



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 №1

**Тема:** Поразрядные операции и их применение.

**Дисциплина:** «Структуры и алгоритмы обработки данных»

Выполнил студент

Хан А.А.

группа

ИКБО-04-20

Москва 2021

## Оглавление

1. Тема .....	3
2. Цель .....	3
3. Постановка задачи .....	3
4. Код приложения .....	3
5. Тесты .....	4
6. Выводы.....	8
7. Список информационных источников.....	9

# 1. Тема

Поразрядные операции и их применение

## 2. Цель

Получить навыки применения поразрядных операций в алгоритмах

## 3. Постановка задачи

Выполнить упражнения по применению битовых операций по изменению значений битов в ячейке оперативной памяти, созданию маски для изменения значения ячейки. Создание выражения, содержащего поразрядные операции, для выполнения определенной операции над значением ячейки.

Персональный вариант:  $28\%19+1=10$

## 4. Код приложения

```
1 #include <iostream>
2 #include <bitset>
3
4 using namespace std;
5
6 void exe1( unsigned int x){
7     cout <<"Константа: \n"<< x << "_10 \t\t=" << std::bitset<16>(x) << "_2\n"; //эта функция переводит число в двоич
8     unsigned int mask=0x0F;
9     cout <<"Маска (4 младших бита): \n"<< mask << "_10 \t\t=" << std::bitset<16>(mask) << "_2\n";
10    unsigned int ans=x|mask;
11    cout <<"Результат: \n"<< ans << "_10 \t\t=" << std::bitset<16>(ans) << "_2\n";
12 }
13
14 void exe2(){
15     unsigned int x;
16     cout<<"Введите число:"<<endl;
17     cin>>x;
18     cout <<"Исходное число: \n"<< x << "_10 \t\t\t=" << std::bitset<16>(x) << "_2\n";
19     unsigned int mask=0xFAEF;
20     cout <<"Маска (4 младших бита): \n"<< mask << "_10 \t\t\t=" << std::bitset<16>(mask) << "_2\n";
21     unsigned int ans=x&mask;
22     cout <<"Результат: \n"<< ans << "_10 \t\t\t\t=" << std::bitset<16>(ans) << "_2\n";
23 }
```

```
24 void exe3(){
25     unsigned int x;
26     cout<<"Введите число:"<<endl;
27     cin>>x;
28     cout <<"Исходное число: \n"<< x << "_10 \t\t\t=" << std::bitset<16>(x) << "_2\n";
29     unsigned int m=4;
30     cout <<"Множитель: \n2^"<< m << "_10 \t\t\t=" << std::bitset<16>(m) << "_2\n";
31     unsigned int ans=x<<m;
32     cout <<"Результат: \n"<< ans << "_10 \t\t\t=" << std::bitset<16>(ans) << "_2\n";
33 }
34 void exe4(){
35     unsigned int x;
36     cout<<"Введите число:"<<endl;
37     cin>>x;
38     cout <<"Исходное число: \n"<< x << "_10 \t\t\t=" << std::bitset<16>(x) << "_2\n";
39     unsigned int m=4;
40     cout <<"Делитель: \n2^"<< m << "_10 \t\t\t=" << std::bitset<16>(m) << "_2\n";
41     unsigned int ans=x>>m;
42     cout <<"Результат: \n"<< ans << "_10 \t\t\t=" << std::bitset<16>(ans) << "_2\n";
43 }
```

```

45 void exe5(int n){
46     unsigned int x;
47     cout<<"Введите число:"<<endl;
48     cin>>x;
49     cout <<"Исходное число: \n"<< x << "_10 \t\t= " << std::bitset<16>(x) << "_2\n";
50     unsigned int mask=32768;
51     cout <<"Изменяемый бит: \n"<< n << "_10 \t\t= " << std::bitset<16>(n) << "_2\n";
52     unsigned int ans=x|(mask>>(16-n));
53     //cout <<"Результат: \n"<< (~mask>>n) << "_10 \t\t= " << std::bitset<16>(~(mask>>n)) << "_2\n";
54     cout <<"Результат: \n"<< ans << "_10 \t\t= " << std::bitset<16>(ans) << "_2\n";
55 }
56
57
58 int main()
59 {
60     setlocale(LC_ALL, "Russian"); //что бы руссифицировать консоль
61     unsigned int x=0x01; //можно менять его для 1 задания
62     //exe1(x); //для выполнения нужно раскомментировать нужный пункт и проверять результат
63     // exe2();
64     //exe3();
65     //exe4();
66     int n=4;
67     exe5(n);
68     return 0;
69 }

```

## 5. Тесты

### 1.

```

void exe1(unsigned int x){
    cout <<"Константа: \n"<< x << "_10 \t= " << std::bitset<32>(x) <<
    "_2\n";
    unsigned int mask=0x0F;
    cout <<"Маска (4 младших бита): \n"<< mask << "_10 \t= " <<
    std::bitset<32>(mask) << "_2\n";
    unsigned int ans=x|mask;
    cout <<"Результат: \n"<< ans << "_10 \t= " << std::bitset<32>(ans) <<
    "_2\n";
}

```

В 10 варианте маска представляет из себя 4 младших бита, значит ее значение  $1111_2$ . В шестнадцатеричной системе счисления – это  $0x0F$ .

Для установления значения 1 в соответствие с маской, воспользуемся операцией поразрядное ИЛИ ( $x|mask$ ). В битах, где у маски значение равно 1, в исходном сообщении, так же будет стоять 1. Отдельно стоит отметить отрицательные числа. Их представление в двоичном коде отличается (единицы заменяются на нули, и добавляется 1), из-за чего, при переводе в другую систему счисления получаются большие числа. Но при этом поразрядное ИЛИ работает точно так же.

Пример выполнения:

X=0x33

```
Константа:  
51_10    = 00000000000000000000000000000000000000000000000000000  
Маска (4 младших бита):  
15_10    = 00000000000000000000000000000000000000000000000000000  
Результат:  
63_10    = 00000000000000000000000000000000000000000000000000000
```

X=-0x33

```
Константа:  
4294967245_10    = 11111111111111111111111111001101_2  
Маска (4 младших бита):  
15_10    = 00000000000000000000000000001111_2  
Результат:  
4294967247_10    = 11111111111111111111111111001111_2
```

X=0x0F

```
Константа:  
15_10    = 000000000000000000000000000000001111_2  
Маска (4 младших бита):  
15_10    = 000000000000000000000000000000001111_2  
Результат:  
15_10    = 000000000000000000000000000000001111_2
```

2.

```
void exe2(){
    unsigned int x;
    cout<<"Введите число:"<<endl;
    cin>>x;
    cout <<"Исходное число: \n"<< x << "_10 \t\t\t=" <<
std::bitset<16>(x) << "_2\n";
    unsigned int mask=0xFAEF;
    cout <<"Маска (4 младших бита): \n"<< mask << "_10 \t\t\t=" <<
std::bitset<16>(mask) << "_2\n";
    unsigned int ans=x&mask;
    cout <<"Результат: \n"<< ans << "_10 \t\t\t=" << std::bitset<16>(ans)
<< "_2\n";
}
```

В 10 варианте в 3, 11 и 5 битах маски должны стоять 0. Из этого условия получается двоичное число 1111101011101111, что в шестнадцатеричной системе FAEF. В задании требуется поставить 0 на заданные биты. Для этого воспользуемся операцией поразрядного И ( $x \& \text{mask}$ ).

Работа с отрицательными числами ничем не отличается от предыдущего задания.

Пример выполнения:

X=51

```

Введите число:
51
Исходное число:
51_10 = 0000000000110011_2
Маска (4 младших бита):
64239_10 = 1111101011101111_2
Результат:
35_10 = 0000000000100011_2

```

X=-51

```

Введите число:
-51
Исходное число:
4294967245_10 = 1111111111001101_2
Маска (4 младших бита):
64239_10 = 1111101011101111_2
Результат:
64205_10 = 1111101011001101_2

```

### 3.

```

void exe3(){
    unsigned int x;
    cout<<"Введите число:"<<endl;
    cin>>x;
    cout <<"Исходное число: \n"<< x << "_10 \t\t=" << std::bitset<16>(x) <<
    "_2\n";
    unsigned int m=4;
    cout <<"Множитель: \n"<< m << "_10 \t\t=" << std::bitset<16>(m) <<
    "_2\n";
    unsigned int ans=x<<m;
    cout <<"Результат: \n"<< ans << "_10 \t\t=" << std::bitset<16>(ans) <<
    "_2\n";
}

```

В 10 варианте множитель равен 16. Для умножения числа нужно воспользоваться операцией побитового сдвига влево ( $x \ll n$ ), а число 16 преобразовать к виду  $2^n$ . Таким образом получим, что сдвиг необходимо произвести на 4. На примерах будет видно, как осуществляется сдвиг.

Пример выполнения:

X=51 (51\*16=816)

```

Введите число:
51
Исходное число:
51_10      = 0000000000110011_2
Множитель:
2^4_10      = 0000000000000100_2
Результат:
816_10      = 0000001100110000_2

X=0
Введите число:
0
Исходное число:
0_10        = 0000000000000000_2
Множитель:
2^4_10      = 0000000000000100_2
Результат:
0_10        = 0000000000000000_2

```

#### 4.

В 10 варианте делитель равен 16. Для деления числа нужно воспользоваться операцией побитового сдвига вправо ( $x \gg n$ ), а число 16 преобразовать к виду  $2^n$ . Таким образом получим, что сдвиг необходимо произвести на 4. На примерах будет видно, как осуществляется сдвиг.

#### Пример выполнения:

X=32 (32/16=2)

```

Введите число:
32
Исходное число:
32_10      = 0000000000100000_2
Делитель:
2^4_10      = 0000000000000100_2
Результат:
2_10       = 0000000000000010_2

```

X=51 (остаток от деления опускается)

```

Введите число:
51
Исходное число:
51_10      = 0000000000110011_2
Делитель:
2^4_10      = 0000000000000100_2
Результат:
3_10       = 0000000000000011_2

```

#### 5.

```

void exe5(int n){
    unsigned int x;
    cout<<"Введите число:"<<endl;
    cin>>x;
}

```

```

    cout << "Исходное число: \n" << x << "_10 \t\t= " << std::bitset<16>(x) <<
    "_2\n";
    unsigned int    mask=32768;
    cout << "Изменяемый бит: \n" << n << "_10 \t\t= " << std::bitset<16>(n)
    << "_2\n";
    unsigned int ans=x|(mask>>(16-n));
    //cout << "Результат: \n" << (~(mask>>n)) << "_10 \t\t= " <<
    std::bitset<16>(~(mask>>n)) << "_2\n";
    cout << "Результат: \n" << ans << "_10 \t\t= " << std::bitset<16>(ans) <<
    "_2\n";
}

```

Необходимо установить n-ый бит в 1, используя маску пункта 2. Маска равна числу 32768 (число, где старший разряд равен 1). Для того, чтобы установить 1 в необходимый бит, нужно сдвинуть маску на число 16-n (16 разрядов в числе, n отсчитывается с начала), а затем применить поразрядное ИЛИ.

#### Пример выполнения:

x=51

```

Введите число:
51
Исходное число:
51_10          = 00000000000110011_2
Изменяемый бит:
4_10           = 0000000000000100_2
Результат:
59_10          = 00000000000111011_2

```

X=0

```

Введите число:
0
Исходное число:
0_10           = 0000000000000000_2
Изменяемый бит:
4_10           = 0000000000000100_2
Результат:
8_10           = 0000000000001000_2

```

## 6. Выводы

В результате проделанной работы, я научилась выполнять упражнения с применением битовых операций, изменять значения битов в ячейке оперативной памяти, создавать маски для значений ячейки, создавать выражения, содержащие поразрядные операции, для выполнения определенной операции над значением ячейки.



## **7. Список информационных источников**

1. Лекционный материал по структурам и алгоритмам обработки данных Сартакова М. В.