

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**  
«Информатика»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

«Системы счисления. Степень двойки»

**Выполнил:**

Суханкулиев Мухаммет,  
студент группы N3146

---

(подпись)

**Проверил:**

Безруков Вячеслав Алексеевич,  
преподаватель информатики

---

(отметка о выполнении)

---

(подпись)

Санкт-Петербург

2023 г.

## **СОДЕРЖАНИЕ**

Введение

- 1 Представление и операции над числами в различных системах счисления
  - 1.1 Задание
  - 1.2 Ход работы
- 2 Алгоритм представления степени двойки
  - 2.1 Задание
  - 2.2 Ход работы

Заключение

## ВВЕДЕНИЕ

Цель работы:

- Изучить представление чисел в различных системах счисления и выполнение арифметических операций над ними;
- Изучить алгоритм, позволяющий узнать, является ли число степенью двойки.

## 1. ПРЕДСТАВЛЕНИЕ И ОПЕРАЦИИ НАД ЧИСЛАМИ В РАЗЛИЧНЫХ СИСТЕМАХ СЧИСЛЕНИЯ

### Задание

- Представить произвольное число в произвольной системе счисления;
- Примеры перевода чисел в разные системы счисления и операции над ними.

### Ход работы

#### Представить произвольное число в произвольной системе счисления

$123_{10}$  представим через степени числа 10:

$$1 * 10^2 + 2 * 10^1 + 3 * 10^0 = 123$$

Переведем  $123_{10}$  в семеричную систему счисления:

$$123_{10} = 234_7$$

$123_{10}$  представим через степени числа 7:

$$2 * 7^2 + 3 * 7^1 + 4 * 7^0 = 123$$

123	14	6
12	2	3
2	0	2
$123_{10} = 234_7$		

#### Примеры перевода чисел в разные системы счисления и операции над ними

a)  $1234_{10}$  в двадцатеричную систему счисления:

$$1234_{10} = 31E_{20}$$

Представление через степени:

$$3 * 20^2 + 1 * 20^1 + E * 20^0 = 1234$$

1234	64	14
61	3	14
3	0	3
$1234_{10} = 31E_{20}$		

b)  $321_{10}$  в троичную систему счисления:

$$321_{10} = 102220_3$$

Представление через степени:

$$1 * 3^5 + 0 * 3^4 + 2 * 3^3 + 2 * 3^2 + 2 * 3^1 + 0 * 3^0 = 321$$

321	81	27	9	3	1
32	1	0	0	0	1
16	1	0	0	0	1
8	0	2	0	0	0
4	0	2	0	0	0
2	0	2	0	0	0
1	0	2	0	0	0
$321_{10} = 102220_3$					

c)  $999_{10}$  в восьмеричную систему счисления:

$$999_{10} = 1747_8$$

Представление через степени:

$$1 * 8^3 + 7 * 8^2 + 4 * 8^1 + 7 * 8^0 = 999$$

999	512	128	64	32	16	8	4
99	1	5	3	3	3	3	3
49	0	5	3	3	3	3	3
24	0	5	3	3	3	3	3
12	0	5	3	3	3	3	3
6	0	5	3	3	3	3	3
3	0	5	3	3	3	3	3
$999_{10} = 1747_8$							

a) Сложим числа  $234_7$  и  $102220_3$ . Переведем все в одну систему счисления для удобства вычислений, пусть это будет троичная, чтобы перевести только одно число.

$$234_7 = 123_{10}; 123_{10} = 11120_3$$

$$11120_3 + 102220_3 = 121110_3 = 1 * 3^5 + 2 * 3^4 + 1 * 3^3 + 1 * 3^2 + 1 * 3^1 + 0 * 3^0 = 444$$

( $123 + 321 = 444$  - верно)

123	81	27	9	3	1
12	1	0	0	0	1
6	1	0	0	0	1
3	0	2	0	0	0
1	0	2	0	0	0
$123_{10} = 11120_3$					

b) Вычтем из числа  $31E_{20}$  число  $1747_8$ . Попробуем перевести  $1747_8$  в двадцатеричную систему счисления:

$$1747_8 = 999_{10}; 999_{10} = 29J_{20}$$

$$31E_{20} - 29J_{20} = BF_{20} = B * 20^1 + F * 20^0 = 235$$

( $1234 - 999 = 235$  - верно)

1747	400	100	40	20	10	5
174	4	14	7	7	7	7
87	0	14	7	7	7	7
43	0	14	7	7	7	7
21	0	14	7	7	7	7
10	0	14	7	7	7	7
5	0	14	7	7	7	7
$1747_{10} = 29J_{20}$						

Санкт-Петербург

2023 г.

с) Умножим число  $234_7$  на  $1747_8$ .

$$1747_8 = 999_{10}; 999_{10} = 2625_7$$

$$2625_7 * 234_7 = 1021146_7 = 1*7^6 + 0*7^5 + 2*7^4 + 1*7^3 + 1*7^2 + 4*7^1 + 6*7^0 = 122877$$

$$(999 * 123 = 122877 - \text{верно})$$



## 2. АЛГОРИТМ ПРЕДСТАВЛЕНИЯ СТЕПЕНИ ДВОЙКИ

### Задание

- Создать алгоритм, позволяющий узнать, является ли число степенью двойки.

### Ход работы

Число, равное самому себе с противоположным знаком, имеет все биты, противоположные по состоянию тем, что есть в исходном числе.

Если число  $x$  является степенью двойки, то в его двоичном представлении есть один установленный бит в левом крайнем разряде. В этом случае результат операции побитового умножения  $x$  на  $-x$  будет равен самому числу  $x$ .

Например, рассмотрим число 8. В двоичном виде оно представлено как 1000. Число, равное самому себе с противоположным знаком, это  $-8$ , которое в двоичном виде представлено как 1111. Результат операции побитового умножения  $8 * -8$  равен 8.

### Алгоритм:

- Проверка, является ли число положительным. Если число отрицательное, то оно не может быть степенью двойки.
- Побитово умножить число на число, равное самому себе с противоположным знаком.
- Если результат операции побитового умножения равен самому числу  $x$ , то число является степенью двойки.

### Код на C++

```
#include <iostream>

int main() {
    std::string input;
    while (true) {
        std::cout << "Введите число x (close для выхода): ";
        std::cin >> input;
        if (input == "close") {
            break;
        }

        int x = std::stoi(input);

        // Проверяем, является ли число положительным.
        if (x <= 0) {
            std::cout << "Число x должно быть положительным." << std::endl;
            continue;
        }

        // Побитово умножаем число на число, равное самому себе с противоположным знаком.
        int y = x & (-x);

        // Если результат операции побитового умножения равен самому числу x, то число является степенью двойки.
        if (y == x) {
            std::cout << "Число x является степенью двойки." << std::endl;
        } else {
            std::cout << "Число x не является степенью двойки." << std::endl;
        }
    }

    return 0;
}
```

## **Заключение**

В ходе лабораторной работы были изучены основы представления чисел в различных системах счисления и выполнения арифметических операций над ними. Были рассмотрены примеры перевода чисел в разные системы счисления и операции над ними.

Мы узнали, что степень двойки имеет один установленный бит в левом крайнем разряде. Это позволяет нам разработать алгоритм, который работает на этом свойстве.