

# Информатика и программирование



**ПОРАЗЯДНЫЕ (ПОБИТОВЫЕ) ОПЕРАТОРЫ**

# Поразрядные операторы



- Применяются для работы с булевыми векторами.
- **Булев вектор** – набор 0 и 1, который представляет собой набор признаков (каждый разряд – признак). 1 – признак присутствует, 0 – признак отсутствует.
- **Поразрядные операторы** воздействуют на отдельные двоичные разряды (биты) своих операндов. Они определены только для целочисленных операндов, поэтому их нельзя применять к данным типа `bool`, `float` или `double`.

# Поразрядные операторы



**Оператор**

**Значение**

&

Поразрядное И

|

Поразрядное ИЛИ

^

Поразрядное исключающее ИЛИ

<<

Сдвиг влево

>>

Сдвиг вправо

~

Инверсия  
(унарный оператор НЕ)

# Поразрядные операторы И, ИЛИ, исключающее ИЛИ и НЕ



- Поразрядные операторы И, ИЛИ, исключающее ИЛИ и НЕ обозначаются следующим образом:  $\&$ ,  $|$ ,  $\wedge$  и  $\sim$ . Они выполняют те же функции, что и их логические аналоги.
- Но в отличие от логических операторов, поразрядные операторы действуют на уровне отдельных двоичных разрядов.

# Поразрядную операцию И



- Поразрядную операцию **И** можно рассматривать как способ подавления отдельных двоичных разрядов. Это означает, что если какой-нибудь бит в любом из операндов равен 0, то соответствующий бит результата будет сброшен в 0.

# Поразрядный *оператор ИЛИ*



- Поразрядный *оператор ИЛИ* может быть использован для установки отдельных двоичных разрядов. Если в 1 установлен какой-нибудь бит в любом из операндов этого оператора, то в 1 будет установлен и соответствующий бит результата.

# Поразрядный *оператор* **исключающее ИЛИ**



- Поразрядный *оператор* **исключающее ИЛИ** устанавливает двоичный разряд операнда в том и только в том случае, если двоичные разряды сравниваемых операндов оказываются разными, как в приведенном ниже примере.

# Пример



1101 0011	1101 0011	1101 0011	
1001 1001	1001 1001	1001 1001	1101 0011
&		^	~
_____	_____	_____	_____
1001 0001	1101 1011	0100 1010	0010 1100

# Пример программного кода



- // Метод, проверяющий является ли число четным
- void provChet(int x)
- {
- for (int i = 1; i <= x; i++)
- { if ((i & 1) == 0)
- cout<<"Число"<<i<<" - является четным";
- else
- cout<<"Число"<<i<<" - является нечетным"; }
- }

# Еще пример



- // Метод, преобразующий четные числа в нечетные
- // с помощью поразрядного оператора |
- void nechet(int x)
- { int result;
- for (int i = 0; i <= x; i++)
- { result = i | 1;
- printf(“%d “, result);}
- }

# Операторы сдвига



- В С имеется возможность сдвигать двоичные разряды, составляющие целое значение, влево или вправо на заданную величину. Ниже приведена общая форма для этих операторов:

*значение << число\_битов*

*значение >> число\_битов*

- где *число\_битов* — это число двоичных разрядов, на которое сдвигается указанное значение.

# Сдвиг влево



- При сдвиге влево на 1 все двоичные разряды в указываемом значении сдвигаются на одну позицию влево, а младший разряд сбрасывается в нуль.
- Пример.
- $10111011 \ll 5 = 01100000$ .

# Сдвиг вправо беззнакового числа



- При сдвиге вправо на 1 все двоичные разряды в указываемом значении сдвигаются на одну позицию вправо.
- Если вправо сдвигается целое значение без знака, то старший разряд сбрасывается в нуль.
- Пример.
- $10111011 \gg 1 = 01011101$ .

# Сдвиг вправо знакового числа



- Если вправо сдвигается целое значение со знаком, то разряд знака сохраняется.
- Для представления отрицательных чисел старший разряд целого числа устанавливается в 1.
- Если сдвигаемое значение является отрицательным, то при каждом сдвиге вправо старший разряд числа устанавливается в 1.
- Пример.
- $10111011 \gg 1 = 11011101$ .
- Если сдвигаемое значение является положительным, то при каждом сдвиге вправо старший разряд числа сбрасывается в нуль.
- Пример.
- $00111011 \gg 1 = 00011101$ .



- При сдвиге влево и вправо крайние двоичные разряды теряются. Восстановить потерянные при сдвиге двоичные разряды *нельзя*, поскольку сдвиг в данном случае не является циклическим.



- Short int n = 6, result;
- // Умножить на 2
- result = (int)(n << 1);
  
- // Умножить на 4
- result = (int)(n << 2);
  
- // Разделить на 2
- result = (int)(n >> 1);

# Работа с булевыми векторами



- Когда работаем с данными как с булевым вектором, нужно описывать их **беззнаковым типом**.
- `unsigned int x;`

# Примеры



- Установить 1 в  $i$ -м разряде булева вектора  $a$
- $a = a | (1 \ll i)$
- Удалить 0 в  $i$ -м разряде булева вектора  $a$
- $a = \sim(1 \ll i) \& a$
- Проверить – есть ли 1 в  $i$ -ом разряде
- $\text{if } ((1 \ll i) \& a)$  оператор

# Задание



- 1. Создать функцию ввода двоичного вектора.
- 2. Создать функцию вывода двоичного вектора.
- 3. Найти вес двоичного вектора.