**Отчет по лабораторной работе №**

**Тема**

*РК6-16Б*

16. Петраков Станислав Альбертович

Преподаватель: Федорук Елена Владимировна

**Цель работы**

Изучение алгоритма перевода чисел из одной позиционный системы счисления (ПСС) в другую.

**Ответы на контрольные вопросы**

1. *Что такое позиционная система счисления?*

Позиционная система счисления (ПСС) — система, для которой значение цифры определяется ее положением в числе.

Любая позиционная система счисления характеризуется основанием (базисом). Для естественной позиционной системы — целое положительное число.

1. *Что такое основание позиционной системы счисления?*

Основание позиционной системы счисления — количество знаков или символов, используемых для изображения числа в данной системе.

1. *Что такое алфавит позиционной системы счисления?*

Алфавит ПСС – набор символов, используемый для обозначения цифр.

1. *Переведите число 6810 в двоичную, восьмеричную и шестнадцатеричную системы счисления.*

*6810 = 10001002*

*6810 = 1048*

*6810 = 4416*

1. *Переведите число 0.72510 в двоичную, восьмеричную и шестнадцатеричную системы счисления.*

*0.72510 = 0.101(1100)2*

*0.72510 = 0.5(6314)8*

*0.72510 = 0.B(9)16*

1. *Каково внутреннее (машинное) представление целых чисел?*

Естественная форма представления числа в компьютере характеризуется тем, что положение его разрядов в автоматном представлении остается всегда постоянным независимо от величины самого числа. Эта форма записи чисел также называется представлением чисел с фиксированной точкой.

Так как числа бывают положительные и отрицательные, то формат (разрядная сетка) автоматного представления числа разбивается на знаковую часть и поле числа. В знаковую часть записывается 0 для положительного числа и 1 для отрицательного.

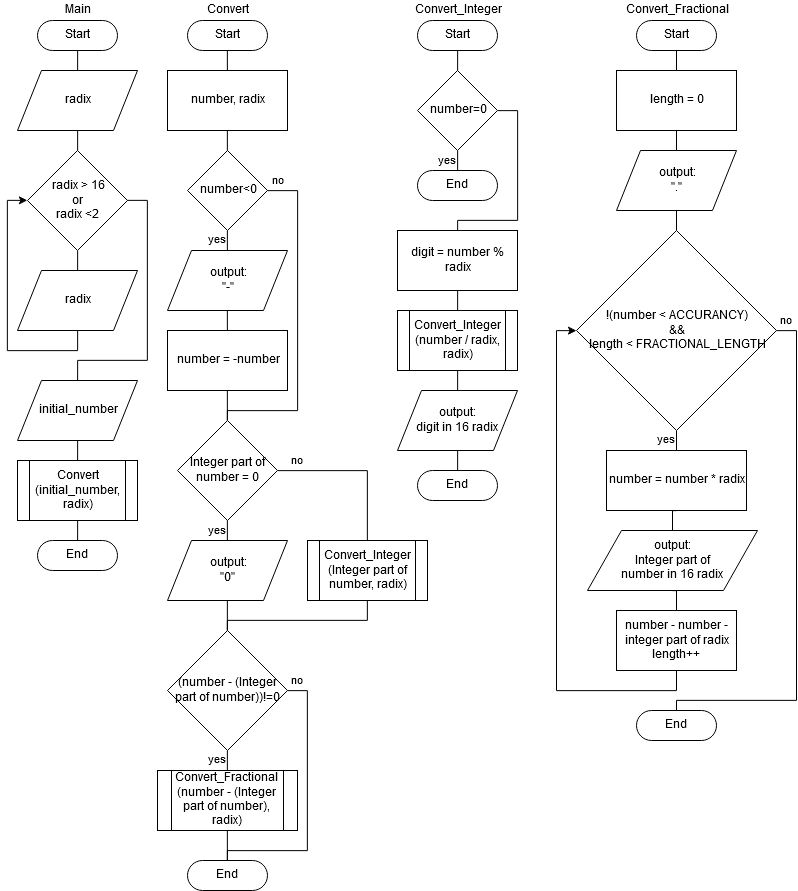
1. *Каково внутреннее (машинное) представление вещественных чисел?*

Формат автоматного представления числа с ПТ должен содержать знаковые части и поля мантиссы и порядка. Выделяются специальные разряды для изображения знака числа (мантиссы) и знака порядка. Кодирование знаков такое же, как для формы с фиксированной точкой. Этот формат является машинно-зависимым.

**Программа**

*Задание:* *Разработайте программу перевода десятичного вещественного числа в новую систему счисления. Само число и основание новой системы счисления должны вводится со стандартного потока ввода. Необходимо организовать проверку правильности ввода данных. Основание может принадлежать отрезку [2,16]. Введенное число должно быть разбито на целую и дробную части. Каждая часть переводится по своему алгоритму. Результат перевода вывести на стандартное устройство вывода.*

**Блок-схема:**

****

**Сама программа:**

#include <stdio.h>

#include <stdlib.h>

#define FRACTIONAL\_LENGTH 10

#define ACCURANCY 0.000001

void Convert(double number, int radix);

void Convert\_Integer(int number, int radix);

void Convert\_Fractional(double number, int radix);

int Get\_Int()

{

int a;

while (scanf("%d", &a) != 1)

{

while (getchar() != '\n');

printf("Incorrect input. Try again: ");

}

return a;

}

double Get\_Double()

{

double a;

while (scanf("%lf", &a) != 1)

{

while (getchar() != '\n');

printf("Incorrect input. Try again: ");

}

return a;

}

void Convert(double number, int radix)

{

if (number < 0)

{

printf("-");

number = -number;

}

if ((int)number == 0)

printf("0");

else

Convert\_Integer((int)number, radix);

if ((number - (int)number) != 0)

Convert\_Fractional(number - (int)number, radix);

}

void Convert\_Integer(int number, int radix)

{

if (number == 0)

return;

int digit = number % radix;

Convert\_Integer(number / radix, radix);

printf("%X", digit);

}

void Convert\_Fractional(double number, int radix)

{

int length = 0;

printf(".");

while (!(number < ACCURANCY) && length < FRACTIONAL\_LENGTH)

{

number = number \* radix;

printf("%X", (int)number);

number = number - (int)number;

length++;

}

}

int main()

{

int radix;

printf("Enter radix in range [2;16]: ");

radix = Get\_Int();

while (radix > 16 || radix < 2)

{

printf("Radix is not in range [2;16]. Enter again: ");

radix = Get\_Int();

}

double initial\_number;

printf("Enter number to convert it to %d radix: ", radix);

initial\_number = Get\_Double();

printf("Your number in %d radix is: ", radix);

Convert(initial\_number, radix);

return 0;

}

**Тестовые примеры**

|  |  |  |  |
| --- | --- | --- | --- |
| **Число в 10** | **Основание системы счисления** | **Ручной подсчет** | **Машинный подсчет** |
| **546** | 2 | 1000100010 | 1000100010 |
| 8 | 1042 | 1042 |
| 16 | 222 | 222 |
| 9 | 666 | 666 |
| **-654** | 2 | -1010001110 | -1010001110 |
| 8 | -1216 | -1216 |
| 16 | -28E | -28E |
| 4 | -22032 | -22032 |
| **0.564** | 2 | 0.1001000001 | 0.1001000001 |
| 8 | 0.4406111564 | 0.4406111564 |
| 16 | 0.90624DD2F1 | 0.90624DD2F1 |
| 12 | 0.692712B866 | 0.692712B866 |
| **-0.315** | 2 | -0.0101000010 | -0.0101000010 |
| 8 | -0.2412172702 | -0.2412172702 |
| 16 | -0.50A3D70A3D | -0.50A3D70A3D |
| 6 | -0.1520123501 | -0.1520123501 |
| **697.431** | 2 | 1010111001.0110111001 | 1010111001.0110111001 |
| 8 | 1271.3345300406 | 1271.3345300406 |
| 16 | 2B9.6E56041893 | 2B9.6E56041893 |
| 7 | 2014.3005555014 | 2014.3005555014 |
| **-463.489** | 2 | -111001111.0111110100 | -111001111.1001011011 |
| 8 | -717.3722743247 | -717.3722743247 |
| 16 | -1CF.7D2F1A9FBE | -1CF.7D2F1A9FBE |
| 14 | -251.6BBB5D1655 | -251.6BBB5D1655 |
| **0** | 2 | 0 | 0 |
| 8 | 0 | 0 |
| 16 | 0 | 0 |
| 4 | 0 | 0 |

**Результаты работы программы**

Программа переводит число в необходимую систему счисления.

**Вывод**

Научился работать с %X в функции printf. Понял, как можно сделать перевод целой части 2 способами. Массивом и рекурсией.