

## Задача А. Определить значение бита

```
n >> (k & 1)
```

## Задача В. Инвертировать бит

```
n ^ (1 << k)
```

## Задача С. Установить значение бита в 1

```
n | (1 << k)
```

## Задача D. Установить значение бита в 0

```
n & ~(1 << k)
```

## Задача Е. 2 в степени n

Обратите внимание, что  $2^{31}$  не помещается в `int` в C++

```
1 << n
```

## Задача F. Вывести значение байта побитно

Переведем число в двоичную с помощью `while` и выведем получившийся массив.

Или воспользуемся знаниями из лекции

```
for (int i = 7; i >= 0; i--) {  
    cout << ((a >> i) & 1);  
}
```

## Задача H. Обнулить все биты, кроме последних

Хотим получить число вида  $1\dots 10\dots 0$ , в котором  $n$  нулей на конце.

Возьмем число из всех 1 — это отрицание 0. Теперь подвинем его на  $n$  вправо, затем на  $n$  влево. Получили число, которое хотим.

```
A & ~((~0 >> n) << n)
```

## Задача I. МегаXOR

Заметим, что  $A \oplus (A + 1) = 1$ , если  $A$  — четное.

Еще заметим, что  $A \oplus A = 0$ . И операция XOR коммутативна. Отсюда можно получить,  
 $A \oplus (A + 1) \oplus (A + 2) \oplus \dots \oplus B = (0 \oplus \dots \oplus B) \oplus (0 \oplus \dots \oplus (A - 1))$

Научимся считать  $0 \oplus \dots \oplus B$ . Несложно понять, что значение выражения может принимать 4 значения: 0, 1,  $B$  и  $1 \oplus B$ . Эти значения зависят от остатка  $B$  при делении на 4.