Bit Manipulation

Bitwise Operator

- |
- &
- ^
- ~
- <<(left shift)
- >> (right shift)

Odd & Even Number

```
void bitwise even_odd(int num)
{
    if(num & 1==0)
      {
        cout<<"even"<<endl;
      }
    else
      {
        cout<<"odd"<<endl;
      }
}
```

Swap

```
void bitwise_swap(int &a,int &b)
{
a=a^b;
b=a^b;
a=a^b;
}
```

Bit Masking

Find i’th bit

4th bit of a number

n→ 1 0 0 0 0 1 0 1 0

mask→ 0 0 0 0 0 1 0 0 0 (i << 4)

n&mask→0 0 0 0 0 1 0 0 0

```
bool find_i'th_bit(int n,int p)
{
int mask=1<<p;
mask=n&mask;
mask=mask>>p;
```

```
return mask&1;
}
```

Set i’th bit

set 4th bit of a number

num → 1 0 0 0 0 1 0

mask→ 0 0 0 1 0 0 0

num | mask →1 0 0 1 0 1 0

```
void set_i'th_bit(int n,int p)
{
int mask=1<<p;
n=n| mask;
}
```

Clear i’th bit

clear 4 th bit of num

num → 1 0 0 1 1 0 1 1

mask→ 1 1 1 1 0 1 1 1

num & mask→ 1 0 0 1 0 0 1 1

```
void clear_i'th_bit(int n,int p)
{
  int mask=1<<p
  mask=~mask;
  n=n & mask;
}
```

#Find number of bit to change a to b

a→ 1 0 1 1 0

b→ 1 1 0 1 1

---

a^b→0 1 1 0 1 →13

count=1

13→ 1 1 0 1

12→ 1 1 0 0

(13&12)→1 1 0 0 →12

count=2

12→ 1 1 0 0

11→ 1 0 1 1

12&11→ 1 0 0 0 →8

count =3

8→ 1 0 0 0

7→ 0 1 1 1

8&7→ 0 0 0 0

```
int different_bit(int x,iny y)
{
x=x^y;
int count=0;
while(x)
{
x=x&(x-1);
count++;
}
return count;
}
```

#Find the only non repeating element in an array where every element repeat twice?

💡 x=0^x

💡 0=x^x;

a[ ]→ [1 , 2 , 5 , 1, 2]

x=1 ^ 2 ^ 5 ^ 1 ^ 2 =5

```
int non_repeating_element(int a[],int n)
{
x=0;
for(i=0;i<n;i++)
{
x=x^a[i];
}
return x;
}
```

 # Find two non repeating element in an array where every element repeat twice?

💡 x=x^0

💡 0=x^x

a[ ]→ [1 , 2 , 3 , 2, 5 , 1]

x= 1 ^ 2 ^ 3 ^ 2 ^ 5 ^ 1;

x=3 ^ 5 = 6 = 1 1 0

2nd bit 1 ——>2, 3 , 2

2nd bit 0 ——-> 1 ,5 , 1

2^3^2—→3

x^3—> 5

```
void non_repeating_two_element(int a[],int n)
{
int x=0;
for(int i=0;i<n;i++)
{
  x=x^a[i];
}
int p=0;
while(!find_i'th_bit(x,p))
{
  p++;
}
int one_element=0;
for(int i=0;i<n;i++)
{
  one_element=one_element^a[i];
}
int another_element=x^one_element;

cout<<one_element<<" "<<another_element<<endl;
}
```

#Find the only non repeating element in an array where every element repeat k time?

a[ ]—> [2 , 2 , 1 , 5 , 1 , 1, 2]

k=3

number of bit for every bit position of array

rest of bit which is zero for this example	0	0	0	0	0	0	1

perform % k from every bit position

rest of bit which is zero for this example	0	0	0	0	0	0	1

ans=1 * 4 + 0 * 2 + 1 * 1

```
int  non_repeating_element(int a[],int n, int k)
{
```

```
  vector<int> b(64);
   for(int i=0;i<64;i++)
     {
       for(int j=0;j<n;j++)
          {
              b[i]=b[i]+find_i'th_bit(a[j],i);
          }
    }
int ans=0;
for(i=0;i<64;i++)
    {
        if(b[i]%k)
        ans=ans+pow(2,i);
    }
return ans;
}
```