

Лабораторна робота №1 (2017).

Тема: Вивчення побітових операцій мови C/C++.

Побітова (бітова) операція в програмуванні – це операція над ланцюжками бітів. У програмуванні, як правило, розглядаються лише деякі види цих операцій: логічні побітові операції та бітові зсуви. Побітові операції застосовуються для типів (з цілочисельним значенням): char, short int, int, long int, long long, а також їх signed і unsigned модифікації.

Важливість вміння працювати з побітовими логічними операціями та операціями зсувів продиктовано такими напрямками сучасних комп'ютерних технологій: комп'ютерна графіка, криптографія, комп'ютерні мережі, для множення / ділення числа відповідного під формулу 2^n та ін. Теоретичною основою логічних операцій є булева алгебра. Основними операціями булевої алгебри є: NOT – заперечення (інверсія), AND – кон'юнкція (логічне множення, і), OR – диз'юнкція (логічне додавання, або) та XOR – виняткове або (додавання за модулем 2). Значення вказаних операцій булевої алгебри наведені в таблиці 1.

Таблиця 1.

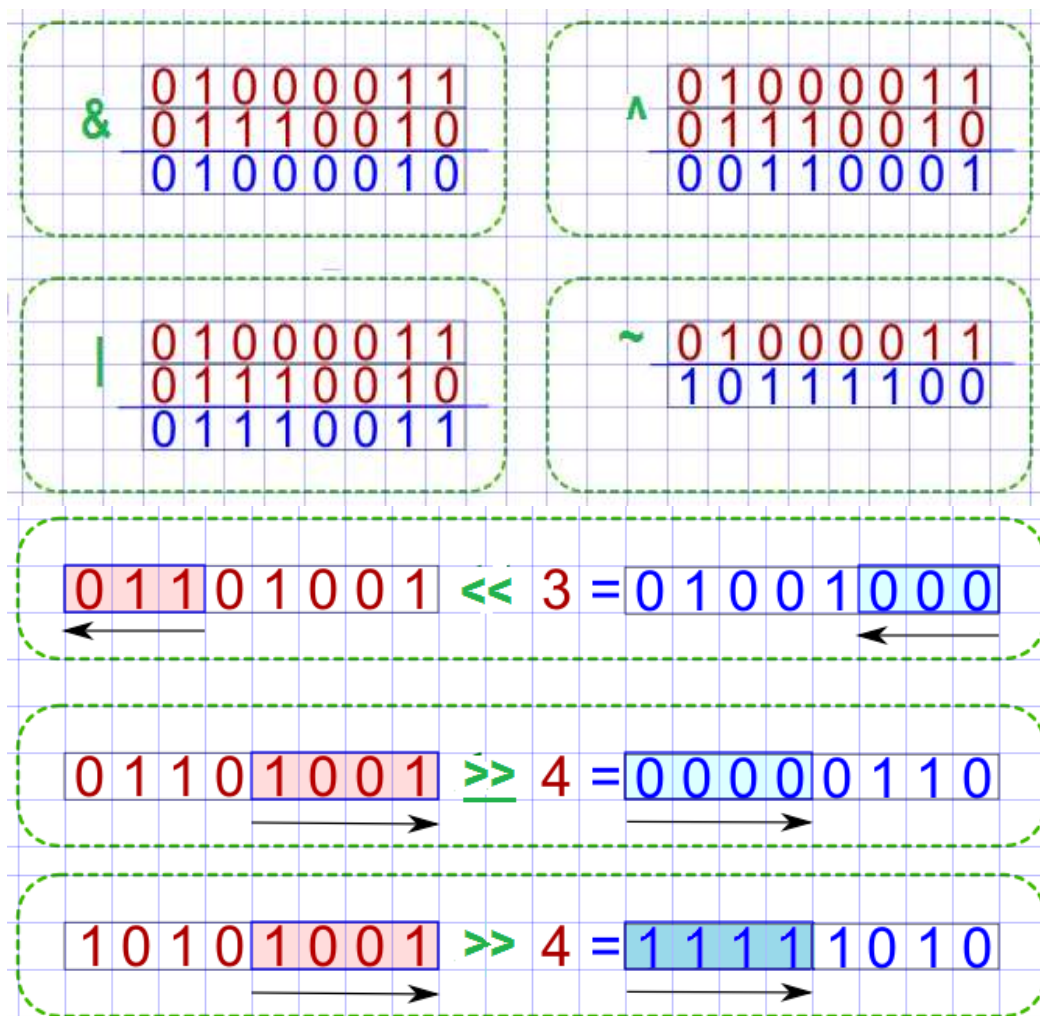
& (і, кон'юнкція, AND) : $a \& b$			(або, диз'юнкція, OR): $a b$		
a \ b	false(0)	true(1)	a \ b	false(0)	true(1)
false(0)	false(0)	false(0)	false(0)	false(0)	true(1)
true(1)	false(0)	true(1)	true(1)	true(1)	true(1)
^ (виняткове або, XOR) : $a \wedge b$			~ (заперечення, інверсія, NOT)		
a \ b	false(0)	true(1)	a	~ a	
false(0)	false(0)	true(1)	false(0)	true(1)	
true(1)	true(1)	false(0)	true(1)	false(0)	

До бітових операцій також відносять бітові зсуви. При зсуві значення бітів копіюються в сусідні за напрямом зсуву. Розрізняють декілька видів зсувів – *логічний*, *арифметичний* та *циклічний*, залежно від обробки крайніх бітів. Циклічного зсуву в мові C/C++ – немає (тільки в мові Асемблер). Також розрізняють зсув *вліво* (<<, в напрямку від молодшого біта до старшого) і *вправо* (>>, в напрямку від старшого біта до молодшого). При логічному зсуві (значення вважається беззнаковим числом, тип даних описується з модифікатором unsigned) значення останнього біта за напрямом зсуву втрачається (копіюємо в біт перенесення), а перший набуває нульове значення. Арифметичний зсув аналогічний логічному, але значення слова вважається знаковим числом, представленим в додатковому коді. Так, при правому зсуві старший біт зберігає своє значення. Лівий арифметичний зсув ідентичний логічному.

Побітові операції на мові C/C++:

Операція	Синтаксис	Опис
&	d=x&y;	Побітове & між x та y та результат присвоїти d.
	d=x y;	Побітове або між x та y та результат присвоїти d.
^	d=x^y;	Побітове виключне або між x та y та результат присвоїти d.
~	d=~x;	Побітове заперечення
<<	d=x<<n;	Зсув x вліво на n розрядів та результат присвоїти d.
>>	d=x>>n;	Зсув x вправо на n розрядів та результат присвоїти d.

Приклад застосування даних операцій для довільних чисел від 0 до 255, які розташовані в одному байті: $67_{10} = 0100\ 0011_2$, $114_{10} = 0111\ 0010_2$ та $105_{10} = 0110\ 1001_2$ (unsigned char)
 $-87_{10} = 1010\ 1001_2$ (char, signed char)



Приклади.

Завдання 1. Розглянемо програму перекладу чисел з десяткової системи у двійкову:

```
#include <iostream>
using namespace std;
void main(void)
{
    char out[255];
    int i, in, ix, n=0;
    double x;

    cout<<" Введіть ціле додатне число:";
    cin>>in;
    if( in!=0 ){
        x=in;
        do{
            x = x/2.;
            ix=x;
            if((x-ix)!=0) out[n]='1';
            else out[n]='0';
            n++;
            x=ix;
        } while(x>=1);
    }
    cout<<" Результат:";
    for( i=n-1; i>=0; i--)
```

Об'єкто – зорієнтоване програмування мовою C++

```
        cout<<out[i];  
    }  
}
```

Завдання 2. Використовуючи тільки побітові операції обчислити вираз: $x=8*h$; $y=x/4$; Де h – це данні, які отримано на вході. Вхідні данні та результати обчислення роздрукувати.

Побітовий зсув на один розряд вліво відповідає множенню на 2. Відповідно на один розряд вправо діленню на 2. Програма буде мати вигляд:

```
#include <iostream.h>  
void main(void)  
{  
    cout<<"Enter h";  
    int h,x,y;  
    cin>>h; x=h<<3; y=x>>1;  
    cout<<" h="<<h<<" x="<<x<<" y="<<y<<endl;  
}
```

Завдання 3. Використовуючи тільки побітові операції обчислити вираз: $x=33*a + (a * 16 - b*17)/8 + (15*b + 300)/128$. Де a, b – це данні, які отримано на вході. Вхідні данні та результати обчислення роздрукувати.

```
(x = a + a * 32 + ( a *16 - b - b*16)/ 8 + (b*16 - b+300)/(128)  
    = a + a * 25 + ( a * 24 - b - b * 24)/ 23 + (b*24 -b +300)/( 27)  
    = a + a <<5 + ( a <<4 - b - b <<4) >>3 + (b<<4-b +300)>>7).  
#include <iostream.h>  
void main(void)  
{ int x,a,b;  
    cout<<"Enter a, b";  
    cin>>a>>b; x= a + a <<5 + ( a <<4 - b - b <<4) >>3 + (b<<4-b +300)>>7;  
    cout<<" x="<<x<<" a="<<a<<" b="<<b<<endl;  
}
```

Завдання 4. Нехай отримано повідомлення - h , яке записано у змінну int . Якщо біт 1 дорівнює 1, а біт 3 дорівнює 0, то необхідно розрахувати вираз $x=4*h$, інакше $x=h/2$, Де h – це повідомлення.

Для визначення величини i -го розряду у двійковій системі числення, необхідно сформулювати відповідну маску (можна використати операцію зсуву вліво), далі виконати побітну операцію $\&$, та для зручності аналізу, зсунути результат вправо на відповідну кількість розрядів. Наприклад. Нехай є повідомлення: $h=00001010$. Необхідно визначити величину третього розряду (зліва). Необхідно використати маску $mb=00000100$ ($1<<2$). Далі застосовуємо побітве $\&$ і отримуємо:

$h=00001010$
$mb=00000100$
$h\&mb=00000000$

Програма для цього завдання буде мати вигляд:

```
#include <conio.h>  
#include <iostream.h>  
  
void main(void)  
{  
    clrscr();  
    cout<<"Enter h";  
    int h,x,mb1=1, mb3=1<<2;  
    cin>>h;  
    if( (h&mb1)==1 && ((h&mb3)>>2)==0 ) x=h<<2;  
    else x=h>>1;  
    cout<<" h="<<h<<" x="<<x<<endl;  
}
```

В обчисленнях з цілими числами, де використовується множення або ділення кратне 2, можна використовувати операції зсуву. Це значно прискорює виконання програми.

Завдання 5. Задано текст ASCII (текстовий рядок), який складається з 64 букв. Написати функцію шифрування заданого тексту. Кожна при шифруванні буква тексту буде записана у елемент, який складається з двох байтів та має структуру:

- у бітах 0-3 старша частина, тобто біти 4-7 ASCII - коду букви (4 біти),
- у бітах 4-10 позиція букви у рядку (7 біти),
- у бітах 12-15 молодша частина, тобто біти 0-3 ASCII - коду букви (4 біти),
- 11 біт – біт парності отриманого запису (1 біт).

```
#include <iostream.h>
void Shifruvanna(char S[64], unsigned short Rez[64])
{
    unsigned char c;
    unsigned short r, t, i, b;
    short j;
    cin.get(S, 64);
    for (i = 0; i < 64; i++)          //
    {
        r = 0;                       // 0000 0000 0000 0000
        c = S[i];                     // s - 0x73 = 0111 0011
        t = c;
        r |= t >> 4;                   // 0000 0000 0000 0111
        r |= i << 4;                   // 0000 0000 0101 0111 if i=5 0000 0000 0000 0101
        t = c;
        r |= t << 12;                  // 0011 0000 0101 0111 if i=5 0000 0000 0000 0101
        t = 1;
        b = 0;
        for (j = 0; j < 16; j++)       // обчислення біта парності
        {
            if (r & t) {
                if (b == 0) b = 1; else b = 0;
            }
            t <<= 1;
        }
        r |= b << 11;                  // 0011 0000 0101 0111 if i=5 0000 0000 0000 0101
        Rez[i] = r;
    }
}

int main()
{
    char S[64];
    unsigned short Rez[64];
    unsigned short i;
    cin.get(S, 64);
    Shifruvanna(S, Rez);

    for (i = 0; i < 64; i++)
        cout << hex << Rez[i] << endl;

    return 0;
}
```

}Завдання до лабораторної роботи.

Завдання 1.

1. Вхідний рядок байтів має довжину, кратну 8, написати програму що кодує його за наступною схемою: вісім біт першого байта записуються, як нулеві біти перших восьми байтів, вісім біт другого – перші біти других восьми байтів і т.д. для кожних восьми байтів вхідної послідовності.
2. Цифровим каналом передаються *слова* (пари байтів); два старших біта кожного з них містять контрольну суму (CRC), що формується за таким правилом: 14-й біт – сума за

- модулем 2 непарних бітів, 15-й біт – сума за модулем 2 парних бітів. Перевірити, чи не відбулось втрати інформації при передачі вказаної користувачем послідовності слів.
3. Задано числа $0 \leq a \leq 12, 0 \leq b \leq 9, 1940 \leq c \leq 2065, 0 \leq d \leq 31, 0 \leq e \leq 1$. Компактно розмістити задані числа у змінній розміром у 3 байти.
 4. Зашифрувати текст, помінявши у бітовому представленні кожної букви 5-й та 6-й біти.
 5. Зашифрувати текст, помінявши у бітовому представленні кожного символу старші та молодші 4 біти.
 6. Зашифрувати рядок символів циклічним зсувом вліво його бітів на вказану величину.
 7. Користувачем вказуються цілі без знакові числа a та b . Обчислити добуток цих чисел, використовуючи лише оператори побітового зсуву та додавання.
 8. Обчислити значення вказаного виразу, не використовуючи операторів множення та ділення:
$$\frac{x+320 \cdot y}{128} - x \cdot 120$$
, де x, y – цілі числа, вказані користувачем.
 9. Бітовий образ 8×8 задається вісьмома байтами, кожен з яких задає одну сканлінію з 8 точок (0-7 біти): нульовий біт – біла точка, одиничний – чорна. За вказаними користувачем вісьмома байтами вивести на екран (у текстовому режимі) відповідне йому зображення. Для відображення білих точок використати символ “пробіл”, для відображення чорних – символ “O”.
 10. Цифровим каналом передається набір із 8 дійсних чисел, а також CRC-байт, кожний біти якого відображають знаки відповідних чисел вхідного набору: $b_i = \begin{cases} 0, & a[i] < 0 \\ 1, & a[i] \geq 0 \end{cases}, i = \overline{0,7}$, де b_i – номер біту CRC, $a[i]$ – i -й елемент послідовності. За введеними числами та CRC байтами, перевірити, чи правильно передана вхідна послідовність.
 11. Закодувати вхідну послідовність байтів $a[i], i = \overline{0, n-1}$, за правилом:
 $a[i] = a[i] \oplus (255 - (i \bmod 256)), i = \overline{0, n-1}$.
 12. Закодувати вхідну послідовність, з використанням XOR-маски, яка вибирається у залежності від її розмірності: у випадку парного розміру використати маску “01010101”, у випадку непарного – “10101010”.

Завдання 2.

1. Задано масив з 128 байтів шифрованого текстового документу. Кожні два байта, представляють собою елемент, який має таку структуру:
 - у бітах 0-2 знаходиться номер рядка букви (3 біти),
 - у бітах 3-5 позиція букви у рядку (3 біти),
 - 6 біт – біт парності перших двох полів (1 біт)
 - у бітах 7-14 ASCII - код букви (8 біт),
 - 15 біт - біт парності попереднього поля (1 біт).Написати функцію перевірки правильності зашифрованої інформації. Інформація буде вважатися правильною, якщо сума бітів вказаних полів та біта парності буде парною.
2. Задано масив з 128 байтів шифрованого текстового документу. Кожні два байта, представляють собою елемент, який має таку структуру:
 - у бітах 0-2 знаходиться номер рядка букви (3 біти),
 - у бітах 3-5 позиція букви у рядку (3 біти),
 - 6 біт – біт парності перших двох полів (1 біт)
 - у бітах 7-14 ASCII - код букви (8 біт),
 - 15 біт - біт парності попереднього поля (1 біт).Написати функцію розшифровки тексту.
3. Задано текст, який складається з 8 рядків по 8 букв. Кожна при шифруванні буква тексту буде записана у елемент, який складається з двох байтів та має структуру: у бітах 0-2 знаходиться номер рядка букви (3 біти),
 - у бітах 3-5 позиція букви у рядку (3 біти),

- 6 біт – біт парності перших двох полів (1 біт)
- у бітах 7-14 ASCII - код букви (8 біт),
- 15 біт - біт парності попереднього поля (1 біт).

Написати функцію шифрування тексту .

4. Задано масив з 128 байтів зашифрованого текстового документу. Кожні два байта, представляють собою елемент, який має таку структуру :

- у бітах 0-1 знаходиться номер рядка букви (2 біти),
- у бітах 2-5 позиція букви у рядку (4 біти),
- у бітах 6-13 ASCII - код букви (8 біт),
- 14 біт – біт парності перших двох полів (1 біт)
- 15 біт - біт парності коду букв (1 біт).

Написати функцію перевірки правильності зашифрованої інформації. Інформація буде вважатися правильною, якщо сума бітів вказаних полів та біта парності буде парною.

5. Задано масив з 128 байтів зашифрованого текстового документу. Кожні два байта, представляють собою елемент, який має таку структуру :

- у бітах 0-1 знаходиться номер рядка букви (2 біти),
- у бітах 2-5 позиція букви у рядку (4 біти),
- у бітах 6-13 ASCII - код букви (8 біт),
- 14 біт – біт парності перших двох полів (1 біт)
- 15 біт - біт парності коду букв (1 біт).

Написати функцію розшифровки тексту.

6. Задано текст, який складається з 4 рядків по 16 букв. Кожна при шифруванні буква тексту буде записана у елемент, який складається з двох байтів та має структуру: у бітах 0-1 знаходиться номер рядка букви (2 біти),

- у бітах 2-5 позиція букви у рядку (4 біти),
- у бітах 6-13 ASCII - код букви (8 біт),
- 14 біт – біт парності перших двох полів (1 біт)
- 15 біт - біт парності коду букв (1 біт).

Написати функцію шифрування тексту .

7. Задано масив з 256 байтів шифрованого текстового рядка. Кожні два байта, представляють собою елемент, який має таку структуру :

- у бітах 0-3 старша частина ASCII - коду букви (4 біти),
- у бітах 4-10 позиція букви у рядку (3 біти),
- у бітах 11-14 молодша частина ASCII - коду букви (4 біти),
- 15 біт – біт парності двох букви полів (1 біт).

Написати функцію перевірки правильності зашифрованої інформації. Інформація буде вважатися правильною, якщо сума бітів вказаних полів та біта парності буде парною.

8. Задано масив з 256 байтів шифрованого текстового рядка. Кожні два байта, представляють собою елемент, який має таку структуру :

- у бітах 0-3 старша частина ASCII - коду букви (4 біти),
- у бітах 4-10 позиція букви у рядку (3 біти),
- у бітах 11-14 молодша частина ASCII - коду букви (4 біти),
- 15 біт – біт парності двох букви полів (1 біт).

Написати функцію розшифрування інформації.

9. Задано текст рядок, який складається з 128 букв. Кожна при шифруванні буква тексту буде записана у елемент, який складається з двох байтів та має структуру: у бітах 0-3 старша частина ASCII - коду букви (4 біти),

Об'єкт – зорієнтоване програмування мовою C++

у бітах 4-10 позиція букви у рядку (3 біти),
у бітах 11-14 молодша частина ASCII - коду букви (4 біти),
15 біт – біт парності двох букви полів (1 біт).

Написати функцію шифрування тексту .