

Информатика

Кочанов Марк

10 ноября 2018 г.

МИФИ

Побитовые операции

Унарные и бинарные побитовые операции

```
1 char a = 14; // 0000 1110
2 char b = 7;  // 0000 0111
3
4 a & b        // 0000 0110 AND
5 a | b        // 0000 1111 OR
6 a ^ b        // 0000 1001 XOR
7 ~a           // 1111 0001 NOT
8
9 a << 2        // 0011 1000 LEFT SHIFT
10 b >> 1       // 0000 0011 RIGHT SHIFT
```

- Для типов данных, занимающих больше одного байта, операции действуют аналогично
- Побитовый правый сдвиг эквивалентен умножению на степень двойки

Взятие бита из числа

Взятие 2-го бита (нумерация с нуля, с младшего бита)

```
1  int a = 101          // 0110 0101
2  int mask = 1 << 2    // 0000 0100
3  char b = (a & mask) >> 2 // 1
4  //      0110 0101
5  // & 0000 0100
6  // = 0000 0100
7  // >> 2
8  // = 0000 0001
```

Использование битовых масок

```
1 char u1_read = 1 << 0; // 0000 0001
2 char u1_write = 1 << 1; // 0000 0010
3 char u2_read = 1 << 2; // 0000 0100
4 char u2_write = 1 << 3; // 0000 1000
5 char u3_read = 1 << 4; // 0001 0000
6 char u3_write = 1 << 5; // 0010 0000
7 char u4_read = 1 << 6; // 0100 0000
8 char u4_write = 1 << 7; // 1000 0000
9
10 char file_permission = u1_read | u2_read | u3_write |
    ↪ u4_write; // 1010 0101
11
12 char is_u1_read = file_permission ^ u1_read // != 0, true
13 char is_u3_read = file_permission ^ u3_read // 0, false
```

Задача

БИТОВЫЙ МАССИВ

Реализовать код для создания битового массива на основе массива типа `char`. Для хранения массива битов использовать структуру, внутри которой хранится указатель на массив и длина этого массива. Предполагается, что длина массива кратна 8. Таким образом для хранения 56-битового массива потребуется выделения памяти под хранения массива типа `char*` длины 7.

Необходимо объявить и реализовать функции для создания битового массива, его удаления (очистка динамически выделенной памяти), задания значения k-го бита (нумерация с нуля, с младших разрядов), получения значения k-го бита и вывода в консоль всех элементов битового массива.

По желанию реализовать проверку на корректность аргументов функции (например, выводить ошибку, если пользователь пытается взять 100-ый бит массива, в котором хранится 16 бит).