МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3 |
| « Программирование поразрядных операций» |
| по курсу: Основы программирования |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Опарин С.Н. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

1. **Цель работы:**

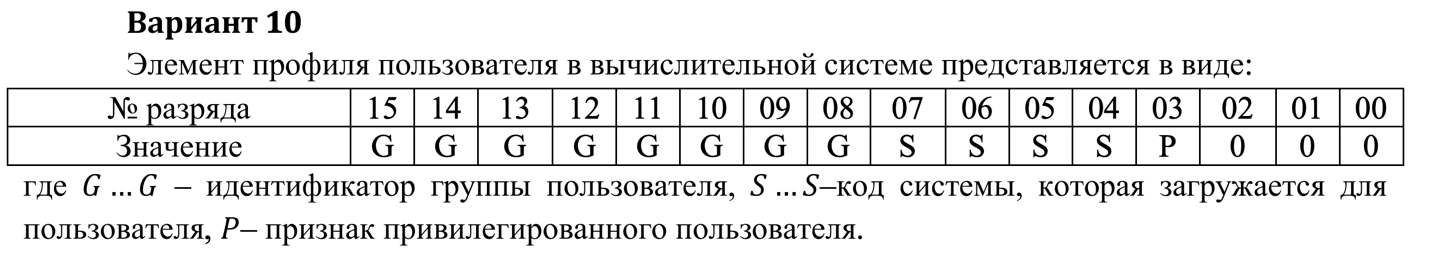
Целью работы является изучение поразрядных операций типа НЕ, И, ИЛИ, исключающее ИЛИ и операций сдвига.

1. **Задание**

Составить две программы, первая из которых вводит составные части структуры данных, приведѐнных 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.



1. **Описание созданных функций**

Для реализации задания нам потребуется следующие функции:

**Имя:** check

**Назначение:** проверка на корректность введенных данных, исходя из варианта

**Входные данные:** количество переменных и

**Выходные данные:** сформированное значение

**Побочный эффект:** 16-ричный формат

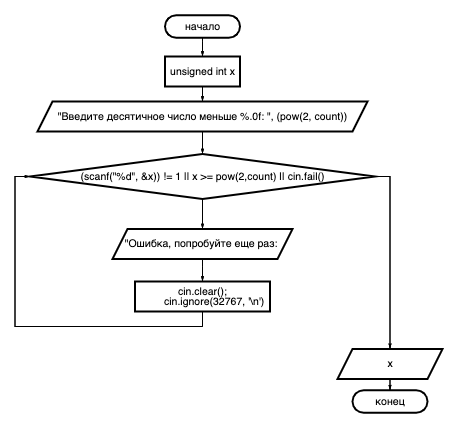
**Прототип:** unsigned short int check\_2(int count)

**Псевдокод:** объявляем переменные  
 выводим сообщение

Пока условие   
 выведи сообщение

очисти

Верни х

**Блок-схема:  
**

**Имя:** main

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

**Входные данные:** 3 ввода с клавиатуры

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

**Побочный эффект:** 16-ричный формат

**Прототип:** int main()

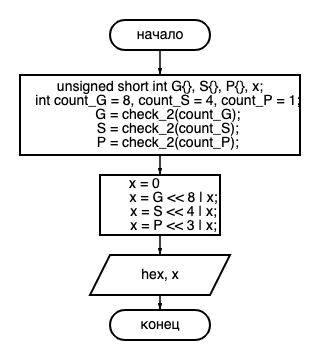
**Псевдокод:** Объявление переменных

Формирование значений  
 Вывод

**Тестовые данные:**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 234 11 1 | eab8 |
| 105 4 1 | 6948 |

**Блок-схема:**

****

**Имя:** main(unpacking)

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

**Входные данные:**

**Input –** значение введенное с клавиатуры

**Выходные данные:**

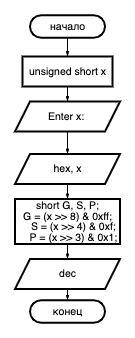
**Input –** 4 переменные, восстановленные из упаковки

**Побочный эффект:** отсутствует

**Прототип:** main()

**Псевдокод:** Объявляем переменные   
 Выводим сообщение «введите х»  
 Вводим Х  
 Формируем обратные значения   
 Выводим значения

**Блок-схема:**

****

1. **Текст программы**

#include <iostream>

#include <string>

#include <cmath>

using namespace std;

unsigned int check(int count){

unsigned int x;

printf ("Введите десятичное число меньше %.0f: ", (pow(2, count)));

while((scanf("%d", &x)) != 1 || x >= pow(2,count) || cin.fail()){

printf("Ошибка, попробуйте еще раз: ");

cin.clear();

cin.ignore(32767, '\n');

}

return x;

}

int main() {

unsigned short int G{}, S{}, P{}, x;

int count\_G = 8, count\_S = 4, count\_P = 1;

G = check(count\_G);

S = check(count\_S);

P = check(count\_P);

x = 0; //пакуем все данные

x = G << 8 | x;

x = S << 4 | x;

x = P << 3 | x;

cout << hex << x << endl;

}

1. **Текст программы 2**

#include <iostream>

#include <string>

using namespace std;

int main()

{

unsigned short x;

cout << "Enter x: ";

cin >> hex >> x;

short G, S, P;

G = (x >> 8) & 0xff;

S = (x >> 4) & 0xf;

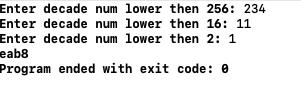
P = (x >> 3) & 0x1;

cout << dec << "G = " << G << endl << "S = " << S << endl << "P = " << P << endl << endl;

}

1. **Пример выполнения программы**

Ниже показан пример выполнения программы



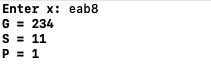


Рис. 1 - Пример выполнения программы

1. **Анализ результатов и выводы**

В результате выполнения лабораторной работы было изучено использование поразрядных операций типа НЕ, И, ИЛИ, исключающее ИЛИ и операций сдвига.

К достоинствам программы можно отнести:

* Высокая скорость работы программы
* Оптимизация

Недостатков нет