ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Лабораторная работа 3. Программирование поразрядных операций |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № |  |  |  |  |  |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы**

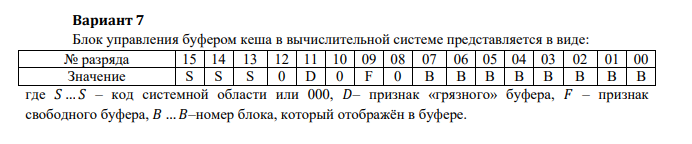
Целью работы является изучение поразрядных операций типа НЕ, И, ИЛИ, исключающее ИЛИ и операций сдвига.

**Задание на лабораторную работу**

Составить две программы, первая из которых вводит составные части структуры данных, приведѐнной 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 еѐ составных частей как десятичные числа.

Программы должны быть оформлены как решение с двумя проектами (см. «Приложение 8. Полезные советы по работе в среде VisualC++ 2008» на стр. 82).

Для чтения и записи в потоки ввода/вывода следует использовать манипуляторы 𝑑𝑒𝑐 и ℎ𝑒𝑥.



**Ход выполнения:**

**Использованные функции:**

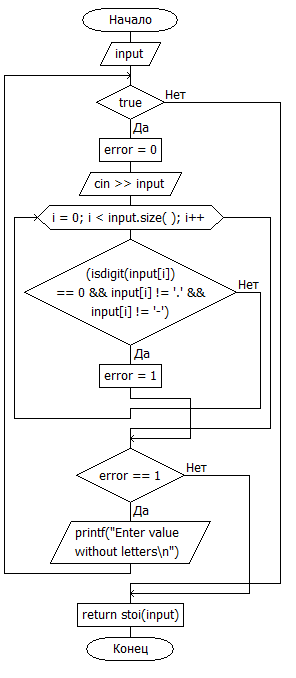
**Имя:** fill  
**Назначение:** улучшенный ввод с клавиатуры, Который запрашивает строку, проверяет ее на возможность перевода в double

**Входные данные:** Запрос с клавиатуры

**Выходные данные:** Ввод с клавиатуры, преобразованный в double

**Побочные эффекты:** Отсутствует

**Блок схема:**

****

**Псевдокод**

Ввод с клавиатуры

Пока 1

Проверка каждого элемента введенной строки

Если элемент не подходит условиям isdigit()

Ошибка=1

Если ошибки есть, просим ввести заново

Иначе конец

**Имя:** main()

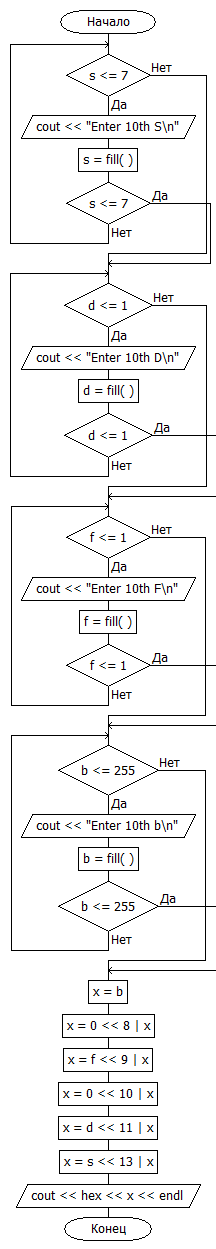
**Назначение:** запаковать данные согласно варианту в одну ячейку памяти

**Входные данные:** 4 ввода с клавиатуры

**Выходные данные:** ячейка памяти, содержащая в себе 4 переменные

**Побочные эффекты:** Шестнадцатеричный формат

**Блок-схема:**

****

**Псевдокод:**

Объявления 5 переменных

Ввод 4 переменных с клавитуры с параллельной проверкой область допостимых значений

Упаковка всех введенных значений в X

Вывод X

**Имя:** unpack()

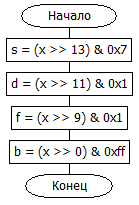
**Назначение:** распаковать данные согласно варианту из одной ячейки памяти

**Входные данные:** 1 ввод с клавиатуры

**Выходные данные:** 4 переменные, восстановленные из упаковки

**Побочные эффекты:** нет

**Блок-схема:**



Псевдокод:

Ввод упаковки с клавиатуры

Пошаговое извлечение каждого элемента из ячейки памяти

Их вывод

**Листинг всей программы**

#include <iostream>

#include <string>

using namespace std;

double fill() {

std::string input;

while (true) //проверка введенного числа

{

bool error = 0;

std::cin >> input;

for (int i = 0; i < input.size(); i++) {

if ((isdigit(input[i]) == 0 && input[i] != '.' && input[i] != '-')) {

error = 1;

break;

}

}

if (error == 1) {

printf("Enter value without letters\n");

}

else

{

break;

}

}

return stoi(input);

}

int main()

{

unsigned short int s{}, d{}, f{}, b{}, x;

while (s <= 7) { //ввод приемлемых значений

cout << "Enter decade num lower then 8\n";

s = fill();

if (s <= 7) {

break;

}

}

while (d <= 1) { //ввод приемлемых значений

cout << "Enter decade num lower then 2\n";

d = fill();

if (d <= 1) {

break;

}

}

while (f <= 1) { //ввод приемлемых значений

cout << "Enter decade num lower then 2\n";

f = fill();

if (f <= 1) {

break;

}

}

while (b <= 255) { //ввод приемлемых значений

cout << "Enter decade num lower then 256\n";

b = fill();

if (b <= 255) {

break;

}

}

x = b; //пакуем все данные

x = 0 << 8 | x;

x = f << 9 | x;

x = 0 << 10 | x;

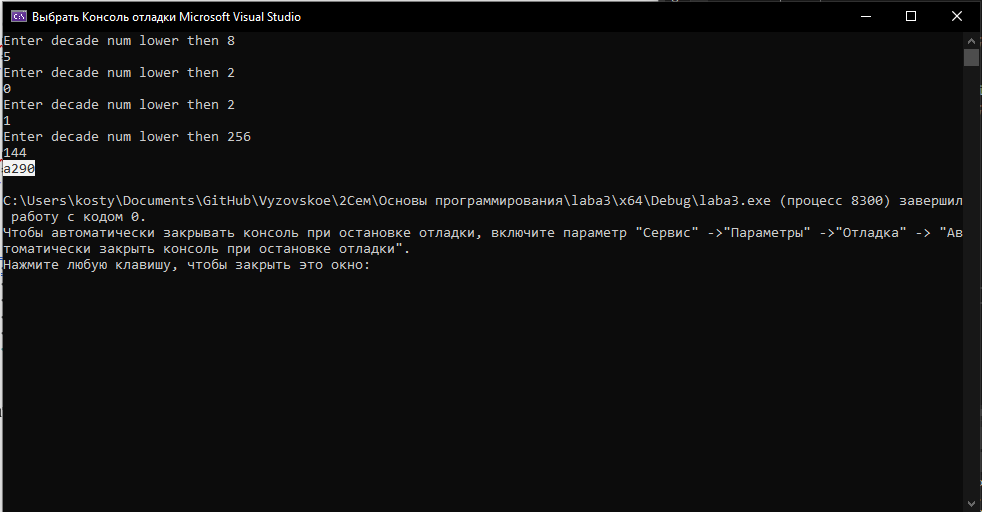
x = d << 11 | x;

x = s << 13 | x;

cout <<hex << x << endl;

}

**Результат работы программы**



**Листинг программы Unpack**

#include <iostream>

#include <string>

int main()

{

unsigned short x;

std::cout << "enter x\n";

std::cin >> std::hex >> x;

short b, d, f, s;

s = (x >> 13) & 0x7;

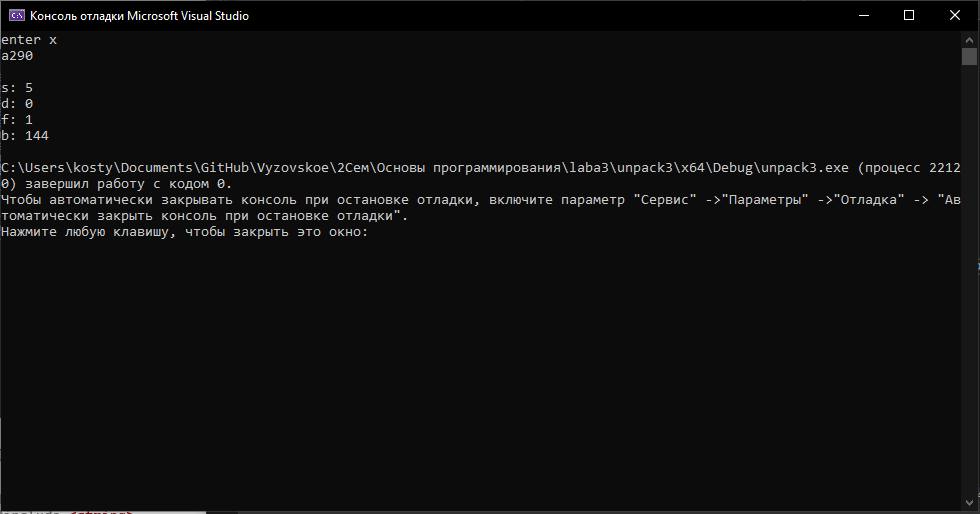
d = (x >> 11) & 0x1;

f = (x >> 9) & 0x1;

b = (x >> 0) & 0xff;

std::cout << std:: dec << "s " << s << " d " << d << " f " << f << " b " << b << std::endl;

}

**Результат раб**

**Вывод**

**Я изучил операции сдвига и смог запаковать 4 переменные в одну ячейку памяти, а потом распаковать ее**

**Достоинства прораммы**

Скорость работы

Проверка значений

Наличие пользовательского интерфейса

**Недостатки программы**

нет