t y){

Long toRet, middle;

// X.b \* Y.b

toRet.b = (x % (1<<16)) \* (y % (1<<16));

toRet.a = (x >> 16) \* (y >> 16);

middle.b = (x >> 16) \* (y % (1<<16));

middle += (x % (1<<16)) \* (y >> 16);

toRet += (middle << 16);

return toRet;

}

void divideLong(const Long &X, const Long &Y, Long &Q, Long &R){

Q.a = 0;

Q.b = 0;

R.a = 0;

R.b = 0;

Long cX = X; // copy of X

for(int i=63; i>=0; --i){

R = R << 1;

R += ((cX>>i).b % 2);

if(R >= Y){

R -= Y;

Q += Long(0, 1) << i;

}

}

}

Long operator\*(const Long &X, const Long &Y){

Long toRet;

toRet = helpMul(X.b, Y.b);

toRet += helpMul(X.a, Y.b) << 32;

toRet += helpMul(X.b, Y.a) << 32;

return toRet;

}

Long operator/(const Long &X, const Long &Y){

Long Q, R;

divideLong(X, Y, Q, R);

return Q;

}

Long operator%(const Long &X, const Long &Y){

Long Q, R;

divideLong(X, Y, Q, R);

return R;

==> main.cpp <==

/\* Савченко Илья Владимирович

\* М8О-208Б-19

\* вар 13

\* Создать класс Long для работы с целыми беззнаковыми числами из 64 бит.

\* Число должно быть представлено двумя полями unsigned int.

\* Должны быть реализованы арифметические операции,

\* присутствующие в С++, и сравнения.

\*/

#include<iostream>

#include"myLong.h"

using namespace std;

void myTest(uint64\_t x, uint64\_t y, uint64\_t z, const Long &C, char op){

uint64\_t test = C.getUInt64();

if(z == test){

cout

<< x <<" "<< op <<" "<< y

<< "\n = "

<< z << endl;

}

else{

cout << "FAIL: "

<< x << " "<<op<<" "

<< y

<< "\n ! = "

<< z

<< "\n ! = "

<< test << endl;

}

}

int main(){

uint64\_t x, y, z, test;

Long A, B, C;

cin >> x >> y;

A.setBy64(x);

B.setBy64(y);

z = x+y;

C = A+B;

myTest(x, y, z, C, '+');

z = x-y;

C = A-B;

myTest(x, y, z, C, '-');

z = x\*y;

C = A\*B;

myTest(x, y, z, C, '\*');

z = x/y;

C = A/B;

myTest(x, y, z, C, '/');

z = x%y;

C = A%B;

myTest(x, y, z, C, '%');

cout

<< "A == B " << (A==B) <<endl

<< "A < B " << (A< B) <<endl

<< "A >= B " << (A>=B) <<endl;

Литература:

1. Уроки программирования на C++ | Ravesli [Электронный ресурс]  
   URL: <https://ravesli.com/uroki-cpp/>  
   (дата обращения: 23.11.2020)