**Maximum Product Subarray**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)[Amazon](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Amazon&isCmp=1)[Microsoft](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Microsoft&isCmp=1)[Morgan Stanley](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Morgan%20Stanley&isCmp=1)

Given an array that contains both positive and negative integers, find the product of the maximum product subarray.  
**Assumption: There is always a positive product possible, i.e., no array of this form: {0,-20,0,0} or {-20}.**

**Input:**  
First line of input contain number of test cases T. First line of test case contain the size of array and second line of test case contain the array elements.

**Output:**  
Maximum product of subarray is displayed to the user.

**Constraints:**  
1 <=T<= 50  
1 <=N<= 9  
-10 <=arr[i]<= 10

**Example:**

**Input:**

3  
5  
6 -3 -10 0 2  
6  
2 3 4 5 -1 0   
10  
8 -2 -2 0 8 0 -6 -8 -6 -1

**Output:**  
180  
120  
288

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=433>

#include <iostream>

#include <stdio.h>

using namespace std;

int main() {

    int t;

    scanf("%d", &t);

    while(t--) {

       int n;

       scanf("%d", &n);

       int arr[n];

       for(int i =0; i < n; i++) {

          scanf("%d", &arr[i]);

        }

       int max\_prod = 1;

       for(int i =0; i < n; i++) {

          int prod = 1;

          for(int j=i; j < n; j++) {

              prod \*= arr[j];

              max\_prod = std::max(max\_prod, prod);

            }

        }

        printf("%d**\n**", max\_prod);

    }

 return 0;

}