

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» им.В.И.УЛЬЯНОВА (ЛЕНИНА)

Кафедра вычислительной техники

Отчет по лабораторной работе № 13
по дисциплине «Программирование»
Тема: «Битовые поля в структурах»

Студент гр. 8307

_____ Овечко Д.А.

Преподаватель

_____ Перязева Ю.В.

Содержание

Задание	2
Постановка задачи и описание решения	2
Описание переменных.....	3
Контрольные примеры	4
Текст программы	5
Пример работы программы	9
Вывод	11

Цель

Получить практические навыки в разработке алгоритма и написании программы на языке Си с использованием битовых полей в структурах, классов, а также битовые операции, потому что эти знания необходимы при написании практических заданий.

Задание

Структура содержит 4 битовых поля типа `unsigned char` по 1 байту каждое. Значения полей структуры задаются шестнадцатеричными кодами. Разработать алгоритм и реализовать функцию циклического сдвига влево для заданного поля на заданное число позиций (если задано число N больше 7, то сдвиг производится на $N\%8$ позиций). Вывести результаты в виде шестнадцатеричных значений полей структуры.

Исходные данные: Ввод с клавиатуры.

Постановка задачи и описание решения

Задаётся структура, состоящая из 4 битовых полей типа `unsigned char`, с помощью функции `enter` производится считывание элементов для этой структуры с клавиатуры.

После происходит вывод введённых чисел и их коды в двоичной системе.

Далее у пользователя уточняется с каким полем будет производиться работа. После выбора, пользователя просят ввести число, которое будет равно числу позиций сдвига (добавлена проверка на случай, если число больше 7).

С помощью функции `shift` происходит сдвиг поля на необходимое число позиций.

Выводятся сами шестнадцатеричные числа и их двоичные коды, включая изменяемое число и его код с произведённым сдвигом.

Описание переменных

Таблица 1. Описание переменных.

Имя переменной	Тип	Назначение
code	struct hex	Структура с битовыми полями
i	int	Счетчик
N	int	Число, определяющее сдвиг
k	Int	Переменная для работы в меню (выбор числа для проведения преобразований)

Таблица 2. Описание функций.

Имя переменной	Тип	Назначение
enter	struct hex	Ввод шестнадцатеричного кода
shift	struct hex	Сдвиг поля на необходимое число полей

Таблица 3. Описание структуры.

Имя переменной	Тип	Назначение
char_ ch[4]	union	Шестнадцатеричное число

Таблица 4. Описание класса.

Имя переменной	Тип	Назначение
a	*char	Введённое шестнадцатеричное число
bit7: 1	unsigned	7 бит
bit6: 1	unsigned	6 бит
bit5: 1	unsigned	5 бит
bit4: 1	unsigned	4 бит
bit3: 1	unsigned	3 бит
bit2: 1	unsigned	2 бит
bit1: 1	unsigned	1 бит
bit0: 1	unsigned	0 бит

Контрольные примеры

1. Пример 1

Исходные данные:

Введите 4 шестнадцатиричных числа

Первое: 12

Второе: 23

Третье: 34

Четвёртое: A4

Выберем для преобразования номер 3

Сдвиг 34

Результат:

Преобразованный список, в котором изменения произведены над 3-им элементом и двоичные коды элементов списка.

2. Пример 2

Исходные данные:

Введите 4 шестнадцатиричных числа

Первое: A3

Второе: A4

Третье: A5

Четвёртое: A6

Выберем для преобразования номер 4

Сдвиг 124

Результат:

Преобразованный список, в котором изменения произведены над 4-ым элементом и двоичные коды элементов списка.

3. Пример 3

Исходные данные:

Введите 4 шестнадцатиричных числа

Первое: B5

Второе: F6

Третье: 15

Четвёртое: D3

Выберем для преобразования номер 1

Сдвиг 876

Результат:

Преобразованный список, в котором изменения произведены над 1-ым элементом и двоичные коды элементов списка.

Текст программы

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
#include <ctype.h>
```

```
//////////////////////////////////////////Структурные
поля////////////////////////////////////////
```

```
union char_ /// класс размером в 8 бита
{
    char bit;
    struct {
        unsigned char bit7: 1;
        unsigned char bit6: 1;
        unsigned char bit5: 1;
        unsigned char bit4: 1;
        unsigned char bit3: 1;
        unsigned char bit2: 1;
        unsigned char bit1: 1;
        unsigned char bit0: 1;
    };
};
```

```
//////////////////////////////////////////
////////////////////////////////////////
```

```
struct hex
{
    union char_ ch[4];
};
```

```
////////////////////////////////////////Описание
функций////////////////////////////////////////
```

```
struct hex enter(); /// Ввод
struct hex shift(struct hex , int, int); /// преобразование
```

```
////////////////////////////////////////main////////////////////////////////////////
////////////////////////////////////////
```

```
int main() {
    SetConsoleCP(1251); // установка кодовой страницы win-ср 1251
    в поток ввода
    SetConsoleOutputCP(1251); // установка кодовой страницы win-
    ср 1251 в поток вывода
    struct hex code;
    code = enter();
    printf("Заданы шестнадцатеричные коды: 1) %3x 2) %3x 3) %3x
    4) %3x\n", code.ch[0].bit,
        code.ch[1].bit, code.ch[2].bit, code.ch[3].bit);
    int N, k;
    printf("Коды в двоичной системе: 1) %d%d%d%d%d%d%d%d 2)
    %d%d%d%d%d%d%d%d"
        " 3) %d%d%d%d%d%d%d%d 4) %d%d%d%d%d%d%d%d\n",
    code.ch[0].bit0, code.ch[0].bit1,
        code.ch[0].bit2, code.ch[0].bit3, code.ch[0].bit4,
    code.ch[0].bit5, code.ch[0].bit6,
        code.ch[0].bit7, code.ch[1].bit0, code.ch[1].bit1,
    code.ch[1].bit2, code.ch[1].bit3,
        code.ch[1].bit4, code.ch[1].bit5, code.ch[1].bit6,
    code.ch[1].bit7, code.ch[2].bit0,
        code.ch[2].bit1, code.ch[2].bit2, code.ch[2].bit3,
    code.ch[2].bit4, code.ch[2].bit5,
        code.ch[2].bit6, code.ch[2].bit7, code.ch[3].bit0,
    code.ch[3].bit1, code.ch[3].bit2,
        code.ch[3].bit3, code.ch[3].bit4, code.ch[3].bit5,
    code.ch[3].bit6, code.ch[3].bit7);
    printf("Введите номер кода, который вы хотите преобразовать
    (или 1, или 2, или 3, или 4): ");
    scanf("%d", &k);
    printf("Введите на какое кол-во знаков произвести сдвиг: ");
```

```

scanf("%d", &N);
N = N % 8;
code = shift(code, N, k);
printf("Преобразованные шестнадцатеричные коды: 1) %3x 2)
%3x 3) %3x 4) %3x\n", code.ch[0].bit,
        code.ch[1].bit, code.ch[2].bit, code.ch[3].bit);
printf("Коды в двоичной системе: 1) %d%d%d%d%d%d%d%d 2)
%d%d%d%d%d%d%d%d"
        " 3) %d%d%d%d%d%d%d%d 4) %d%d%d%d%d%d%d%d\n",
code.ch[0].bit0, code.ch[0].bit1,
        code.ch[0].bit2, code.ch[0].bit3, code.ch[0].bit4,
code.ch[0].bit5, code.ch[0].bit6,
        code.ch[0].bit7, code.ch[1].bit0, code.ch[1].bit1,
code.ch[1].bit2, code.ch[1].bit3,
        code.ch[1].bit4, code.ch[1].bit5, code.ch[1].bit6,
code.ch[1].bit7, code.ch[2].bit0,
        code.ch[2].bit1, code.ch[2].bit2, code.ch[2].bit3,
code.ch[2].bit4, code.ch[2].bit5,
        code.ch[2].bit6, code.ch[2].bit7, code.ch[3].bit0,
code.ch[3].bit1, code.ch[3].bit2,
        code.ch[3].bit3, code.ch[3].bit4, code.ch[3].bit5,
code.ch[3].bit6, code.ch[3].bit7);
return 0;
}

```

```

////////////////////////////////////Функции////////////////////////////////////
////////////////////////////////////

```

```

////////////////////////////////////
////////////////////////////////////

```

```

struct hex enter() /// Ввод
{
    struct hex add;
    printf("Введите 4 шеснадцатеричных числа");
    int i;
    printf("\nПервое: ");
    scanf("%x", &i);
    add.ch[0].bit = i;
}

```



```

printf("Второе: ");
scanf("%x", &i);
add.ch[1].bit = i;
printf("Третье: ");
scanf("%x", &i);
add.ch[2].bit = i;
printf("\nчетвертое: ");
scanf("%x", &i);
add.ch[3].bit = i;
return add;
}

```

```

////////////////////////////////////
////////////////////////////////////

```

```

struct hex shift(struct hex current, int N, int k) ///
Преобразование шеснадцатиричного кода
{
    if (k == 1)
        current.ch[0].bit = current.ch[0].bit << N;
    else if (k == 2)
        current.ch[1].bit = current.ch[1].bit << N;
    else if (k == 3)
        current.ch[2].bit = current.ch[2].bit << N;
    else if (k == 4)
        current.ch[3].bit = current.ch[3].bit << N;
    return current;
}

```

```

////////////////////////////////////
////////////////////////////////////

```

Пример работы программы

1. Исходные данные (Пример 1)

Введите 4 шестнадцатиричных числа

Первое: 12

Второе: 23

Третье: 34

Четвёртое: A4

Выберем для преобразования номер 3

Сдвиг 34

Вывод программы:

```
Введите 4 шеснадцатиричных числа
Первое: 12
Второе: 23
Третье: 34
Четвертое: A4
Заданы шестнадцатеричные коды: 1) 12 2) 23 3) 34 4) fffffffa4
Коды в двоичной системе: 1) 00010010 2) 00100011 3) 00110100 4) 10100100
Введите номер кода, который вы хотите преобразовать (или 1, или 2, или 3, или 4): 3
Введите на какое кол-во знаков произвести сдвиг: 34
Преобразованные шестнадцатеричные коды: 1) 12 2) 23 3) fffffffd0 4) fffffffa4
Коды в двоичной системе: 1) 00010010 2) 00100011 3) 11010000 4) 10100100
```

2. Исходные данные (Пример 2)

Введите 4 шестнадцатиричных числа

Первое: A3

Второе: A4

Третье: A5

Четвёртое: A6

Выберем для преобразования номер 4

Сдвиг 124

Вывод программы:

```
Введите 4 шеснадцатичных числа
Первое: A3
Второе: A4
Третье: A5
Четвертое: A6
Заданы шеснадцатичные коды: 1) fffffffa3 2) fffffffa4 3) fffffffa5 4) fffffffa6
Коды в двоичной системе: 1) 10100011 2) 10100100 3) 10100101 4) 10100110
Введите номер кода, который вы хотите преобразовать (или 1, или 2, или 3, или 4): 4
Введите на какое кол-во знаков произвезти сдвиг: 124
Преобразованные шеснадцатичные коды: 1) fffffffa3 2) fffffffa4 3) fffffffa5 4) 60
Коды в двоичной системе: 1) 10100011 2) 10100100 3) 10100101 4) 01100000
```

3. Исходные данные (Пример 3)

Введите 4 шеснадцатичных числа

Первое: B5

Второе: F6

Третье: 15

Четвёртое: D3

Выберем для преобразования номер 1

Сдвиг 876

Вывод программы:

```
Введите 4 шеснадцатичных числа
Первое: B5
Второе: F6
Третье: 15
Четвертое: D3
Заданы шеснадцатичные коды: 1) fffffffb5 2) fffffffb6 3) 15 4) fffffffd3
Коды в двоичной системе: 1) 10110101 2) 11110110 3) 00010101 4) 11010011
Введите номер кода, который вы хотите преобразовать (или 1, или 2, или 3, или 4): 1
Введите на какое кол-во знаков произвезти сдвиг: 876
Преобразованные шеснадцатичные коды: 1) 50 2) fffffffb6 3) 15 4) fffffffd3
Коды в двоичной системе: 1) 01010000 2) 11110110 3) 00010101 4) 11010011
```

Вывод

При выполнении лабораторной работы были получены практические навыки в работе с битовыми полями в структурах, в классах, а также с битовыми операциями.