ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИСCЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет безопасности информационных технологий

**Дисциплина:**

«Информатика»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

**Выполнил:**

Студент гр. N3151, Мочеков Семён Сергеевич

**Проверил:**

Безруков Вячеслав Алексеевич

Санкт-Петербург

2023

**I. ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ**

Позиционная система счисления – система счисления, в которой значение (вес) каждой цифры изменятся от её положения (позиции) в последовательности цифр, изображающих число. Размер такой системы счисления – её основание (только целое число ), набор чисел, из которых она состоит – базис. Если основание позиционной системы счисления = N, то размер её базиса тоже равен N, а сам он содержит числа: {0, 1, …, N-1}

Представление числа в произвольной позиционной системе счисления с основанием N аналогично представлению в Десятичной, однако каждый старший разряд равняется N единицам предыдущего разряда. Тогда для системы счисления с основанием **p** и базисом **{0, 1, 2 …, ap}** представление числа **d**:

***d = ak \* pk + ak-1 \* pk-1 + ak-2 \* pk-2 + … + a2 \* p2 + a1 \* p1 + a0 \* p0 + a-1 \* p-1 + a-2 \* p-2 + …***

где k – номер позиции цифры в числе.

Так для 16-ичной: D1916 = D \* 162 + 1 \* 161 + 9 \* 160

Для 4-ичной: 13324 = 1 \* 43 + 3 \* 42+ 3 \* 41+ 2 \* 40

Перевод между системами счисления:

DF145C16 = 6761213410

3.1410 = 3.0(6600)7

OBEY35 = 42370712

1100102 = 3B13

В произвольных позиционных системах счисления арифметические операции над числами осуществляются аналогично 10-ичной системе счисления, однако перенос значений в следующий разряд (увеличение при сложении/умножении, уменьшение при вычитании/делении) происходит после достижения значения равного основанию при увеличении, и отсчитывается от основания системы счисления при уменьшении.

Сложение: 6-ная: 15 + 5 = 246

16-ная: D4 + F =E316

Вычитание: 2-ная: 101100 – 10010 = 110102

8-ная: 7612 – 234 = 73568

Умножение: 7-ная: 54 \* 3 = 2257

16-ная: F7 \* 3 = 2E516

Деление: 10-ная: 140 : 2 = 7010

8-ная: 76 : 2 = 378

Изображение выглядит как Шрифт, типография, рукописный текст, каллиграфия

Автоматически созданное описаниеИзображение выглядит как Шрифт, символ, рукописный текст, типография

Автоматически созданное описаниеИзображение выглядит как Шрифт, рукописный текст, символ, каллиграфия

Автоматически созданное описание

Изображение выглядит как рукописный текст, Шрифт, каллиграфия, типография

Автоматически созданное описание

Изображение выглядит как Шрифт, типография, каллиграфия, рукописный текст

Автоматически созданное описание

**II. ПРОВЕРКА, ЯВЛЯЕТСЯ ЛИ НАТУРАЛЬНОЕ ЧИСЛО СТПЕНЬЮ ДВОЙКИ**

Используя язык программирования C:

#include <stdio.h>

int main(){

    int Number = 2048; // так же проверим для Number = 1344

    // Исследуем результат битовой операции "И" чисел (Number) и (Number-1)

    int log\_and = Number & (Number-1)

    if (log\_and == 0) {

        printf("Число %d - степень 2. Операция \"И\": %d\n", Number, log\_and);

        // Если Number является степенью 2, то результат битовой операции "И" равен 0

    }

    else {

        printf("Число %d - не степень 2.Операция \"И\": %d\n", Number, log\_and);

        // В противном случае, результат будет любым другим

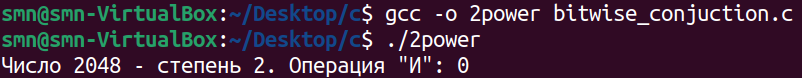
    }

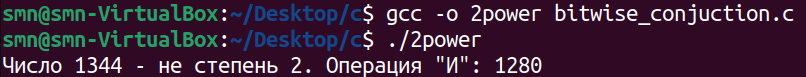
    printf("\n");

    return 0;

}

**Результаты работы программы:**





**III. ЧИСЛА В ПАМЯТИ КОМПЬЮТЕРА**

Распространённым порядком байтов памяти компьютера является порядок от младшего к старшему байту, если данные состоят из нескольких. Однако числа в позиционных системах счисления записываются в порядке от старшего к младшему. Значит в памяти числа записаны в зеркальном виде. Тогда младший байт будет иметь меньший адрес в памяти среди других и считаться адресом всего числа.

Использую язык программирования C:

#include <stdio.h>

int main(){

    int Number = 0xF15AD788;

    int \*Index = &Number; // Получаем адрес числа в памяти компьютера

    // Выводим каждый байт из памяти компьютера

    for (int i = 0; i<4; i++) {

        unsigned char Bytte = \*((char\*)Index+i);

        printf("| %x |", Bytte);

    }

    printf("\n");

    return 0;

}

**Результат работы программы:**

