2105A41178 DAY3

BITWISE OPERATORS

|  |  |  |  |
| --- | --- | --- | --- |
| Operators | C | Python | Java |
| and | & | and | & |
| or | | | or | | |
| negotitaon | ~ | ~ | ~ |
| not | ! | not | ! |
| 1’s compliment |  |  |  |
| 2’s compliment |  |  |  |

Priority of operators:

1st

# include<stdio.h>

int main()

{

printf("%d",10\*4/6+3-1%2);

}

OUTPUT:8

2nd

# include<stdio.h>

int main()

{

printf("%d",7+2&4+3&9);

}

OUTPUT :1

3rd

# include<stdio.h>

int main()

{

printf("%d",10/3&4);

}

OUTPUT:0

* Logic used on expression are:

1.Valid

2. Priority

3. System output

4th

# include<stdio.h>

int main()

{

printf("%d",10&4~2);

}

OUTPUT:invalid

5th

# include<stdio.h>

int main()

{

printf("%d",6|3&9+6);

}

OUTPUT: 7

6th

# include<stdio.h>

int main()

{

printf("%d",~9+4&6);

}

OUTPUT: 2

EXCLUSIVE (XOR)

|  |  |  |
| --- | --- | --- |
| A | B | OUTPUT |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

* XOR is with ‘0’ and any number is number itself.
* XOR of two same numbers is 0.

Example:5^0=5

1^2^3=0

4^6^5=7

INCLUSIVE (OR)

|  |  |  |
| --- | --- | --- |
| A | B | OUTPUT |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

RIGHT SHIFT(>>>)

1. 5>>2(it means we have to shift the 5 two times right shift)

ANS:1

1. 14>>2

ANS:3

LEFT SHIFT(<<<)

10<<3

ANS:80

PROGRAMS:

* Write a program to find the smallest positive missing integer from the given array or create a array?

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

System.out.println("Enter array size:");

int n=in.nextInt();

int[] arr=new int[n];

System.out.println("Enter array elements:");

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

arr[i]=in.nextInt();

}

checkWithCurr(arr); //Sorting O(n) time complexity

int ans=missing(arr);

System.out.println(ans);

}

static void checkWithCurr(int[] arr)

{

int i=0;

while(i<arr.length)

{

int correct=arr[i];

if(arr[i]>=arr.length)

{

i++;

}

else if(arr[i]<0){

i++;

}

else if(arr[i]==arr[correct])

{

i++;

}

else {

int temp=arr[i];

arr[i]=arr[correct];

arr[correct]=temp;

}

}

}

static int missing(int[] arr)

{

for(int i=0;i<arr.length;i++){

if(arr[i]!=i){

return i;

}

}

return arr.length;

}

}

* In the given array every integer occurs twice one integer occurs once.find the once occur integer.

Program:

def findSingle(ar,n):

res=ar[0]

for i in range(1,n):

res=res^ar[i]

return res

ar=[2,3,5,4,3,4,2,88]

print(findSingle(ar,len(ar)))

* SWAP 2 NUMBERS USING XOR

#include <stdio.h>

int main(){

int x, y;

//input

printf("Enter x: ");

scanf("%d", &x);

printf("Enter y: ");

scanf("%d", &y);

//swap

x = x ^ y;

y = x ^ y;

x https://www.onlinegdb.com/#editor\_1= x ^ y;

printf("After swap:\nx = %d\ny = %d\n", x, y);

return 0;

}

* For the given number check whether kth bit is set or not.

Program:

n=int(input())

k=int(input())

a=n&(1<<(k-1))

if(a==0):

print("not a set")

else:

print("set")

* For the given number n find out XOR of all n numbers

1^2^3^4^5^6^7^8^9^10

|  |  |
| --- | --- |
| EXOR | O/P |
| 1^2 | 3 |
| 1^2^3 | 0 |
| 1^2^3^4 | 4 |
| 1^2^3^4^5 | 1 |
| 1^2^3^4^5^6 | 7 |
| 1^2^3^4^5^6^7 | 0 |
| 1^2^3^4^5^6^7^8 | 8 |

Program:

n=int(input("enter num"))

xor=0

if n%4==0:

print(n)

elif n%4==1:

print(1)

elif n%4==2:

print(n+1)

elif n%4==3:

print(0)

* Find EXOR in a given range.

Lets Say

4 to 8

from operator import xor

# Function to return the XOR of elements

# from the range [1, n]

def findXOR(l, r):

ans = 0

for i in range(l,r+1):

ans = xor(ans,i)

return ans

# Driver code

l = 4; r = 8;

print(findXOR(l, r));

OUTPUT:8

* CHECK EVEN OR ODD USING BITWISE

Program:

n=int(input())

if(n&1==0):

print("even")

elif(n&1==1):

print("odd")

OUTPUT: 5

ODD