**Left shifts**

 int x = 5;

   cout << (x << 3) << "\n";

**bits on off**

int x = 2;

   if(x & (1 << 2)){

      cout << "bits trun on"<< "\n";

   }

   else{

      cout << "bits trun off\n";

   }

long long w = 1e13 + 9;

   //common mistake if (1LL) dosent give

   if(w & (1LL >> 40)){

      cout << "on\n";

   }

   else{

      cout << "off\n";

   }

**On k position bit**

 int n = 11;

   cout << (n | (1 << 2)) << "\n";

**off k position bit**

**(2^k – 1) =1111**

int n = 6;

   cout << (n & ((1 << 2) -1)) << "\n";

**bitmask**

**[4, 1, 3] array dewa ache**

**Ami each subset er sum ber korte chacchi so**

**[4], [1], [3], [4,1],[4,3],[1,3],[4,1,3]**

**Binary number e lekte pari**

**0 0 0**

**0 0 1**

**0 1 0**

**0 1 1**

**1 0 0**

**1 0 1**

**1 1 0**

**1 1 1**

**[0 1 1] = [1 , 3];**

**First k ta bit on kore baki sob off kore dibo**

**1111 first 2 ta bit on rakbo**

**0011**

**00 11**

**N & (2^k – 1)**

**N % 2^k same**

   int n = 15;

   cout << (n & (1 << 2) - 1) << "\n";

   cout << n % (1 << 2) << "\n";

**0 xor 1 xor 2 xor 3 = 0 khatay prove ache**

**Building function**

**Koyta one ache ta bole dibe**

**15 = 1111 answer = 4**

 int n = 15;

   cout << \_\_builtin\_popcount(n) << "\n";

**time complexity big(0) = 1**

**count leading zero**

 int n = 15;

   cout << \_\_builtin\_clz(n) << "\n";

**count tralling zero**

int n = 8;

   cout << \_\_builtin\_ctz(n) << "\n";

**n = 16 2 er power**

**n = 1**

**n = 2**

**n = 4**

**n = 8**

**n = 16**

**if \_\_builtin\_popcount(n) == 1**

**yes it is power of 2**

**00001000**

**00000001**

**000010000**

**N = 30 tomake 30 ta set dewa ache tomake bolte hobe maximum koyta number thakbe ja pair e thakbe**

**[1 , 2 , 3, 4]**

**[1, 3, 5, 7]**

**[1, 2 ,4 ,9]**

**Total pair possible 3C2 = 3**

#include<bits/stdc++.h>

using namespace std;

#define faster ios\_base::sync\_with\_stdio(0),cin.tie(0),cout.tie(0);

const int N = 101;

set<int>se[1010];

int mask[1010];

int a[N],pref[N];

 int32\_t main(){

   faster;

   int n = 2;

   se[0] = set<int>({1, 2, 3,6});

   se[1] = set<int>({1, 3, 5});

   for(int i = 0; i < n; i++){

    mask[i] = 0;

      for(auto x : se[i]){

        mask[i] |= 1 << x;

      }

   }

   int ans = 0;

   for(int i = 0; i < n; i++){

    for(int j = i + 1; j < n; j++){

       int cnt = \_\_builtin\_popcount(mask[i] & mask[j]);

       ans = max(ans,cnt);

    }

   }

   cout << ans << "\n";

 }

**N = 1000 er jonno to 1000 index possible na so solution ki**

**Bitset**

  bitset<10>b(16);

   cout << b << "/n";

**b(“1001”)**

**cout binary tei hobe**

#include<bits/stdc++.h>

using namespace std;

#define faster ios\_base::sync\_with\_stdio(0),cin.tie(0),cout.tie(0);

const int N = 101;

set<int>se[1010];

bitset<1000>mask[1010];

int a[N],pref[N];

 int32\_t main(){

   faster;

   int n = 2;

   se[0] = set<int>({1, 2, 3,6});

   se[1] = set<int>({1, 3, 5});

   for(int i = 0; i < n; i++){

      for(auto x : se[i]){

        mask[i][x] = 1;

      }

   }

   int ans = 0;

   for(int i = 0; i < n; i++){

    for(int j = i + 1; j < n; j++){

       int cnt =(mask[i] & mask[j]).count();

       ans = max(ans,cnt);

    }

   }

   cout << ans << "\n";

 }

**Bitset<n>b (here n always work constant)**

**Preparing Olympiad**

#include<bits/stdc++.h>

using namespace std;

#define faster ios\_base::sync\_with\_stdio(0),cin.tie(0),cout.tie(0);

 int32\_t main(){

   faster;

   int ans = 0;

   int n, l, r, x;

   cin >> n >> l >> r >> x;

   vector<int>v(n);

   for(int i = 0; i < n ; i++){

    cin >> v[i];

   }

   for(int mask = 1; mask < (1 << n); mask++){

        int sum = 0;

        int low = INT\_MAX, high = INT\_MIN;

        for(int i = 0; i < n; i++){

              if((mask >> i) & 1){

                  sum += v[i];

                  low = min(low,v[i]);

                  high = max(high,v[i]);

              }

        }

        if((sum >= l and sum <= r) and (high - low) >= x)

                    ans++;

   }

     cout << ans << "\n";

 }

**Petr and a Combination Lock**

#include<bits/stdc++.h>

using namespace std;

int main(){

    ios\_base::sync\_with\_stdio(0);

    cin.tie(0);

    int n; cin>>n;

    int a[n];

    for (int i = 0; i < n; i++) {

        cin>>a[i];

    }

    bool ok = false;

    for (int mask = 0; mask < (1 << n); mask++) {

        int sum = 0;

        for (int i = 0; i < n; i++){

            if ((mask >> i) & 1) sum += a[i];

            else sum -= a[i];

        }

        if (sum % 360 == 0){

            ok = true;

            break;

        }

    }

    if (ok) cout<<"YES\n";

    else cout<<"NO\n";

    return 0;

}

**Xor er prefix**

#include <bits/stdc++.h>

using namespace std;

int main() {

     int n; cin >> n;

     bitset<10>b;

     map<int,bitset<10>>mp;

     for(int i = 0; i < n; i++){

        int x; cin >> x;

        if(b[x])

        b[x] = 0;  //x or 0 == x

        else

        b[x] = 1;  // x or x == 0

        mp[i] =b;

     }

     for(auto it : mp){

        cout << it.first << " " << it.second << "\n";

     }

     int q; cin >> q;

     while(q--){

         int l, r; cin >> l >> r;

         l--;r--;

         b = mp[r];

         if(l){

            b = mp[r] ^ mp[l - 1];

         }

         cout << b.count() << "\n";

     }

}

**Odd Topic xor prefix sum**

#include <bits/stdc++.h>

using namespace std;

int main() {

    // your code goes here

    int n,m,q,t;

    cin>>n>>m>>q;

    map<int,bitset<5>> m1;

    map<int,bitset<5>> m2;

    bitset<5> b1,b2,b;

    for(int i=0;i<n;i++){

        cin>>t;

        if(b1[t]) b1[t]=0;

        else b1[t]=1;

        m1[i]=b1;

    }

    for(int i=0;i<m;i++){

        cin>>t;

        if(b2[t]) b2[t]=0;

        else b2[t]=1;

        m2[i]=b2;

    }

        int l1,r1,l2,r2;

    while(q--){

        cin>>l1>>r1>>l2>>r2;

        if(l1 == 1) b=m1[r1-1];

        else b = (m1[r1-1]^m1[l1-2]);

        if(l2 > 1) b=b^m2[l2-2];

        b=b^m2[r2-1];

        cout<<b.count()<<endl;

    }

    return 0;

}

**Mocha and Math**

1. **#include**<bits/stdc++.h>
2. **using** **namespace** std;
3. **int** main()
4. {
5. **int** t;
6. cin>>t;
7. **while**(t--)
8. {
9. **int** n; cin>>n;
10. **long** **long** **int** a[n];
11. **for**(**int** i=0;i<n;i++)
12. cin>>a[i];
13. **long** **long** **int** s=a[0];
14. **for**(**int** i=1;i<n;i++)
15. {
16. s=(s&a[i]);
17. }
18. cout<<s<<endl;
20. }
21. }

**AND 0, Sum Big**

**X X**

**X X**

**0 0**

**1st col e 1 ta zero bosate pari nC1**

**nC1 \* nC1 = n^2 = n^k**

#include<bits/stdc++.h>

using  namespace  std;

#define ll long long

#define mod 1000000007

ll power(ll base, ll p)

{

    ll ans=1;

    while(p)

    {

        if(p%2==1)

        {

            ans=(ans\*base)%mod;

            p--;

        }

        else{

            base=(base\*base)%mod;

            p/=2;

        }

    }

    return ans%mod;

}

int main()

{

    int t;

    cin>>t;

    while(t--)

    {

        ll n,k;

        cin>>n>>k;

        cout<<power(n,k)<<endl;

    }

}

**Rock and Lever**

#include<bits/stdc++.h>

using namespace std;

#define faster ios\_base::sync\_with\_stdio(0),cin.tie(0),cout.tie(0);

 int32\_t main(){

   faster;

    int t; cin >> t;

    while(t--){

        int n; cin >> n;

        int a[32] = {0};

        for(int i = 1; i <= n; i++){

            int x; cin >> x;

            a[31 - \_\_builtin\_clz(x)]++;

        }

        long long ans = 0;

        for(int i = 0; i < 32; i++){

            ans += (1LL \* a[i] \* (a[i] - 1)) / 2LL;

        }

        cout << ans << "\n";

    }

 }

**Little Girl and Maximum XOR**

**Duita bit er Xor maximum number koto hote pare**

**Let**

**16 = 10000**

**8 = 1000**

**15 = 1111**

**01000**

**11111**

**Duita range er max hobe hocche**

**Right most and left most equal na hole oitai answer**

using namespace std;

#define faster ios\_base::sync\_with\_stdio(0),cin.tie(0),cout.tie(0);

 int32\_t main(){

   faster;

   bitset<1000>b;

   bitset<1000>b1;

   long long l, r; cin >> l >> r;

   b = l;

   b1 = r;

   if(l == r){

    cout << 0 << "\n";

    return 0;

   }

   for(int i = 999; i >= 0; i--){

       if(b[i] != b1[i]){

         cout << (1LL << (i + 1)) - 1 << "\n";

         return 0;

       }

   }

 }

**And Then There Were K**

**N and 2^k – 1 == 0**

**Where k = n er bit length**

**N = Bit length 3 hole (n- 1) er bit length must be 2 hote hobe**

**2bit length er maximun b=number (2^2 – 1)**

#include<bits/stdc++.h>

using namespace std;

#define faster ios\_base::sync\_with\_stdio(0),cin.tie(0),cout.tie(0);

#define w(t) while(t--)

int32\_t main(){

   faster;

   int t; cin >>t;

   w(t)

   {

   long long l, r; cin >> l;

   int x = 31 - \_\_builtin\_clz(l);

   cout << (1 << x) - 1 << "\n";

   }

 }

Polo the Penguin and XOR operation

#include<bits/stdc++.h>

using namespace std;

const int N = 1e6 + 9;

int a[N];

bool vis[N];

#define faster ios\_base::sync\_with\_stdio(0),cin.tie(0),cout.tie(0);

int32\_t main(){

    memset(vis, false,sizeof(vis));

    int n; cin >> n;

    int mask = (1 << 20) - 1;

    for(int i = n; i >= 0; i--){

        while((((mask ^ i) > n) or vis[mask ^ i])){

          mask = (mask >> 1);

        }

        a[i] = mask ^ i;

        vis[mask ^ i] = true;

    }

    cout << 1LL \* n \* (n+1) <<"\n";

    for(int i = 0; i <= n; i++){

      cout << a[i] << " ";

    }

}

**AGAGA XOOORRR**

**Case 1: 5 , 5, 5 , 5**

**2 partition need**

**Case2:5, 5 ,5**

**3 partition need**

#include<bits/stdc++.h>

using namespace std;

#define faster ios\_base::sync\_with\_stdio(0),cin.tie(0),cout.tie(0);

int32\_t main(){

    int t; cin >> t;

    while(t--){

      int n; cin >> n;

      int a[n];

      int pref[n + 10] = {0};

       for(int i=1;i<=n;i++)cin>>a[i];

      for(int i = 1; i <= n; i++){

         pref[i] = pref[i - 1] ^ a[i];

      }

      bool ok = false;

      for(int i = 1; i < n; i++){

         int x = pref[i];

         int y = pref[n] ^ pref[i];

         if(x == y){

          ok = true;

          break;

         }

      }

      for(int i = 1; i < n - 1; i++){

        if(ok == true)

         break;

          int x = pref[i];

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

               int y = pref[j] ^ pref[i];

               int z = pref[n] ^ pref[j];

               if(x == y && x == z){

                ok = true;

                break;

               }

          }

      }

      if(ok){

        cout << "YES\n";

        continue;

      }

      cout << "NO\n";

    }

}

Fedor and New Game

#include<bits/stdc++.h>

using namespace std;

#define faster ios\_base::sync\_with\_stdio(0),cin.tie(0),cout.tie(0);

 int32\_t main(){

   faster;

    int n, m, k; cin>>n>>m>>k;

  int a[m];

  for (int i = 0; i <= m; i++){

    cin>>a[i];

  }

  int cnt = 0;

  for (int i = 0; i < m; i++){

    int nw = a[i]^a[m];

    int dif = \_\_builtin\_popcount(nw);

    if(dif <= k) cnt++;

  }

       cout<<cnt;

}

**Divan and bitwise operations**

**Sum of subsequence of all X-or is 2^(n-1) \* (or of all array)**

#include <iostream>

#include <algorithm>

#include <cmath>

#include <climits>

#include <vector>

#include <map>

#include <set>

#include <queue>

#include <stack>

#define ll long long int

#define mod 1000000007

using namespace std;

ll power(ll x, ll y)

{

    int res = 1;

    x = x % mod;

    if (x == 0)

        return 0;

    while (y > 0)

    {

        if (y & 1ll)

            res = (res \* x) % mod;

        y = y >> 1ll;

        x = (x \* x) % mod;

    }

    return res;

}

int main()

{

    ios\_base::sync\_with\_stdio(false);

    cin.tie(NULL);

    cout.tie(NULL);

    int T;

    cin >> T;

    while (T--)

    {

        ll n, m, l, r, x, totOR = 0, res = 0;

        cin >> n >> m;

        while (m--)

        {

            cin >> l >> r >> x;

            totOR |= x;

        }

        cout << (totOR \* power(2, n - 1)) % mod << endl;

    }

    return 0;

}

**Array Elimination**

**13 & 17 =**

**13 – 1, 17 – 1**

**12 ,16**

**12 & 16 == 0**

**So 0 theke kono number biyog dile ta zero ei hobe**

**2 ta numbers niye zero kora possible na**

**So er jonno sob gulo binary er sum % 2 hote hobe**

**1 & 0 = 0**

**On bit off hoye gelo amake jehuto on rakte hobe (tai k = 2 thakle each index ke 2 dara divide jete hobe)**

1. **#include<iostream>**
2. **#include<vector>**
3. **#define** ll **long** **long**
4. **using** **namespace** std;
5. **int** main()
6. {
7. **int** t; cin>>t;
8. **while**(t--)
9. {
10. ll n; cin>>n;
11. **vector<ll>**v;
12. **vector<ll>**bits(30,0);
13. **for**(ll i=0;i<n;i++)
14. {
15. ll x; cin>>x;
16. v.push\_back(x);
17. }
18. **for**(ll i=0;i<n;i++)
19. {
20. ll temp=v[i];
21. **int** pos=0;
22. **while**(temp)
23. {
24. **if**(temp&1)
25. bits[pos]++;
26. pos++;
27. temp=temp>>1;
28. }
29. }
31. **for**(**int** k=1;k<=n;k++)
32. {
33. **int** flag=1;
34. **for**(ll i=0;i<30;i++)
35. {
36. **if**(bits[i]%k)
37. {
38. flag=0;
39. **break**;
40. }
41. }
42. **if**(flag)
43. cout<<k<<**" "**;
45. }
46. cout<<endl;
47. }
48. }