**Day-3 H:no:2105A41131**

Priority order:

#include<stdio.h>

#include<stdlib.h>

void main(){

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

}

OUTPUT:

8

1. Bit-wise operators

|  |  |  |
| --- | --- | --- |
| C | PYTHON | JAVA |
| & | and | & |
| | | or | | |
| ~ | ~ | ~ |
| ^ | ^ | ^ |

#include<stdio.h>

#include<stdlib.h>

void main(){

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

}

OUTPUT:

2

1. Priority order between bit-wise and arithmetic operators:

* Consider arithmetic first compared with bit-wise

Ex:7+2&4+3&9

* Convert decimal to binary by taking arithmetic as first.

Ans=1

Program:

#include<stdio.h>

#include<stdlib.h>

void main(){

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

}

Output:1

1. Priority order between bit-wise and bit-wise operators:

Q.10/3&4

* Consider or as priority compared with and

#include<stdio.h>

#include<stdlib.h>

void main(){

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

}

Output:0

Priority order of bit-wise operators:

&=2

|=4

~=1(1’s compliment or 2’s compliment)

^=3(XOR)

* 10&4~2=Invalid

#include<stdio.h>

#include<stdlib.h>

void main(){

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

}

Output: error

* 6|3&9+6=valid

Program:

#include<stdio.h>

#include<stdlib.h>

void main(){

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

}

Output:7

* 2~4^3\*2=invalid
* ~9+4&6=valid

Program:

#include<stdio.h>

#include<stdlib.h>

void main(){

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

}

Outpu:2

* Left<< -> shift bits from right to left
* Right>> -> shift bits from left to right

Ex:5>>2

Ans=2

BIT MANIPULATION TRICKS:

Xor-^

Even 1’s:0

Odd 1’s:1

* Xor of the number itself is 0

Ex:5^5=0

* Xor of number with 0 itself is 1

Ex:5^5^5

0^5=5

Ex:4^6^5

* Convert 4&6 in binary form(4->100 & 6->110) apply xor operation answer is (010->3)
* Convert 3&6 in binary form(3->010 & 5->101) apply xor operation answer is (011->7)
* 5-0101

5<<2

5\*pow(2,3)

Ans =20

1. After creating a array find out the the smallest missing positive integer.

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;

}

}

1. In the given array

Arr=[1,5,1,2,3,2,3],every integer ocuurs twice and one numbers occur once?find out that which occurs once?

def findSingle(ar,n):

res=ar[0]

for I in range(1,n):

res=res^ar[i]

return res

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

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

3.performing xor operation:xor of elements of all elements.same elements wiil get canclled.

Program:

a=100

b=200

print("a:",a,"b:",b)

a=a^b

b=a^b

a=a^b

print("a:",a,"b:",b)

4.Perform xor operation

n=12

xor=0

for i in range(1,n+1):

xor=xor^I

print(xor)

OPTIMIZE:

|  |  |
| --- | --- |
| 1 | 1 |
| 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 |
| 1^2^3^4^5^6^7^8^9 | 1 |

Ex: optimize program:

5.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)

6.Check the given number is odd or not by using bit-wise :

n=13

if(n&1==0):

print("Even")

else:

print("odd")