**Array of alternate +ve and -ve no.s**

[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)

Given an unsorted array of positive and negative numbers.Create an array of alternate positive and negative numbers without changing the relative order of positive and negative numbers respectively.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is N,N is the size of array.  
The second line of each test case contains N input a[].  
  
**Output:**

Print an array of alternate positive and negative numbers.  
  
**Constraints:**

1 ≤ T ≤ 30  
1 ≤ N ≤ 100  
-1000 ≤ a[] ≤ 1000  
  
**Example:**

Input  
1  
9  
9 4 -2 -1 5 0 -5 -3 2

Output  
9 -2 4 -1 5 -5 0 -3 2

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=444>

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <vector>

using namespace std;

int main() {

    int T;

    scanf("%d", &T);

    std::string s;

    while(T--) {

        int N;

        scanf("%d", &N);

        int arr[N]; //={9 ,4 ,-2, -1, 5, 0 ,-5, -3, 2};

*for(int i =0; i < N; i++) {*

*scanf("%d", &arr[i]);*

*}*

        std::vector<int> pos;

        std::vector<int > neg;

        for(int i =0; i < N; i++) {

            if(arr[i] < 0) {

                neg.push\_back(arr[i]);

            }else if (arr[i] >= 0) {

                pos.push\_back(arr[i]);

            }

        }

        std::vector<int> ans;

        int i =0, j=0;

        while(i < pos.size() && j < neg.size() ) {

            ans.push\_back(pos[i]);

            ans.push\_back(neg[j]);

            i++;

            j++;

        }

        while(i < pos.size()) {

            ans.push\_back(pos[i++]);

        }

        while(j <neg.size()) {

            ans.push\_back(neg[j++]);

        }

        for(int i =0; i < ans.size(); i++) {

            printf("%d ", ans[i]);

        }

        printf("**\n**");

    }

  system("pause");

 return 0;

}