

КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Программирование поразрядных операций

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2022

1.Цель работы: целью работы является изучение логических операций типа НЕ, И, ИЛИ, исключающее ИЛИ и операций сдвига

2.Задание:

Согласно варианту 17:

Вариант 17

Блок управления памятью в операционной системе имеет формат:

№ разряда	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Значение	0	0	0	0	0	0	0	0	0	P	L	L	L	L	L	L

где 0 ... 0 – идентификатор владельца блока, P – признак программного блока, L ... L–размер блока.

Составить две программы, первая из которых вводит составные части структуры данных, приведённой в индивидуальном варианте, как десятичные числа и формирует из них заданную упакованную структуру как 16-ричное число. Вторая программа вводит упакованную структуру как 16-ричное число и выводит значения отдельных её составных частей как десятичные числа.

3.Описание созданных функций:

Для реализации задания нам потребуются следующие функции:

Имя: main

Назначение: Запрос от пользователя операндов в регистров для получения формата команды сложения.

Входные данные: нет.

Выходные данные: нет.

Побочный эффект: отсутствует.

Тестовые данные:

0	8
P	1
L	11
Ответ	0x84b

0	3
P	4
L	5
Ответ	Числа не входят в диапазон

Прототип: int main()

Псевдокод

Функция проверки ввода данных read_int()

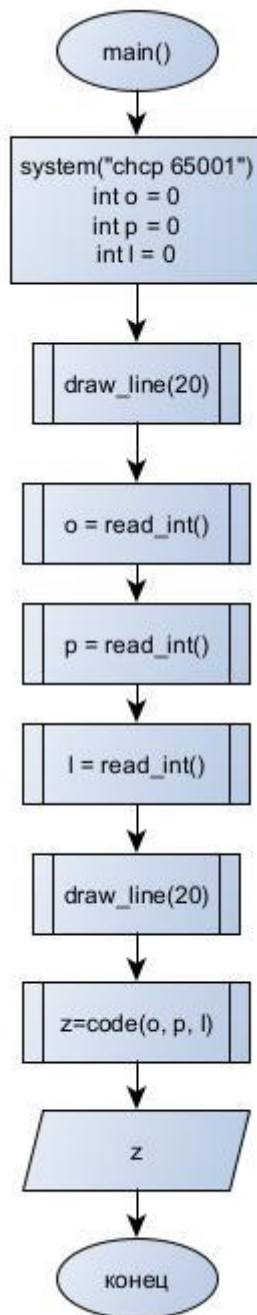
Функция проверки ввода данных read_int()

Функция проверки ввода данных read_int()

Вызов функции code()

Вывод в консоль вернувшегося значения

Блок-схема:



Имя: `read_int`

Назначение: проверка переменной а на корректность ввода

Входные данные: `x`(целочисленное значение).

Выходные данные: `x`(целочисленное значение).

Побочный эффект: отсутствует.

Прототип: `double read_int()`

Псевдокод:

Цикл продолжается до тех пор, пока пользователь не введет корректное значение

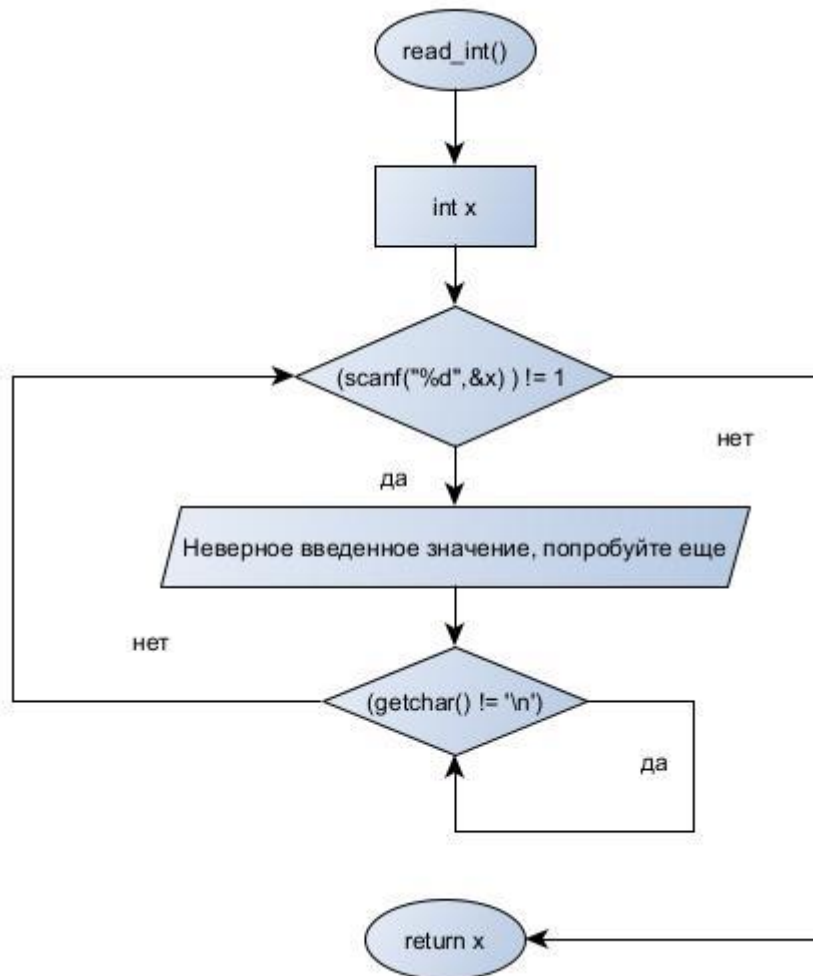
Ввод числа `x`

Если предыдущее извлечение оказалось неудачным, то...

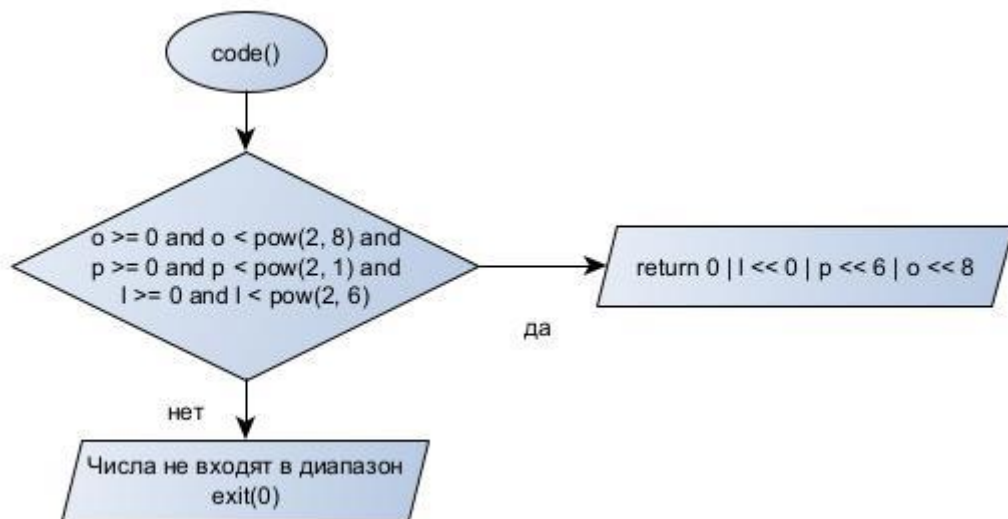
Возвращаем `scanf` в 'обычный' режим работы(обнуление битов состояния)

Удаляем значения предыдущего ввода из входного буфера

Если всё хорошо, то возвращаем x
Возврат значения x
Блок-схема:



Имя: code
Назначение: перевод значений в формат.
Входные данные: o, p, l.
Выходные данные: формат сложения
Побочный эффект: отсутствует.
Прототип: unsigned short code()
Псевдокод: Проверка корректности данных
Возврат команды
Блок-схема:



Пример выполнения программы:

Имя: main

Назначение: Ввод формата сложения для получения введенных значений

Входные данные: нет.

Выходные данные: нет.

Побочный эффект: отсутствует.

Тестовые данные:

X (16-ричное число от 0 до 0xFFFF):	0xfe
Идентификатор владельца блока:	0
Признак программного блока:	1
Размер блока:	62

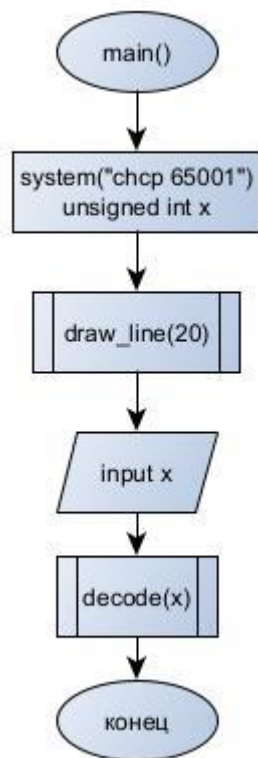
Прототип: int main()

Псевдокод:

Ввод X (Формат сложения)

Вызов функции decode()
Вывод в консоль вернувшегося значения

Блок-схема:



Имя: decode

Назначение: Перевод формата в значения.

Входные данные: X (формат сложения).

Выходные данные: o, p, l.

Побочный эффект: отсутствует.

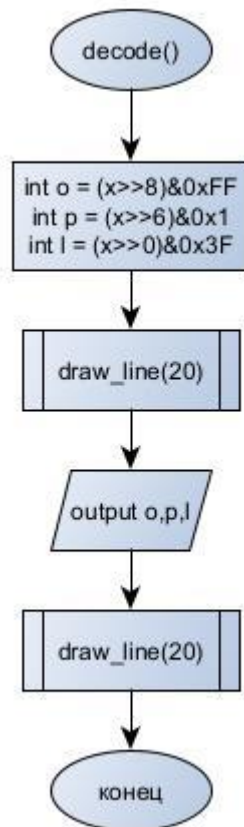
Прототип: void decode()

Псевдокод:

Перевод формата в значение

Вывод всех значений

Блок-схема:



Пример выполнения программы:

```

-----
Введите команду сложения (16-ричное число от 0 до 0xFFFF): 0x84b
-----
Идентификатор владельца блока: 8
Признак программного блока: 1
Размер блока: 11
-----

I:\3\3\Debug\3.exe (процесс 2012) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
  
```

4.Текст программы

Source.cpp:

```

#include <iostream>
#include "lib.h"
#include "memory.h"
#include <windows.h>
using namespace std;

int main() {
    // смена кодировки
    SetConsoleCP(1251);
  
```

```

SetConsoleOutputCP(1251);

int o = 0;
int p = 0;
int l = 0;

draw_line(20);

// вводим числа
cout << "Идентификатор владельца блока: ";
o = read_int();

cout << "Признак программного блока: ";
p = read_int();

cout << "Размер блока: ";
l = read_int();

draw_line(20);

// собираем
unsigned short z = code(o, p, l);

// выводим собранное число
cout << "Блок управления памятью: 0x" << hex << z << endl;

return 0;
}

```

3.cpp:

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include "lib.h"
#include "memory.h"
#include <windows.h>
using namespace std;

int main() {
    // смена кодировки
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    unsigned int x;

    draw_line(20);

    cout << "Введите команду сложения (16-ричное число от 0 до 0xFFFF): ";
    cin >> hex >> x;

    decode(x);

    return 0;
}

```

memory.h:

```

#include <iostream>
#include <cmath> // для pow (возведение в степень)

using namespace std;

// функция для сбора в команду сложения
unsigned short code(int o, int p, int l) {
    if ( // проверка

```



```

        (o >= 0 && o < pow(2, 8)) &&
        (p >= 0 && p < pow(2, 1)) &&
        (l >= 0 && l < pow(2, 6))
    ) {
        // если всё хорошо то объединяем числа
        return o | l << 0 | p << 6 | o << 8;
    }
    else {
        /*
            если числа не входят в диапазон
            (не помещаются в переменную, то выводим пользователю
            об этом сообщение и выходим из программы)
        */
        cout << "Числа не входят в диапазон" << endl;

        exit(0);
    }
}

void decode(unsigned short x) {
    /*
        >> - побитовый сдвиг
        &число - то сколько надо бит вытащить
    */
    int o = (x >> 8) & 0xFF;
    int p = (x >> 6) & 0x1;
    int l = (x >> 0) & 0x3F;

    draw_line(20);
    // выводим числа
    // dec - вывести число в 10 системе счисления
    cout << dec << "Идентификатор владельца блока: " << o << endl;
    cout << dec << "Признак программного блока: " << p << endl;
    cout << dec << "Размер блока: " << l << endl;
    draw_line(20);
}

```

lib.h:

```

#include <iostream>

using namespace std;

// функция рисующая полосу для разделения
void draw_line(int size) {
    for (int i = 0; i < size; i++)
        cout << '-';
    cout << endl;
}

int read_int() {
    int x;
    while ((scanf("%d", &x)) != 1) {
        printf("Неверное введенное значение, попробуйте еще: ");
        while (getchar() != '\n');
    }
    return x;
}

```

5. Пример выполнения программы

```
Консоль отладки Microsoft Visual Studio
-----
Идентификатор владельца блока: 8
Признак программного блока: 1
Размер блока: 11
-----
Блок управления памятью: 0x84b

D:\лабы\3\3\Debug\code.exe (процесс 7640) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

```
Консоль отладки Microsoft Visual Studio
-----
Введите команду сложения (16-ричное число от 0 до 0xFFFF): 0x84b
-----
Идентификатор владельца блока: 8
Признак программного блока: 1
Размер блока: 11
-----

D:\лабы\3\3\Debug\3.exe (процесс 14512) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

6. Анализ результатов и вывода

В результате выполнения лабораторной работы были изучены принципы работы с поразрядными операциями.

К достоинствам программы можно отнести:

- Производится проверка входных данных.
- Программа корректно переводит данные.

Из недостатков можно отметить:

- Программа не оптимизирована.
- Нет всех комментариев к коду.

