

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

**Отчет по выполнению практического задания номер один**

**Тема: Поразрядные операции и их применение**

**Дисциплина: Структуры и алгоритмы обработки данных**

Выполнил студент Михайлюк Д. С.

Группы ИНБО-07-21

Москва 2022

**СОДЕРЖАНИЕ**

[1 ПОСТАНОВКА ЗАДАЧИ 3](#_Toc113653864)

[Задание 1.1 4](#_Toc113653865)

[Задание 1.2 6](#_Toc113653866)

[Задание 1.3 8](#_Toc113653867)

[Задание 1.4 10](#_Toc113653868)

[Задание 1.5 12](#_Toc113653869)

[Задание 2 14](#_Toc113653870)

# ПОСТАНОВКА ЗАДАЧИ

Получить навыки применения поразрядных операций в алгоритмах

В практической работе надо выполнить два задания.

Персональный вариант - №14:

Реализовать задачи:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | 1 Номер бита | 2 Номер бита | 3 множитель | 4 делитель | 5 Задание для выражения |
| 14 | 12-ый, 14-ый, 3-ий | Только с нечетными номерами | 512 | 512 | Установить n-ый бит в 1, используя маску (вар 2) |

# Задание 1.1

Условие задачи: определить переменную целого типа, присвоить ей значение, используя константу в шестнадцатеричной системе счисления.

Разработать оператор присваивания и его выражение, которое установит заданные в задании биты исходного значения переменной в значение 1, используя соответствующую маску и поразрядную операцию.

Персональный вариант 14: 12-ый, 14-ый, 3-ий биты.

**Разработка решения задачи пункта**

Выражение, реализующее операцию: i | 0x2802

Где i – изменяемое число, а 0x2802 – маска, необходимая для установки нужных битов в значение 1.

Двоичный вид маски: 10100000000010

**Тестовый пример с представлением двоичного кода необходимых операндов выражения**

Тестовое изменяемое значение: 1000016 = 6553610 = 100000000000000002

Значение маски: 280216 = 1024210 = 101000000000102

Выполнение операции поразрядное ИЛИ:

**Код функции**

int setbits(int i)

{

i = i | 0x2802;

return i;

}

#include <iostream>

using namespace std;

int main()

{

int n = 0x10000;

cout << n << endl;

cout << "After set 14th, 12th and 3rd bits in 1: " << setbits(n);

}

**Результаты тестирования**

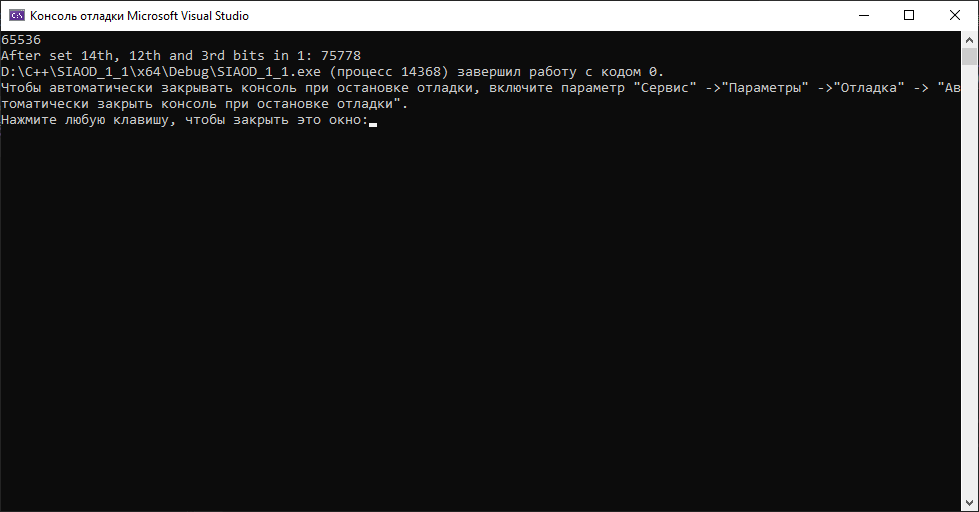


Рисунок 1 – результат выполнения 1го задания

# Задание 1.2

Условие задачи: определить переменную целого типа.

Разработать оператор присваивания и его выражение, которое обнуляет заданные в задании биты исходного значения переменной, используя соответствующую маску и поразрядную операцию. Значение в переменную вводится с клавиатуры.

Персональный вариант 14: только биты с нечетными номерами.

**Разработка решения задачи пункта**

Выражение, реализующее операцию: i & 0xAAAAAAAA

Где i – изменяемое число, а 0xAAAAAAAA – маска, необходимая для установки нужных битов в значение 0.

Двоичный вид маски: 10101010101010101010101010101010

**Тестовый пример с представлением двоичного кода необходимых операндов выражения**

Тестовое изменяемое значение: 1516 = 2110 = 10101 2

Значение маски: AAAAAAAA16 = 286331153010 = 101010101010101010101010101010102

Выполнение операции поразрядное И:

**Код функции**

#include <iostream>

using namespace std;

int killbits(int i)

{

i = i & 0xAAAAAAAA;

return i;

}

int main()

{

cout << "Pease, enter yout num: " << endl;

int n;

cin >> n;

cout << "After set odd bits in 0: " << killbits(n);

}

**Результаты тестирования**

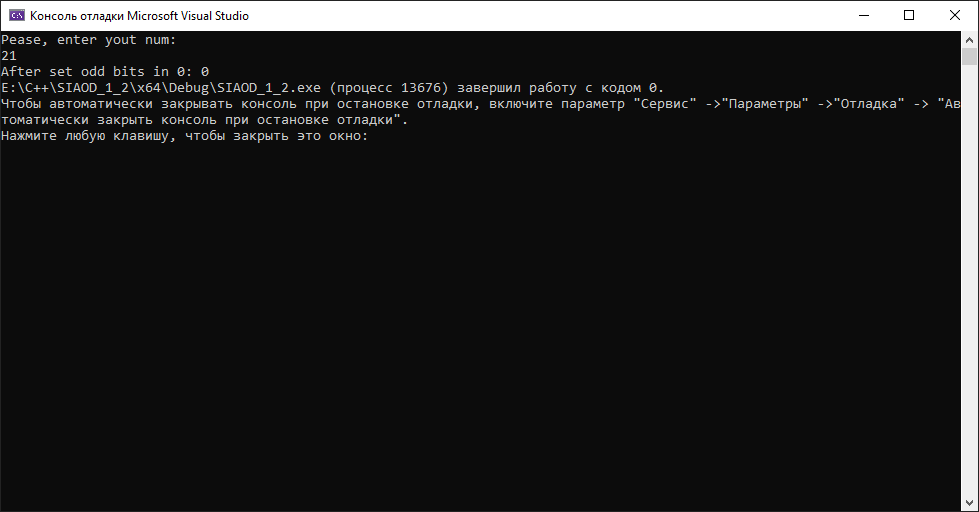


Рисунок 2 – результат выполнения 2го задания

# Задание 1.3

Условие задачи: определить переменную целого типа.

Разработать оператор присваивания и выражение, которое умножает значение переменной на число, указанное в третьем столбце варианта, используя соответствующую поразрядную операцию. Изменяемое число вводится с клавиатуры.

Персональный вариант 14: умножить на 512 (512 = 29 – сдвиг на 9 битов влево).

**Разработка решения задачи пункта**

Выражение, реализующее операцию: i << 9

Где i – изменяемое число, а 9 – маска, необходимая для сдвига нужных битов влево.

Двоичный вид маски: 1001

**Тестовый пример с представлением двоичного кода необходимых операндов выражения**

Тестовое изменяемое значение: 316 = 310 = 11 2

Значение маски: 916 = 910 = 10012

Выполнение операции поразрядный сдвиг вправо:

**Код функции**

#include <iostream>

using namespace std;

int bitMultiply(int i)

{

i = i << 9;

return i;

}

int main()

{

cout << "Pease, enter yout num: " << endl;

int n;

cin >> n;

cout << "After multiplying num by 512: " << bitMultiply(n);

}

**Результаты тестирования**

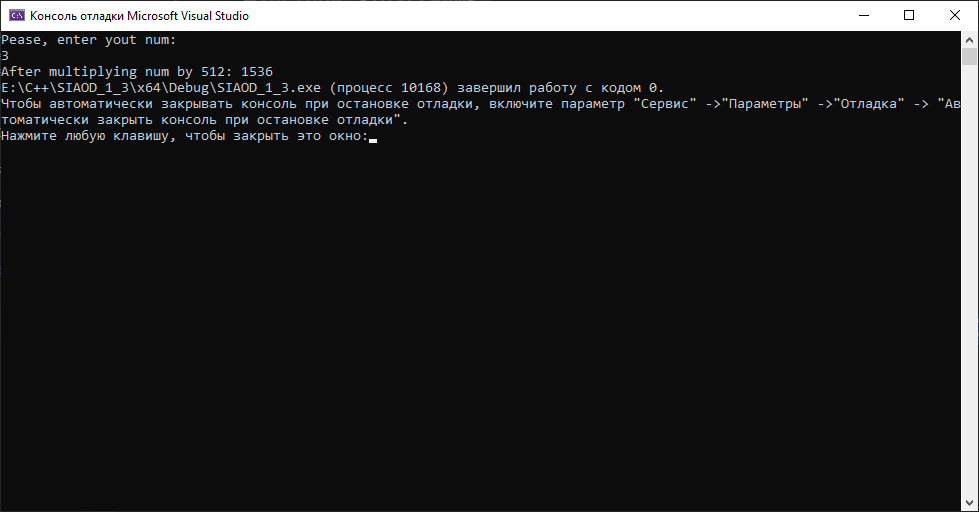


Рисунок 3 – результат выполнения 3го задания

# Задание 1.4

Условие задачи: определить переменную целого типа.

Разработать оператор присваивания и выражение, которое делит значение переменной на число, указанное в четвертом столбце варианта, используя соответствующую поразрядную операцию. Изменяемое число вводится с клавиатуры.

Персональный вариант 14: поделить на 512 (512 = 29 – сдвиг на 9 битов вправо).

**Разработка решения задачи пункта**

Выражение, реализующее операцию: i >> 9

Где i – изменяемое число, а 9 – маска, необходимая для сдвига нужных битов вправо.

Двоичный вид маски: 1001

**Тестовый пример с представлением двоичного кода необходимых операндов выражения**

Тестовое изменяемое значение: A0016 = 256010 = 101000000000 2

Значение маски: 916 = 910 = 10012

Выполнение операции поразрядный сдвиг вправо:

**Код функции**

#include <iostream>

using namespace std;

int bitDivide(int i)

{

i = i >> 9;

return i;

}

int main()

{

cout << "Pease, enter yout num: " << endl;

int n;

cin >> n;

cout << "After divided num by 512: " << bitDivide(n);

}

**Результаты тестирования**

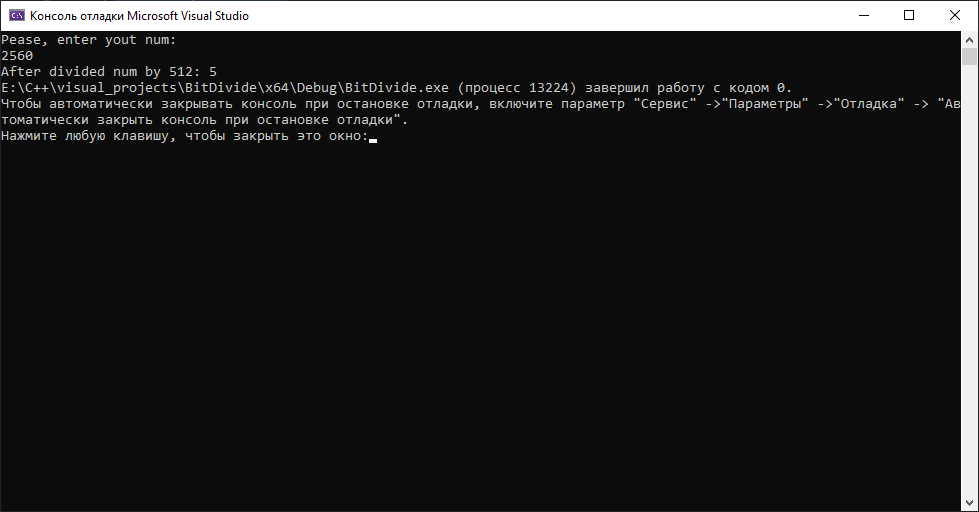


Рисунок 4 – результат выполнения 4го задания

# Задание 1.5

Условие задачи: определить переменную целого типа.

Разработать оператор присваивания и выражение, в котором используются только поразрядные операции. В выражении используется маска – переменная. Маска может быть инициализирована единицей в младшем разряде (вар 1) или единицей в старшем разряде (вар 2). Изменяемое число вводится с клавиатуры.

Персональный вариант 14: обнулить n-ый бит, используя маску.

**Разработка решения задачи пункта**

Выражение, реализующее операцию: вариант 2: дана маска с единицей в старшем разряде: 1)0x100000000

2)поразрядно сдвигаем маску до минимального значения вправо – 0х000000001

3) поразрядно сдвигаем маску до значение требуемого пользователем: 0х000100000

Где i – изменяемое число, а mask – маска, необходимая для установки нужного бита в состояние 1.

**Тестовый пример с представлением двоичного кода необходимых операндов выражения**

Тестовое изменяемое значение: 116 = 110 = 12

Значение маски: 20016 = 51210 = 100000000002

Выполнение операции поразрядное ИЛИ:

**Код функции**

#include <iostream>

#include <cmath>

using namespace std;

int setNBit(int i, int mask, int f)

{

mask = mask >> 8;

mask = mask << f;

i = i | mask;

return i;

}

int main()

{

cout << "Pease, enter yout num: " << endl;

int n;

cin >> n;

cout << "Please, enter number of bit, you want to change: " << endl;

int mask = 0x0x100000000;

int f

cin >> f;

cout << "After setiing num: " << setNBit(n, mask);

}

**Результаты тестирования**

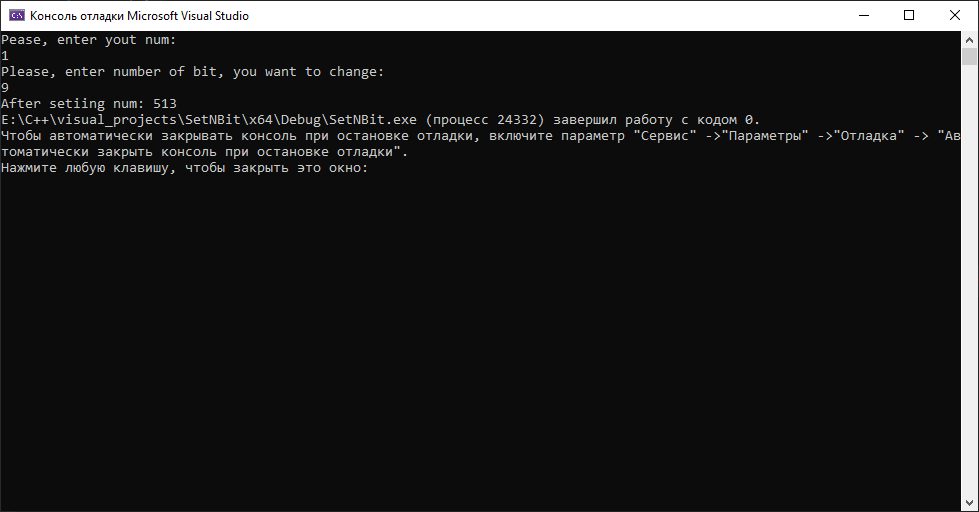


Рисунок 5 – результат выполнения 5го задания

# Задание 2

Условие задачи: реализовать задачу по сортировке данных файла, используя для представления данных файла (107 семизначных чисел) в памяти, массив битов.(алгоритм побитовой сортировки, для его реализации нужен массив членов класса Bitset)

**Разработка решения задачи пункта**

bitset — это специальный класс-контейнер, который предназначен для хранения битовых значений (элементы этого контейнера могут иметь значения: 0 и/или 1, истина или ложь).

Класс bitset очень похож на обычный массив, но, в отличии от массива, под каждый элемент объекта типа bitset отводится всего один бит, то есть пространство в памяти, которое занимают битовые маски максимально оптимизировано. Что позволяет использовать память в восемь раз меньше, чем наименьший элементарный тип данных в C++ — char).

Алгоритм на псевдокоде

//Часть 1: инициализация массива битов нулями

for i←0 t n

bit[i]:=0;

//Часть 2: Заполнение битового массива значениями Чтение числа из файла в переменную i

bit[i]:=1

//Часть 3: Формирование упорядоченного выходного файла

for i←0 t n

if bit[i]=1 then записать i в файл;

**Тестовый пример с представлением двоичного кода необходимых операндов выражения**

Тестовые значения: 7708042 8025495 8630283

Отсортированные значения:

7708042

8630283

8025495

Рисунок 6 – результат тестирования 2го задания на больших числах

**Код функции**

#include <bitset>

#include <iostream>

using namespace std;

int main()

{

bitset<1000000> data;

for (int i = 0; i < 1000000; i++)

{

data[i] = 0;

}

int num;

cin >> num;

while (num)

{

data[num] = 1;

cin >> num;

}

for (int i = 0; i < 1000000; i++)

{

if (data[i] != 0)

{

cout << i << " ";

}

}

}

**Результаты тестирования**

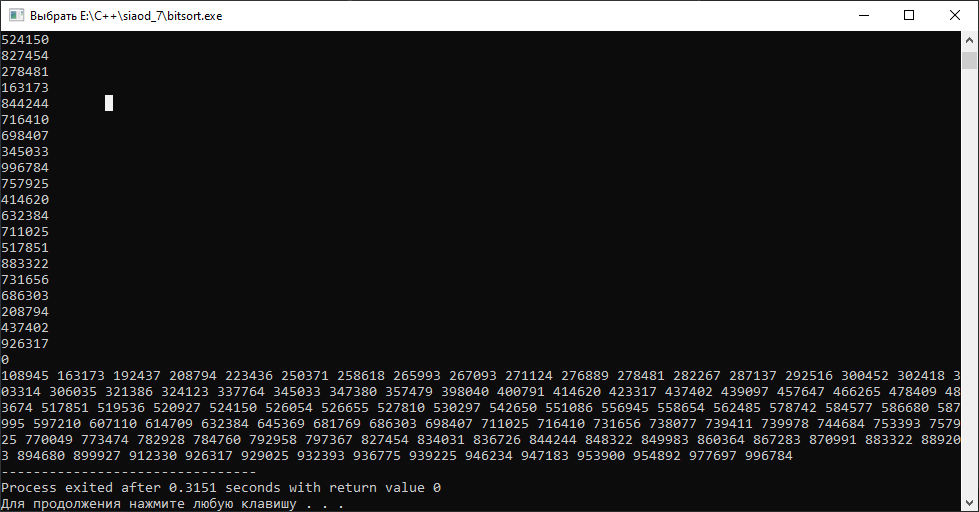


Рисунок 6 – результат работы сортировки на массиве из 100 чисел и время его выполнения

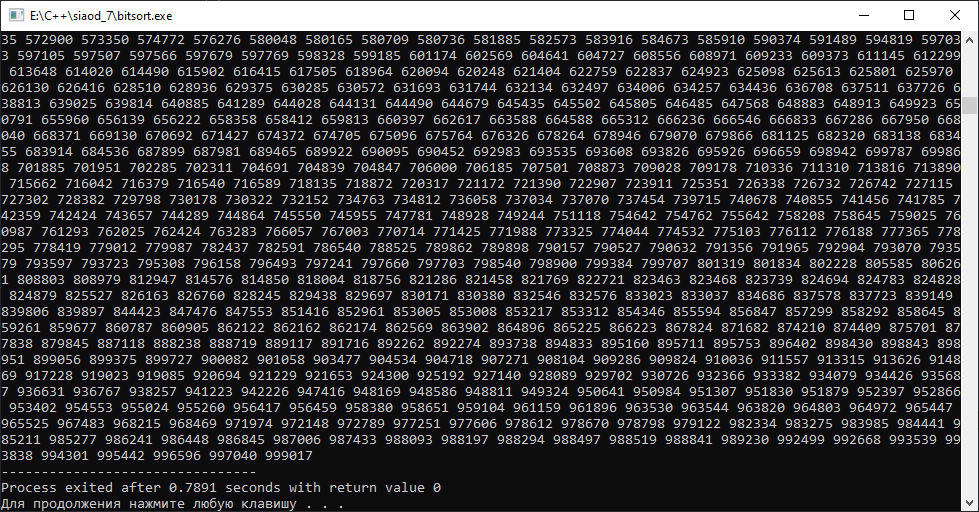


Рисунок 7 – результат работы сортировки на массиве из 1000 чисел и время его выполнения