L13 : Bit Manipulation

1-Tut : Predict The Output

Send Feedback

```
#include <iostream>
using namespace std;

int main(){
    int x = 2;
    x = x << 1;
    cout << x;
}
```

Answer

Type here : 4

Correct Answer

2-Tut : Predict The Output

Send Feedback

```
#include <iostream>
using namespace std;

int main(){
    int x = -2;
    x = x >> 1;
    cout << x;
}
```

Answer

Type here : -1

Correct Answer

3-Tut : Predict The Output

Send Feedback

```
#include <iostream>
using namespace std;

int main(){
    if(~0 == 1) {
        cout << "yes";
    }
    else {
        cout << "no";
    }
}
```

Options

This problem has only one correct answer

yes
no
Compile time error
Undefined

Correct Answer : B (~0 online compiler shows it as -1)

4-Tut : Predict The Output

Send Feedback

```
#include <iostream>
using namespace std;

int main(){
    int y = 0;
    if(1 | (y = 1)) {
        cout << "y is " << y;
    }
    else {
        cout << y;
    }
}
```

Options

This problem has only one correct answer

y is 0
y is 1
1
0

Correct Answer : B

5-Tut : Predict The Output

Send Feedback

```
#include <iostream>
using namespace std;

int main(){
    int y = 1;
    if(y & (y = 2)) {
        cout << "true";
    }
    else {
        cout << "false";
    }
}
```

Answer
Type here : true

Correct Answer

6-Tut : Turn Off The Bit
Send Feedback

Which bitwise operator is suitable for turning off a particular bit in a number?

Options
This problem has only one correct answer

&& operator
& operator
|| operator
| operator

Correct Answer : B

7-Tut : Turn On The Bit
Send Feedback

Which bitwise operator is suitable for turning on a particular bit in a number?

Options
This problem has only one correct answer

&& operator
& operator
|| operator
| operator

Correct Answer : D

8-Tut : Check ith bit
Send Feedback

Which bitwise operator is suitable for checking whether a particular bit is on or off?

Note: Multiple options can be correct

Options
This problem may have one or more correct answers

&& operator
& operator
|| operator
| operator
! operator
^ operator

The solution to this problem has been viewed

Solution Description

If we want to find whether the ith bit is set or not for a given number N.
Then we can right shift given number(N) by (i - 1). Let's call this number b; b=(N>>(i - 1))

a) Using & operator: we take (b&1) if the result is 1, our ith bit was set else it was not set.

b) Using | operator: we take (b|0) if the result is 1, our ith bit was set else it was not set.

c) Using ^ operator: we take (b^0) if the result is 1, our ith bit was set else it was not set.

9-Ass : Set ith Bit

Send Feedback

You are given two integers N and i. You need to make ith bit of binary representation of N to 1 and return the updated N.

Counting of bits start from 0 from right to left.

Input Format:

First line of input will contain T(number of test cases), each test case follows as.
A single line containing two space-separated integers N and i.

Output Format:

Updated N for each test case in new line.

Constraints:

1 <= T <= 10^5
1 <= N <= 10^9
1 <= i <= 30

Sample Input 1 :

```
1
4 1
```

Sample Output 1 :

```
6
```

Sample Input 2 :

```
1
4 4
```

Sample Output 2 :

```
20
```

```
#include<bits/stdc++.h>
using namespace std;
int setibit(int N, int i){
    return (N | (1 << i));
}
int main(){

    // write your code here
    int T; cin >> T;
    while(T--){
        int N,i; cin >> N >> i;
        cout << setibit(N,i) << endl;
    }
    return 0;
}
```

10-Ass : Unset ith Bit

Send Feedback

You are given two integers N and i. You need to make ith bit of binary representation of N to 0 and return the updated N.

Counting of bits start from 0 from right to left.

Input Format:

First line of input contains T(number of test cases), each test case follows as.
Two integers N and i (separated by space)

Output Format :

Updated N for each test case in new line.

Constraints:

1 <= T <= 10^5
1 <= N <= 10^9
1 <= i < 30

Sample Input 1 :

```
1
7 2
```

Sample Output 1 :

```
3
```

Sample Input 2 :

```
1
12 1
```

Sample Output 2 :

```
12
```

```
#include<bits/stdc++.h>
using namespace std;
int unsetibit(int N, int i){

    return (N & ~(1 << i));

}
int main(){

    // write your code here
    int T; cin >> T;
    while(T--){
        int N,i; cin >> N >> i;
        cout << unsetibit(N,i) << endl;
    }
    return 0;
}
```

11-Ass : Find First Set Bit

Send Feedback

You are given an integer N. You need to return an integer M, in which only one bit is set which at the position of a lowest set bit of N (from right to left).

Input Format :

The first line of input will contain T(number of the test case), each test case follows as.
The only line of each test case contains an integer N.

Output Format:

Integer M for each test case in a new line.

Constraints:

1 <= T <= 10^5
1 <= N <= 10^9

Sample Input 1 :

1
7

Sample Output 1 :1

Sample Input 2 :

1
12

Sample Output 2 : 4

```
#include<bits/stdc++.h>
using namespace std;
int firstsetbit(int N){
    int ans = 1;
    while( !(N&1) ){
        ans <<= 1;
        N >>= 1;
    }
    return ans;
}
int main(){

    // write your code here
    int T; cin >> T;
    while(T--){
        int N; cin >> N;
        cout << firstsetbit(N) << endl;
    }
    return 0;
}
```

12-Ass : Turn Off First Set Bit

Send Feedback

You are given an integer Ni. You need to make rightmost set bit of binary representation of N to 0 and return the updated N.

Counting of bits start from 0 from right to left.

Input Format :

The first line of input will contain T(number of test cases), each test case follows as.
A single integer N for each test case in a newline.

Output Format :

Updated N for each test case in a newline.

Constraints:

1 <= T <= 10^5
1 <= N <= 10^9

Sample Input 1 :

1
4

Sample Output 1 :

0

Sample Input 2 :

1
12

Sample Output 2 :

8

```
#include<bits/stdc++.h>
using namespace std;
int firstsetbitoff(int N){
    int N1 = N;
    int ans = 1;
    while( (N1&1) == 0 ){
        ans <<= 1;
        N1 >>= 1;
    }
    return N^ans;
}
int main(){

    // write your code here
    int T; cin >> T;
    while(T--){
        int N; cin >> N;
        cout << firstsetbitoff(N) << endl;
    }
    return 0;
}
```

13-Ass : Clear All Bits From MSB

Send Feedback

You are given two integers N and i. You need to clear all bits from MSB to ith bit (start i from right to left) and return the updated N.

Counting of bits starts from 0 from right to left.

Input Format :

First line of input will contain T(number of test cases), each test case follows as.
Line1: contain two space-separated integers N and i.

Output Format :

Updated N for each test case in a newline.

Constraints:

1 <= T <= 10^5
1 <= N <= 10^9
1 <= i <= 30

Sample Input 1 :

1
15 2

Sample Output 1 :

3

```
#include<bits/stdc++.h>
using namespace std;
int clearMSB(int N, int i){
    int mask = (1 << i)-1;
    return (N & mask);
}
int main(){

    // write your code here
    int T; cin >> T;
    while(T--){
        int N,i; cin >> N >> i;
        cout << clearMSB(N,i) << endl;
    }
    return 0;
}
```

14-Ass : Odd Frequency

Send Feedback

You are given an array of size N with all elements with even frequency except one and you are supposed to find this element.

Input Format:

The first line of input will contain T(number of test cases), each test case follows as.
Line 1: contain an integer N (number of elements in the array)
Line 2: contain N space-separated integers (elements of the array).

Output Format:

For each test case print the element with the odd frequency in a new line.

Constraints:

1 <= T <= 50
1 <= N <= 10^5
1 <= arr[i] <= 10^9

Sample Input:

```
1
5
2 2 2 3 3
```

Sample Output:

```
2
```

```
#include<bits/stdc++.h>
using namespace std;
int oddfreq(int a[],int N){
    int ans = 0;
    for(int i=0;i<N;i++){
        ans=ans^a[i];
    }
    return ans;
}
int main(){

    // write your code here
    int T; cin >> T;
    while(T--){
        int N; cin >> N;
        int *arr = new int[N];
        for(int i = 0; i < N; i++){
            cin>>arr[i];
        }
        cout << oddfreq(arr,N) <<endl;
    }
    return 0; }
```

15-Ass : XOR of Natural Numbers

Send Feedback

You are given an integer N and asked to find the Xor of first N natural numbers.

Input Format:

The first line of input will contain T(number of test cases), each test case follows as.
The only line of input contains an integer N.

Output Format:

For each test case print the Xor of first N natural number in a new line.

Constraints:

1 <= T <= 10^5
1 <= N <= 10^9

Sample Input:

1
8

Sample Output: 8

```
#include<bits/stdc++.h>
using namespace std;
int xorofFNNN(int N){
   int rem = N % 4;
    if(rem == 0){
        return N;
    }
    if(rem == 1){
        return 1;
    }
    if(rem == 2){
        return N+1;
    }
    if(rem == 3){
        return 0;
    }
}
int main(){

    // write your code here
    int T; cin >> T;
    while(T--){
        int N; cin >> N;
        cout << xorofFNNN(N) << endl;
    }
    return 0;
}
```