**CP Problem Statement: 2**

**A Job Ready Bootcamp in C++, DSA and IOT**

MySirG

2. **Product of Array Except Self**

Problem Statement

Given an integer array nums , return an array answer such that answer[i] is equal to the

product of all the elements of nums except nums[i] .

The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.

You must write an algorithm that runs in O(n) time and without using the division

operation

Example 1:

Input: nums = [1,2,3,4]

Output: [24,12,8,6]

Example 2:

Input: nums = [-1,1,0,-3,3]

Output: [0,0,9,0,0]

Sol –

#include<iostream>

#include<math.h>

using namespace std;

int divide(int,int);

int main()

{

int nums[5]={2,1,4,-3,3};

int n=5,i,flag=0,mult=1,answer[5];

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

cout<<nums[i]<<" ";

cout<<endl;

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

{

if(nums[i]==0)

flag++;

}

if(flag==0||flag==1)

{

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

{

if(nums[i]!=0)

mult=mult\*nums[i];

}

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

{

if(nums[i]==0)

answer[i]=mult;

else if(flag==1)

answer[i]=0;

else

answer[i]=divide(mult,nums[i]);

}

}

else

{

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

{

answer[i]=0;

}

}

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

cout<<answer[i]<<" ";

return 0;

}

int divide(int a,int b)

{

int sign=(a<0)^(b<0);

a=abs(a);

b=abs(b);

int ans=exp(log(a)-log(b))+0.000000001;

return sign==0?ans:-ans;

}

OR

#include<iostream>

using namespace std;

int main()

{

int nums[5]={0,1,0,-3,3};

int n=5,i,flag=0,mult=1,answer[5];

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

cout<<nums[i]<<" ";

cout<<endl;

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

{

if(nums[i]==0)

flag++;

}

if(flag==0||flag==1)

{

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

{

if(nums[i]!=0)

mult=mult\*nums[i];

}

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

{

if(nums[i]==0)

answer[i]=mult;

else if(flag==1)

answer[i]=0;

else

answer[i]=mult/nums[i];

}

}

else

{

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

{

answer[i]=0;

}

}

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

cout<<answer[i]<<" ";

return 0;

}