

Практическая работа №1

Тема. Поразрядные операции и их применение.

Цель. Получение навыков применения поразрядных операций в алгоритмах.

Задание.

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

Требования к упражнениям:

- 1) Определить переменную целого типа, присвоить ей значение, используя константу в шестнадцатеричной системе счисления. Разработать функцию, которая установит заданные в задании биты исходного значения переменной в значение 1, используя соответствующую маску и поразрядную операцию.
 - 2) Разработать функцию, которая обнуляет заданные в задании биты исходного значения целочисленной переменной, введенной пользователем, используя соответствующую маску и поразрядную операцию.
 - 3) Разработать функцию, которая умножает значение целочисленной переменной, введенной пользователем, на множитель, используя соответствующую поразрядную операцию.
 - 4) Разработать функцию, которая делит значение целочисленной переменной, введенной пользователем, на делитель, используя соответствующую поразрядную операцию.
 - 5) Разработать функцию, реализующую задание, в которой используются только поразрядные операции. В выражении используется маска – переменная. Маска инициализируется единицей в младшем разряде (маска 1) или единицей в старшем разряде (маска 2). Изменяемое число и n вводится с клавиатуры.
2. Провести тестирование программы на небольших объемах данных, введенных вручную. Разработанные тесты должны покрывать все случаи входных данных (средний, лучший, худший). Результаты тестирования свести в сводные таблицы.
 3. Составить отчет, отобразив в нем описание выполнения всех этапов разработки, тестирования и код всей программы со скриншотами результатов тестирования.

Таблица 1. Варианты заданий

Вариант	Упр. 1	Упр. 2	Упр. 3	Упр. 4	Упр. 5
1	5-ый и 7-ой справа	С 9-ого четыре слева	8	8	Установить n-ый бит в 1, используя маску 1
2	Три старших	12-ый, 14-ый, 3-ий	4	4	Установить n-ый бит в 1, используя маску 2
3	Все четные	7-ой, 9-ый, 11-ый	16	16	Обнулить n-ый бит, используя маску 1
4	Все нечетные	С 5-ого четыре слева	32	32	Установить n-ый бит в 1, используя маску 2
5	17-ый, 15-ый, 1-ый	С 5-ого три справа	64	64	Обнулить n-ый бит, используя маску 2
6	3-ий, 11-ый, 5-ый	Четыре младших	128	128	Установить n-ый бит в 1, используя маску 2
7	Четыре старших	9-ый, 11-ый, 3-ий	512	512	Обнулить n-ый бит, используя маску 1
8	1-ый, 6-ой, 9-ый	1-ый, 6-ой, 9-ый	8	8	Установить n-ый бит в 1, используя маску 1
9	0-ый, 11-ый, 3-ий	Четыре старших	4	4	Обнулить n-ый бит, используя маску 2
10	Четыре младших	3-ий, 11-ый, 5-ый	16	16	Установить n-ый бит в 1, используя маску 2
11	С 5-ого четыре слева	5-ый, 7-ой справа	32	32	Обнулить n-ый бит, используя маску 1
12	С 3-ого три справа	Три старших	64	64	Установить n-ый бит в 1, используя маску 1
13	7-ой, 9-ый, 11-ый	Все четные	128	128	Обнулить n-ый бит, используя маску 1
14	12-ый, 14-ый, 3-ий	Все нечетные	512	512	Установить n-ый бит в 1, используя маску 2
15	С 9-ого четыре слева	17-ий, 15-ый, 1-ый	1024	1024	Обнулить n-ый бит, используя маску 1
16	1-ый, 2-ой, 7-ой	С 7-ого три слева	8	8	Обнулить n-ый бит, используя маску 1
17	С 3-ого четыре слева	12-ый, 11-ый, 1-ый	32	32	Обнулить n-ый бит, используя маску 2
18	3-ий, 5-ый, 8-ой	С 4-ого два слева	16	16	Обнулить n-ый бит, используя маску 1
19	1-ый, 5-ый, 6-ой	15-ый, 12-ый, 3-ий	128	128	Обнулить n-ый бит, используя маску 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 $\begin{array}{r} 111 \\ \mid 100 \\ 111 \end{array}$ 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 \mid 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;
    }
}
```