

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХПІ»

Кафедра «Обчислювальна техніка та програмування»

Розрахункове завдання з дисципліни
«Алгоритми та структури даних»

Пояснювальна записка

Розробники

Виконав:

студент групи КІТ 119А

_____ / Момот Р.Є./

Перевірив:

_____ /Бречко В.А./

Харків 2020

ЗАТВЕРДЖЕНО

КІТ.119А.20205-14 80 01-1 -А3

Розрахункове завдання з дисципліни
«Алгоритми та структури даних»

Пояснювальна записка
КІТ.119А.20205-14 80 01-1 -А3

Листів 7

Харків 2020

1. Завдання.

Довгі числа. Реалізувати предсталення в пам'яті довгих беззнакових цілих чисел. Забезпечити виконання арифметичних операцій множення та ділення таких чисел. Порівняти час виконання зазначених операцій при роботі з даними типу `unsigned long` та запропонованою реалізацією довгих чисел.

2. Теоретичні відомості по темі завдання.

Unsigned long int - це тип даних розширеного розміру для зберігання додатних чисел розміром до 4 байт (32 біт). Змінні типу `unsigned long int` можуть містити числа від 0 до 4 294 967 295 ($2^{32} - 1$).

3. Текст програми

```
#ifndef _LONGNUMBER_H_
#define _LONGNUMBER_H_
#define MAX_BITS 512

#define __CRTDBG_MAP_ALLOC
#include <crtdbg.h>
#define DEBUG_NEW new(_NORMAL_BLOCK, __FILE__, __LINE__)
#define new DEBUG_NEW

#include <iostream>
#include <string>
#include <chrono>
#include <locale>

using namespace std;

class LongNumber {
    char m_bits[MAX_BITS];
    int m_nbits;

public:
    void clear();

    LongNumber operator*(LongNumber);
    LongNumber operator/(LongNumber);
    bool operator==(LongNumber);
    bool operator<(LongNumber);
    bool operator>(LongNumber);
    bool operator<=(LongNumber);

    LongNumber(unsigned long int k = 0);
```

```

        LongNumber(const LongNumber&);
        ~LongNumber();

};

#endif

int main()
{
    setlocale(LC_ALL, "Rus");

    cout << "Оператор *\n";
    auto beginClock = chrono::steady_clock::now();
    for (LongNumber i = 2; i < 1000000; i = i * 2);
    auto endClock = chrono::steady_clock::now();
    auto resultClock = chrono::duration_cast<chrono::nanoseconds>(endClock - beginClock);
    cout << "LongNumber: " << resultClock.count() << " нс\n";

    beginClock = chrono::steady_clock::now();
    for (unsigned long i = 2; i < 1000000; i = i * 2);
    endClock = chrono::steady_clock::now();
    resultClock = chrono::duration_cast<chrono::nanoseconds>(endClock - beginClock);
    cout << "unsigned long: " << resultClock.count() << " нс\n";

    cout << "\n";

    cout << "Оператор /\n";
    beginClock = chrono::steady_clock::now();
    for (LongNumber i = 1000000; i > 0; i = i / 2);
    endClock = chrono::steady_clock::now();
    resultClock = chrono::duration_cast<chrono::nanoseconds>(endClock - beginClock);
    cout << "LongNumber: " << resultClock.count() << " нс\n";

    beginClock = chrono::steady_clock::now();
    for (unsigned long i = 1000000; i > 0; i = i / 2);
    endClock = chrono::steady_clock::now();
    resultClock = chrono::duration_cast<chrono::nanoseconds>(endClock - beginClock);
    cout << "unsigned long: " << resultClock.count() << " нс\n";

    if (_CrtDumpMemoryLeaks())
        cout << "\nУтечка памяти обнаружена." << endl;
    else
        cout << "\nУтечка памяти отсутствует." << endl;

    return 0;
}

LongNumber::LongNumber(unsigned long int value)
{
    clear();

    if (value == 0)
        m_nbits = 1;
    else
        while (value > 0)
        {
            m_bits[m_nbits++] = value % 2;
            value /= 2;
        }
}

LongNumber::LongNumber(const LongNumber& other)
{
    memcpy(m_bits, other.m_bits, MAX_BITS);
    m_nbits = other.m_nbits;
}

LongNumber::~~LongNumber()

```

```

{
    clear();
}

void LongNumber::clear()
{
    memset(m_bits, 0, MAX_BITS);
    m_nbits = 0;
}

LongNumber LongNumber::operator*(LongNumber element)
{
    LongNumber result;

    result.m_nbits = m_nbits + element.m_nbits + 1;

    for (int i = 0; i < m_nbits; i++)
    {
        char carry = 0;
        for (int j = 0; j < element.m_nbits || carry; j++)
        {
            char cur = result.m_bits[i + j] + m_bits[i] * (j < element.m_nbits ?
element.m_bits[j] : 0) + carry;
            result.m_bits[i + j] = cur % 2;
            carry = cur / 2;
        }
    }

    while (result.m_nbits - 1 > 0 && result.m_bits[result.m_nbits - 1] == 0)
        result.m_nbits--;

    return result;
}

LongNumber LongNumber::operator/(LongNumber element)
{
    if (*this < element || *this == 0 || element == 0)
        return 0;

    if (*this == element)
        return 1;

    if (element == 1)
        return *this;

    int i = 0;
    while (element * i <= *this)
        i++;

    return LongNumber(--i);
}

bool LongNumber::operator==(LongNumber element)
{
    if (m_nbits != element.m_nbits)
        return false;

    for (int i = 0; i < m_nbits; i++)
        if (m_bits[i] != element.m_bits[i])
            return false;

    return true;
}

bool LongNumber::operator<(LongNumber element)
{
    if (*this == element)
        return false;

```

```

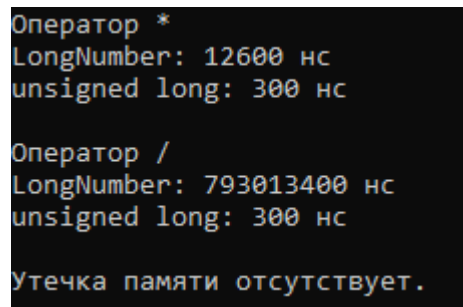
    if (m_nbits != element.m_nbits)
        return m_nbits < element.m_nbits;

    for (int i = m_nbits - 1; i >= 0; i--)
        if (m_bits[i] != element.m_bits[i])
            return m_bits[i] < element.m_bits[i];

    return false;
}
bool LongNumber::operator>(LongNumber element)
{
    return (*this == element || *this < element) ? false : true;
}
bool LongNumber::operator<=(LongNumber element)
{
    return (*this < element || *this == element) ? true : false;
}

```

4. Результати роботи програми (рис.1).



```

Оператор *
LongNumber: 12600 нс
unsigned long: 300 нс

Оператор /
LongNumber: 793013400 нс
unsigned long: 300 нс

Утечка памяти отсутствует.

```

Рис.1 Результат роботи програми

5. Пояснення алгоритму програми

Програма порівнює час роботи розробленого представлення довгих без знакових чисел та типу `unsigned long` на прикладі операцій множення та ділення. Спочатку виконується замір часу множення на 2 чисел від 2 до 1000000 у циклі для розробленого представлення чисел, а потім виконується такий же самий цикл для типу `unsigned long`. Після цього починається замір часу для операції ділення на 2 у циклі розробленої реалізації чисел та `unsigned long`.

6. Висновок

У ході виконання розрахунково-графічного завдання було набуто теоретичні знання та практичні навички роботи з представленням в пам'яті довгих без знакових цілих чисел та типу `unsigned long int`. Результати роботи програми показують, що операції над типом `unsigned long int` виконуються швидше, ніж над запропованою реалізацією, бо операції над `unsigned long int` створені та оптимізовані на найнижчих рівнях машин, на відміну від розробленого представлення.