getOddOccurrence

int getOddOccurrence(int arr[], int n) {

// code here

int res=0;

int i=0;

while(i<n)

res=res^arr[i++];

return res;

}

};

**Find the Missing Number**

int MissingNumber(vector<int>& array, int n) {

// Your code goes here

int x1=array[0];

int x2=1;

for(int i=1;i<n;i++)

{

x1=x1^array[i];

}

for(int i=2;i<=n+1;i++)

{

x2=x2^i;

}

return (x1^x2);

}

Non Repeating Numbers

ic:

vector<int> singleNumber(vector<int> nums)

{

// Code here.

int res=nums[0];

int i=0;

int n=nums.size();

while(i<n)// first we have to find xor of all values

res=res^nums[i++];

//now mask .we do mask because it is usefull to divide the inpujt numbers into two groups

//1)kth bit set 2)kth bit unset

int m;

m=res&~(res-1);

int a=0,b=0;

while(i<n)

{

if(m&nums[i]==0)

{

a=a^nums[i];

}

else

{

b=b^nums[i];

}

i++;

}

Next Permutation

void nextPermutation(vector<int>& nums)

{

next\_permutation(nums.begin(),nums.end());

}