

Задание 1

Тема. Поразрядные операции и их применение

Цель. Получить навыки применения поразрядных операций в алгоритмах

Задание. Выполнить упражнения по применению битовых операций по изменению значений битов в ячейке оперативной памяти, созданию маски для изменения значения ячейки. Создание выражения, содержащего поразрядные операции, для выполнения определенной операции над значением ячейки.

В таблице 1 приведены варианты индивидуальных заданий. Номер варианта определяется по формуле – это (остаток от деления номера студента в списке журнала на 19)+1.

В приложении 1 приведены примеры выполнения поразрядных операций.

В каждом варианте пять упражнений, пронумерованных в столбцах таблицы 1.

Требования к выполнению задания

1. Разработать программу, которая продемонстрирует выполнение упражнений варианта. Результаты выполнения упражнения выводить на монитор.

Требования к упражнениям (номер требования определяет номер упражнения в таблице)

- 1) Определить переменную целого типа, присвоить ей значение, используя константу в шестнадцатеричной системе счисления.
Разработать оператор присваивания и его выражение, которое установит заданные в задании биты исходного значения переменной в значение 1, используя соответствующую маску и поразрядную операцию.
- 2) Определить переменную целого типа.
Разработать оператор присваивания и его выражение, которое обнуляет заданные в задании биты исходного значения переменной, используя соответствующую маску и поразрядную операцию. Значение в переменную вводится с клавиатуры.
- 3) Определить переменную целого типа.
Разработать оператор присваивания и выражение, которое умножает значение переменной на число, указанное в третьем столбце варианта, используя соответствующую поразрядную операцию. Изменяемое число вводится с клавиатуры.
- 4) Определить переменную целого типа.

Разработать оператор присваивания и выражение, которое делит значение переменной на число, указанное в четвертом столбце варианта, используя соответствующую поразрядную операцию. Изменяемое число вводится с клавиатуры.

5) Определить переменную целого типа.

Разработать оператор присваивания и выражение, в котором используются только поразрядные операции. В выражении используется маска – переменная. Маска может быть инициализирована единицей в младшем разряде (вар 1) или единицей в старшем разряде (вар 2). Изменяемое число вводится с клавиатуры.

2. Выполнить тестирование программы

3. Оформить отчет

Варианты упражнений

№	1 Номер бита	2 Номер бита	3 множитель	4 делитель	5 Задание для выражения
1	5-ый и 7-ой справа	С 9-ого четыре бита слева	8	8	Установить n-ый бит в 1, используя маску пункта 1
2	Три старших	12-ый, 14-ый, 3-ий	4	4	Установить n-ый бит в 1, используя маску пункта 2
3	Только с четными номерами	7-ой, 9-ый, 11-ый	16	16	Обнулить n-ый бит в 1, используя маску пункта 1
4	Только с не четными номерами	С 5-ого бита четыре слева	32	32	Установить n-ый бит в 1, используя маску пункта 2
5	17-ый, 15-ый, 1- ый	С 5-ого бита три справа	64	64	Обнулить n-ый бит в 1, используя маску пункта 2
6	3-ий, 11-ый, 5- ый	Четыре младших бита	128	128	Установить n-ый бит в 1, используя маску пункта 2
7	Четыре старших бита	9-ый, 11-ый, 3-ий	512	512	Обнулить n-ый бит в 1, используя маску пункта 1
8	1-ый, 6-ой, 9-ый	1-ый, 6-ой, 9-ый	8	8	Установить n-ый бит в 1, используя маску пункта 1
9	0-ый, 11-ый, 3- ий	Четыре старших бита	4	4	Обнулить n-ый бит в 1, используя маску пункта 2
10	Четыре младших бита	3-ий, 11-ый, 5-ый	16	16	Установить n-ый бит в 1, используя маску пункта 2
11	С 5-ого бита четыре слева	5-ый, 7-ой справа	32	32	Обнулить n-ый бит в 1, используя маску пункта 1

12	С 3-ого бита три справа	Три старших	64	64	Установить n-ый бит в 1, используя маску пункта 1
13	7-ой, 9-ый, 11-ый	Только с четными номерами	128	128	Обнулить n-ый бит в 1, используя маску пункта 1
14	12-ый, 14-ый, 3-ий	Только с нечетными номерами	512	512	Установить n-ый бит в 1, используя маску пункта 2
15	С 9-ого бита четыре слева	17-ий, 15-ый, 1-ый	1024	1024	Обнулить n-ый бит в 1, используя маску пункта 1
16	1-ый, 2-ой, 7-ой	С 7-ого три бита слева	8	8	Обнулить n-ый бит в 1, используя маску пункта 1
17	С 3-ого бита четыре слева	12-ый, 11-ый, 1-ый	32	32	Обнулить n-ый бит в 1, используя маску пункта 2
18	3-ий, 5-ый, 8-ой	С 4-ого два бита слева	16	16	Обнулить n-ый бит в 1, используя маску пункта 1
19	1-ый, 5-ый, 6-ой	15-ый, 12-ый, 3-ий	128	128	Обнулить n-ый бит в 1, используя маску пункта 1

Приложение 1. Поразрядные операции

$x \ll n$	Сдвиг влево двоичного кода (умножение на 2^n)	<code>unsigned int x=7; x=x<<2;</code> результат <code>0x0000001C</code>
$x \gg n$	Сдвиг вправо двоичного кода (деление на 2^n)	<code>unsigned int x=28; x=x>>2;</code> результат <code>=0x00000007</code>
$x \& \text{maska}$	Поразрядное И (применяется для записи в указанный разряд 0)	Правило выполнения операции $\begin{array}{r} 111 \\ \& 100 \\ = 100 \end{array}$ Установить в двоичном коде переменной x только 9-ый справа бит в 0 <code>unsigned short int x=0xAEFF;</code> <code>unsigned short int maska=0xFDFF;</code> <code>x=x & maska</code> результат <code>0xACFF</code>

$X \mid \text{maska}$	Поразрядное ИЛИ (применяется для записи в указанный разряд 1)	Правило выполнения операции $\begin{array}{r} 111 \\ 100 \\ 111 \end{array}$ Установить в двоичном коде переменной x 9-ый справа бит в 1 <code>unsigned short int x=0xACFF;</code> <code>unsigned short int maska=0x0200;</code> <code>x=x maska</code> результат <code>0xAEFF</code> ;
$X \wedge \text{maska}$	Исключающее ИЛИ для поразрядных операций Используется для проверки соответствующих битов двух переменных, если они имеют разные значения, то результат 1, а если равны, то 0.	Правило выполнения операции $\begin{array}{r} 1111 \\ \wedge 0001 \\ = 1110 \end{array}$ <code>unsigned int x=0xF, a=1;</code> <code>a=x^a;</code> Результат: в переменной a значение <code>0x0000000E</code>
\sim	Инверсия (0 заменяет на 1, а 1 на 0)	<code>x=0x0F;</code> <code>~x;</code> результат <code>0xF0</code>

Пример реализации алгоритма вывода двоичного кода заданного значения

```

void coutp(unsigned int x)
{
    int n=sizeof(int)*8;
    unsigned maska=(1<<(n-1));

    for(int i=1; i<=n;i++)
    {
        cout<<((x&maska)>>(n-i));
        maska=maska>>1;
    }
}

```