**Maximum Product Subarray**

**Easy**Accuracy: 15.28% Submissions: 1238 Points: 2

Given an array **Arr** that contains **N** integers (may be **positive**, **negative**or **zero**). Find the product of the maximum product subarray.

**Example 1:**

**Input:**

N = 5

Arr[] = {6, -3, -10, 0, 2}

**Output:** 180

**Explanation:** Subarray with maximum product

is 6, -3, -10 which gives product as 180.

**Example 2:**

**Input:**

N = 6

Arr[] = {2, 3, 4, 5, -1, 0}

**Output:** 120

**Explanation:** Subarray with maximum product

is 2, 3, 4, 5 which gives product as 120.

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **maxProduct()** which takes the array of integers **arr** and **n**as parameters and returns an integer denoting the answer.  
**Note:** Use 64-bit integer data type to avoid overflow.

**Expected Time Complexity:** O(N)  
**Expected Auxiliary Space:** O(1)

**Constraints:**  
1 <= N <= 500  
-102 <= Arri <= 102

#include<bits/stdc++.h>

#define pb push\_back

#define pii pair<int,int>

#define int long long int

#define vec vector<int>

#define inf 1e18

using namespace std;

int32\_t main()

{

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL);

cout.tie(NULL);

int tt=1;

//cin>>tt;

while(tt--)

{

int n;

cin>>n;

int a[n],i;

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

cin>>a[i];

int maxed=0,mined=0,ans=0;

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

{

int x=maxed,y=mined;

maxed=max(a[i],max(x\*a[i],y\*a[i]));

mined=min(a[i],min(x\*a[i],y\*a[i]));

ans=max(maxed,ans);

}

cout<<ans<<"\n";

}

}