[O - XOR of Sum of Pairs](https://vjudge.net/problem/HackerRank-si-xor-of-sum-of-pairs" \t "_blank)

 #include<bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

int sum=0;

vector<int>v(n);

for(int i=0;i<n;i++)

{

cin>>v[i];

}

for(int i=0;i<n;i++)

{

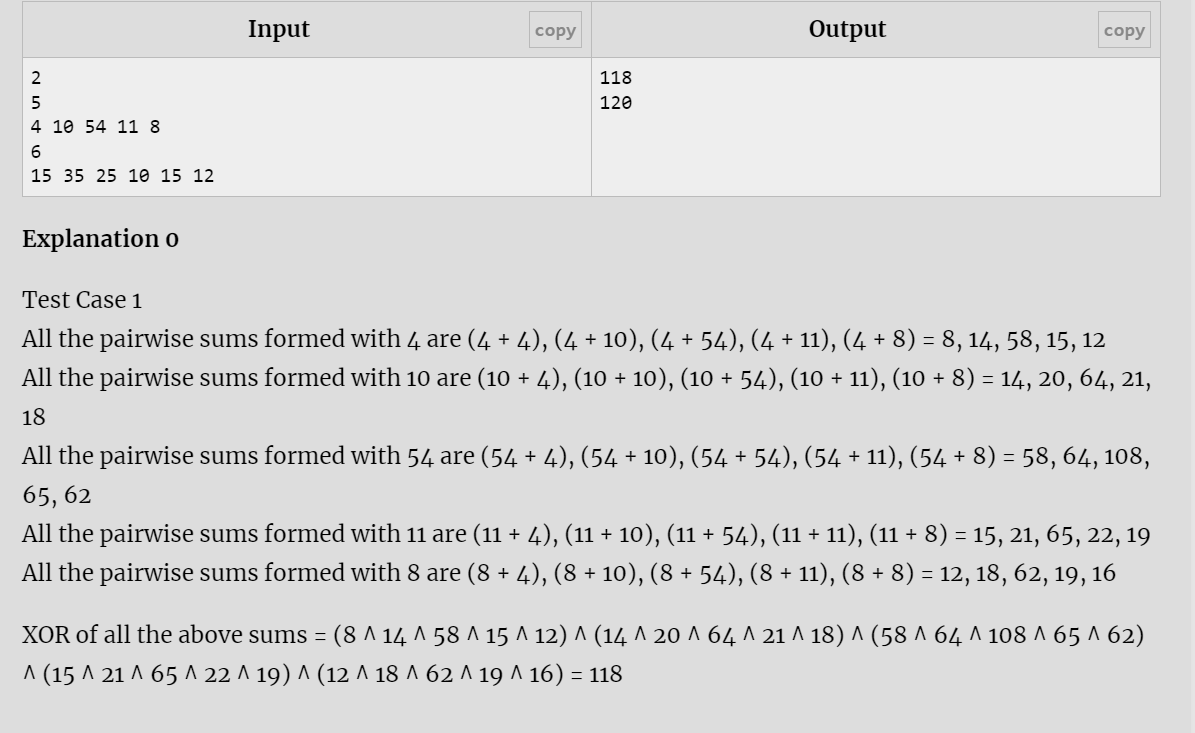
sum^=(v[i]\*2);

}

cout<<sum<<"\n";

}

}



[N - Repeated Numbers](https://vjudge.net/problem/HackerRank-si-repeated-numbers)

 #include<bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

int sum=0;

vector<int>v(n);

for(int i=0;i<n;i++)

{

cin>>v[i];

}

for(int i=0;i<n;i++)

{

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

{

if(v[i]==v[j])

{

cout<<v[i]<<" ";

}

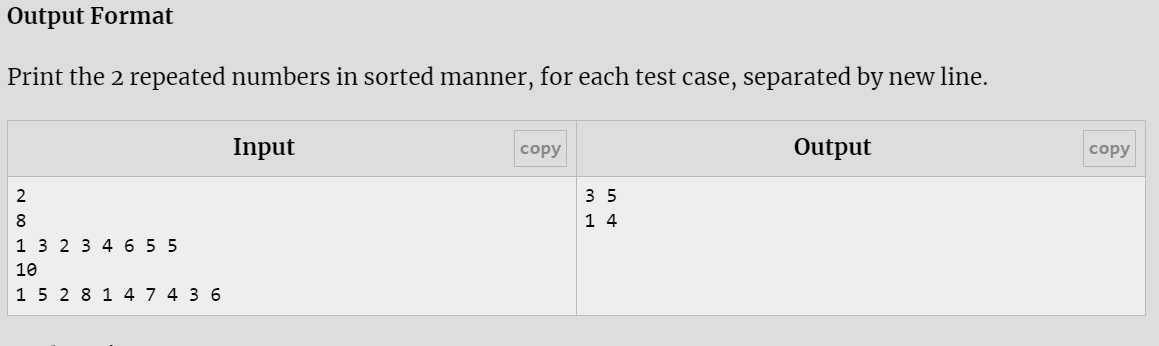
}

}

cout<<"\n";

}

}



[M - Triple Trouble](https://vjudge.net/problem/HackerRank-si-triple-trouble)

#include<bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

vector<long int>v(n);

for(int i=0;i<n;i++)

{

cin>>v[i];

}

for(int i=0;i<n;i++)

{

int flag=0;

for(int j=0;j<n;j++)

{

if(v[i]==v[j]&&i!=j)

{

flag=1;

}

}

if(flag==0)

{

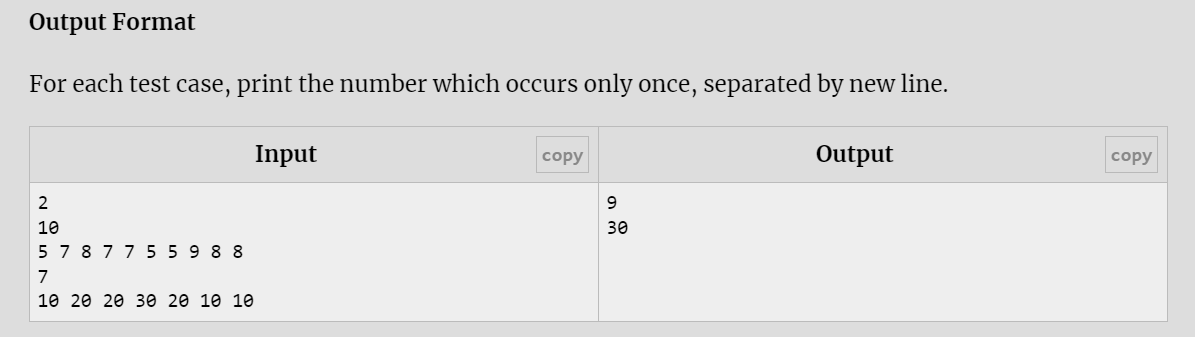
cout<<v[i]<<" \n";

}

}

}

}



[L - Count Set Bits](https://vjudge.net/problem/HackerRank-si-count-set-bits)

 #include<bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

long int n;

cin>>n;

int count=0;

while(n!=0)

{

if((n&1)==1)

count++;

n=n>>1;

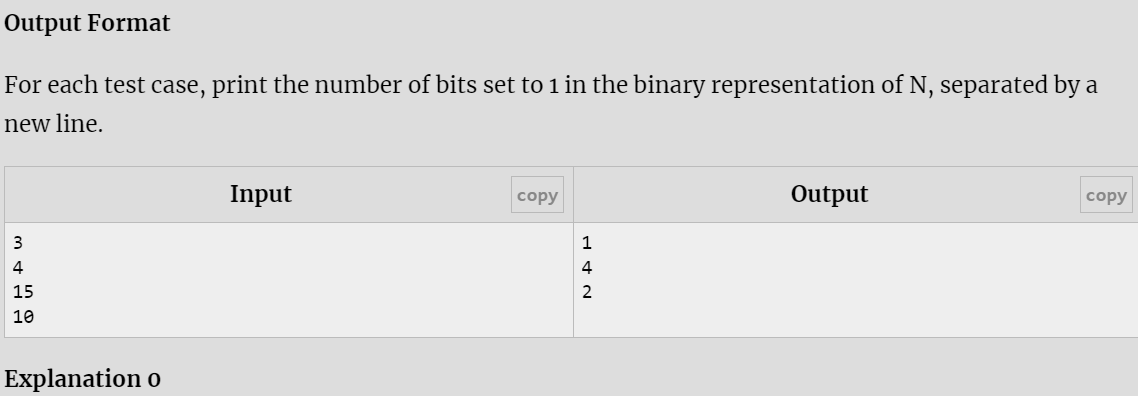
}

cout<<count<<"\n";

}

return 0;

}



[K - Subsets of an array](https://vjudge.net/problem/HackerRank-si-subsets-of-an-array)

 #include <bits/stdc++.h>

using namespace std;

void generate\_subsets(vector<int>& arr, vector<int>& current, int index) {

// Print the current subset

if (!current.empty()) {

for (int i = 0; i < current.size(); ++i) {

if (i > 0) cout << " ";

cout << current[i];

}

cout << endl;

}

// Generate subsets recursively

for (int i = index; i < arr.size(); ++i) {

current.push\_back(arr[i]);

generate\_subsets(arr, current, i + 1);

current.pop\_back();

}

}

int main() {

int t;

cin>>t;

while(t--)

{

int N;

cin >> N;

vector<int> arr(N);

for (int i = 0; i < N; ++i) {

cin >> arr[i];

}

sort(arr.begin(), arr.end()); // Sort array to generate subsets in lexicographical order

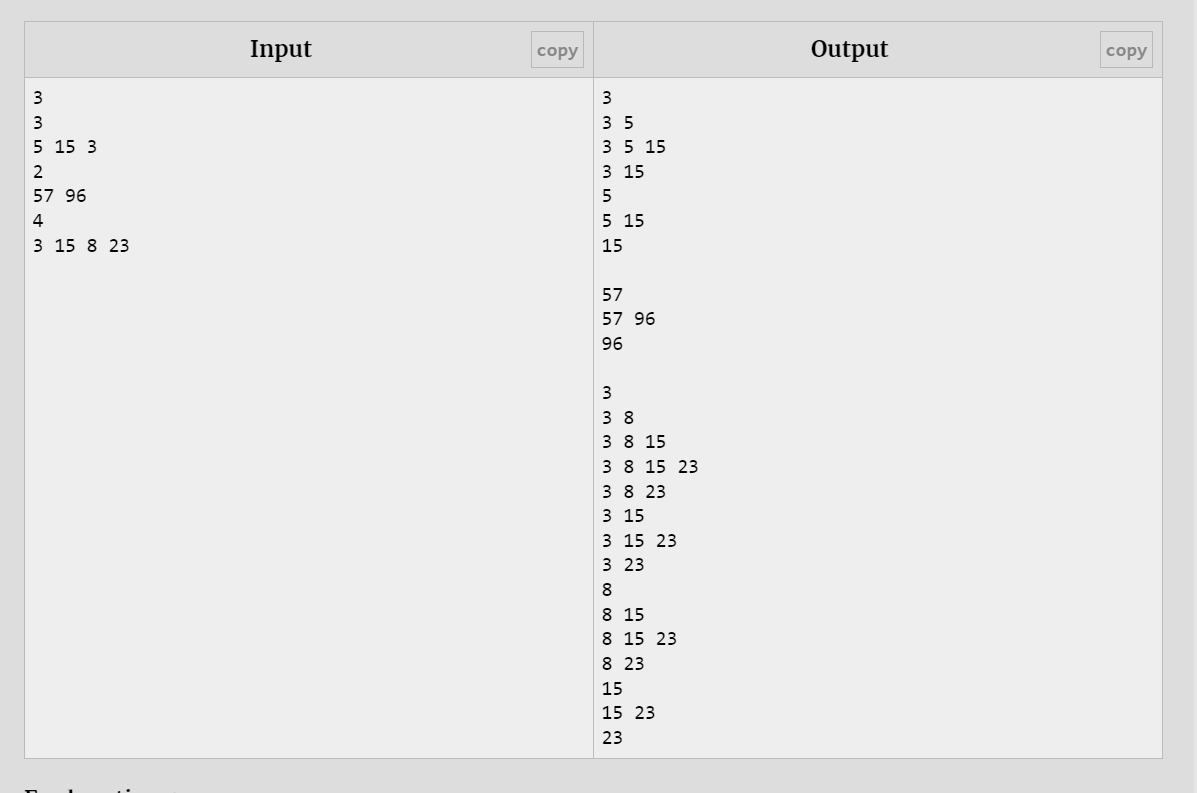
vector<int> current;

generate\_subsets(arr, current, 0);

}

return 0;

}



[J - 2^k + 2^n](https://vjudge.net/problem/EOlymp-5314)

 #include<bits/stdc++.h>

using namespace std;

int main()

{

int k,n;

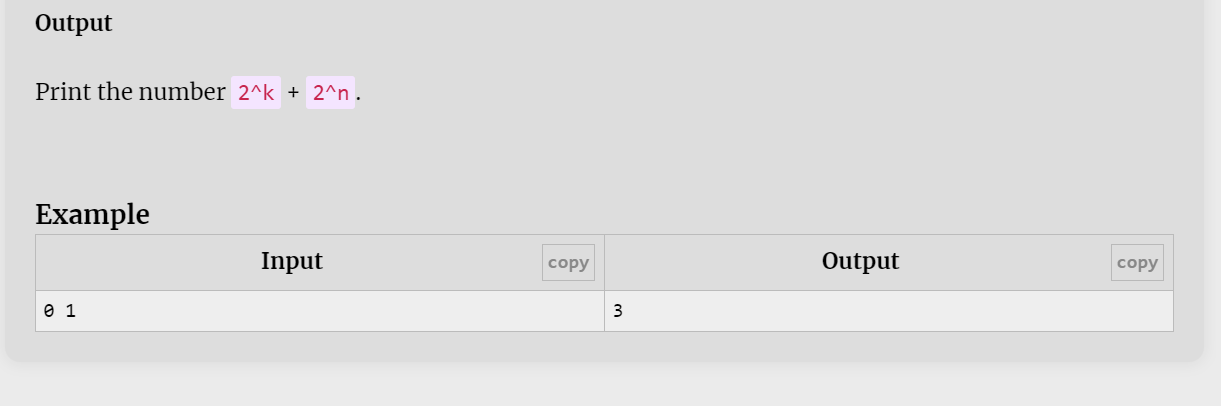
cin>>k>>n;

int a=pow(2,k);

int b=pow(2,n);

cout<<a+b;

}



[I - Flipping bits](https://vjudge.net/problem/HackerRank-flipping-bits)

 #include<bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

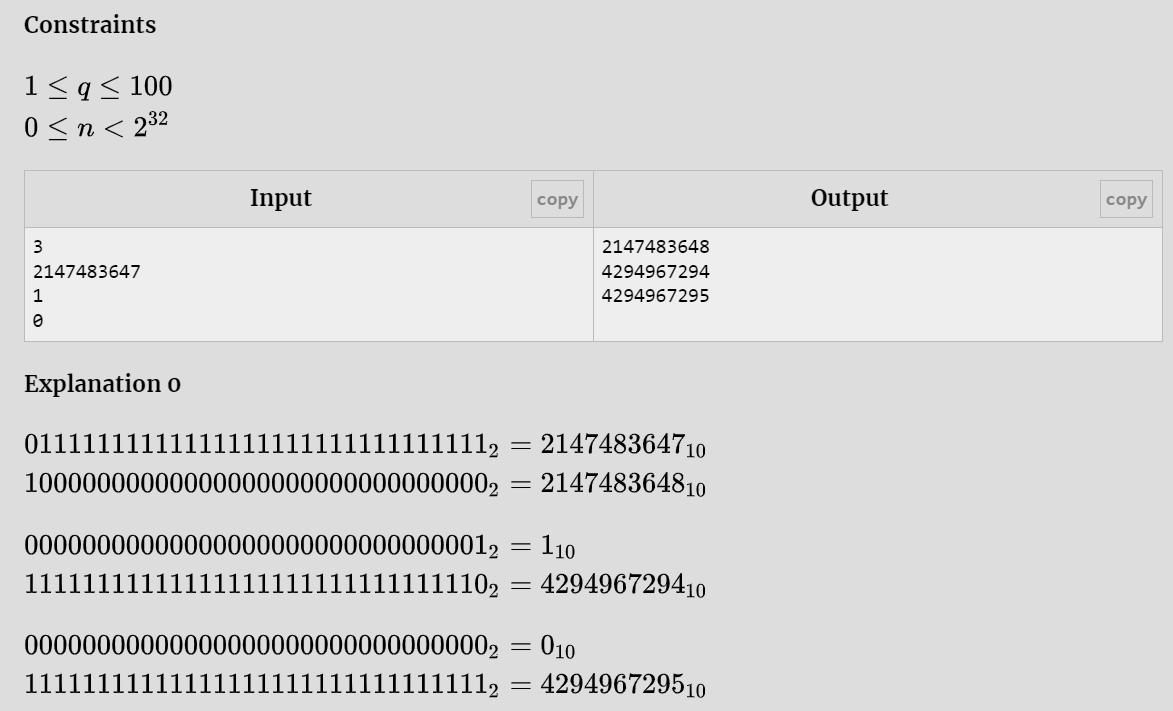
unsigned int n;

cin>>n;

cout<<(~n)<<"\n";

}

}



[H - Set a bit](https://vjudge.net/problem/EOlymp-5315)

 #include<bits/stdc++.h>

using namespace std;

int main()

{

int a,k;

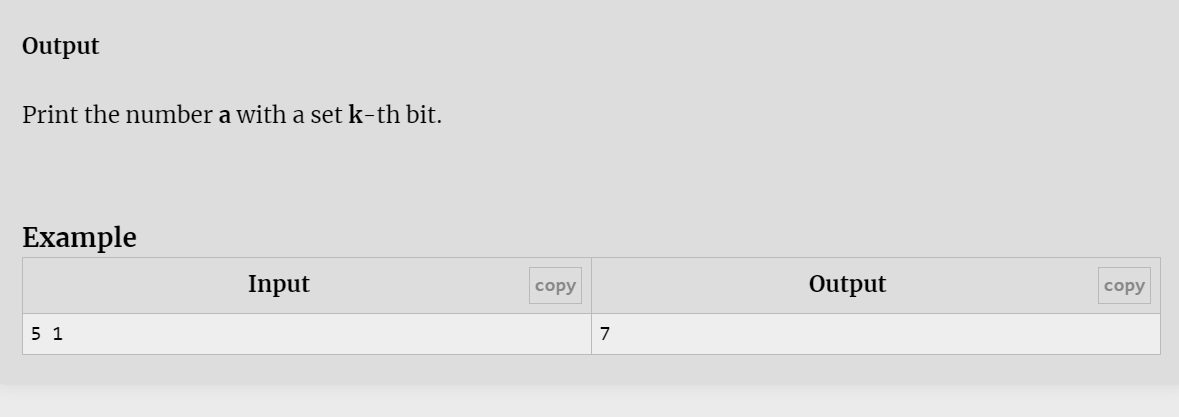
cin>>a>>k;

int c=pow(2,k);

a=(a|c);

cout<<a<<"\n";

}



[G - Compute a power b](https://vjudge.net/problem/HackerRank-si-compute-a-power-b)

 #include<bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

int a,b;

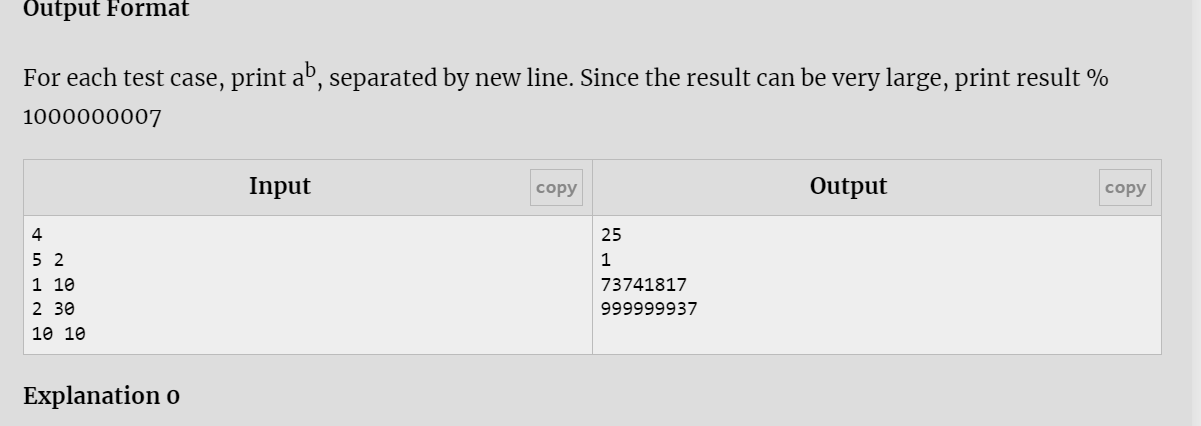
cin>>a>>b;

int c=pow(a,b);

cout<<c<<"\n";

}

}



[F - Swap Bits](https://vjudge.net/problem/HackerRank-si-swap-bits)

#include<bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

int e,o;

o=(n & 0XAAAAAAAA);

e=(n & 0X55555555);

o=o>>1;

e=e<<1;

cout<<(o|e)<<" ";

}

}



[A - Check bit](https://vjudge.net/problem/HackerRank-si-basic-check-bit)

 #include <iostream>

using namespace std;

int main()

{

int n,i;

cin>>n>>i;

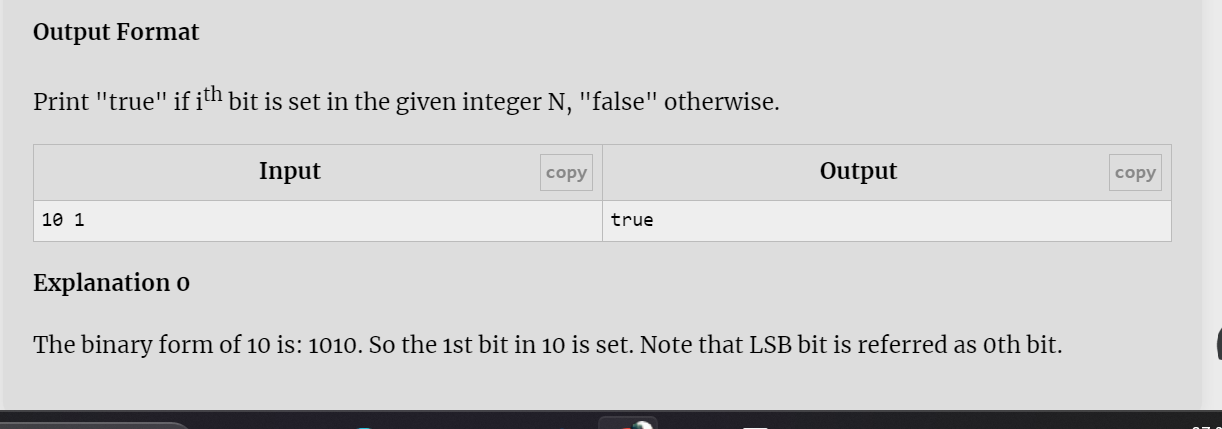
if( (n &(1<<i)) != 0 )

cout<<"true";

else

cout<<"false";

}



[B - Check Power of Two](https://vjudge.net/problem/HackerRank-si-check-power-of-two)

#include <iostream>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

long int n;

cin>>n;

if((n&(n-1))==0)

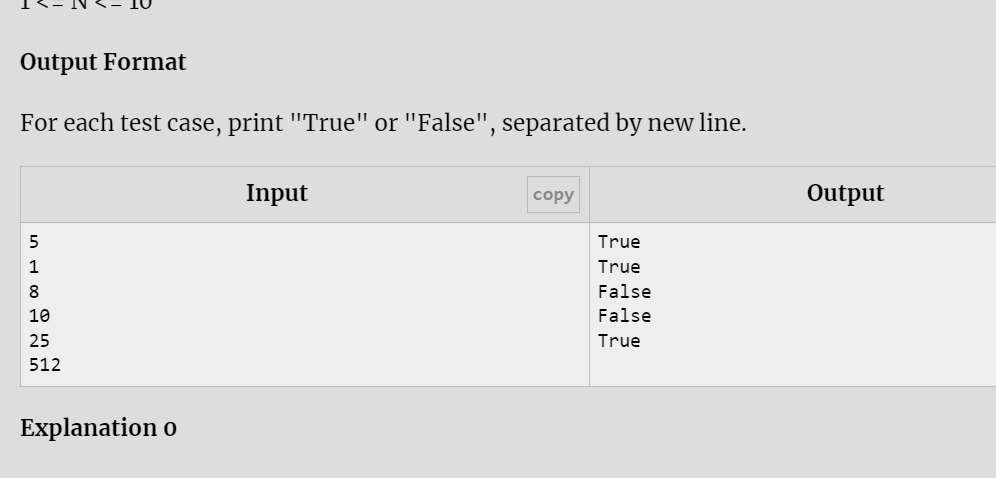
cout<<"True"<<"\n";

else

cout<<"False"<<"\n";

}

}



[C - Finding Missing Number](https://vjudge.net/problem/HackerRank-si-finding-missing-number)

 #include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

int a,n,sum=0,i;

cin>>n;

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

{

cin>>a;

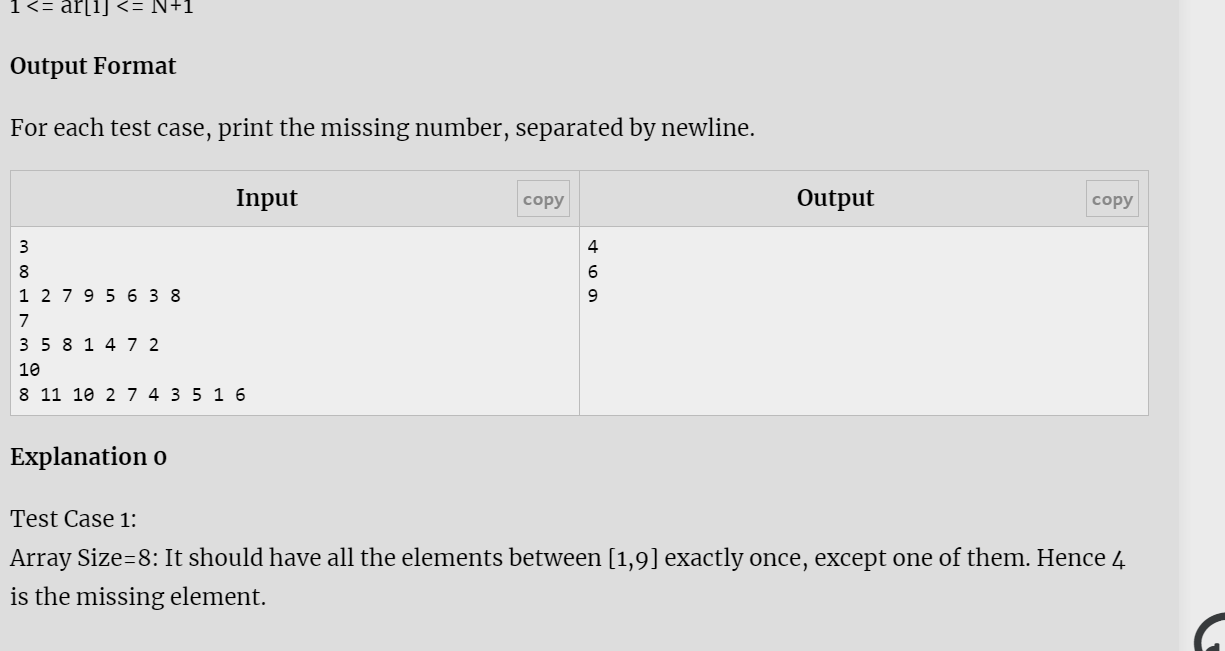
sum=sum^a^i;

}

cout<<(sum^n+1)<<"\n";

}

}



[D - Flip Bits](https://vjudge.net/problem/HackerRank-si-flip-bits)

 #include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

int a,b,c,sum=0;

cin>>a>>b;

c=a^b;

while(c!=0)

{

if( (c&1) == 1 )

sum++;

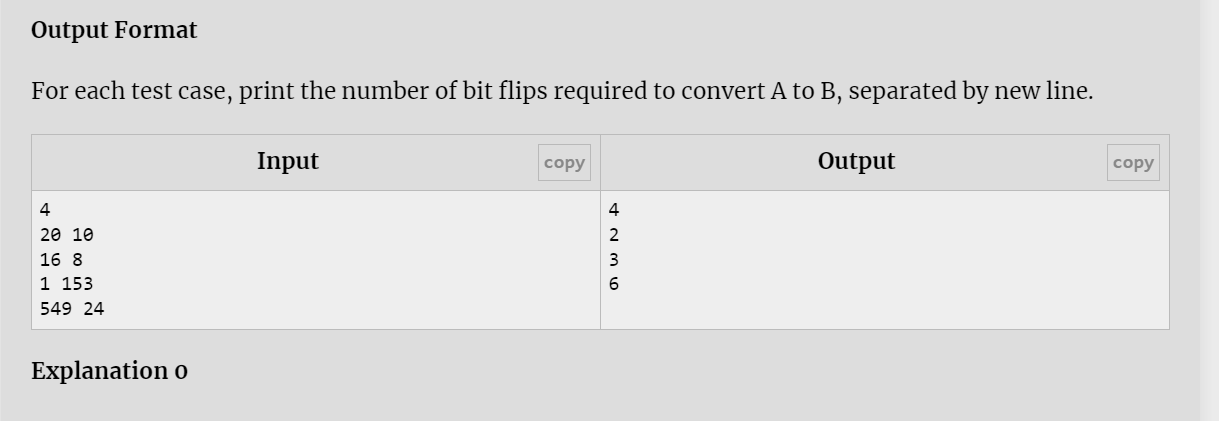
c=c >> 1;

}

cout<< sum <<"\n";

}

}



[E - Reverse Bits](https://vjudge.net/problem/HackerRank-si-reverse-bits)

 #include <bits/stdc++.h>

using namespace std;

int main()

{

unsigned int t,n,res,i;

cin>>t;

while(t--)

{

cin>>n;

res=0;

for(i=0;i<31;i++)

{

if((n&1)==1)

{

res=(res|1);

}

n=n>>1;

res=res<<1;

}

cout<<res<<"\n";

}

}

