Talent Battle 100 Days Coding Series

You are given an array A of N elements. For any ordered triplet (i,j,k) such that i, j, and k are pairwise distinct and $1 \le i,j,k \le N$, the value of this triplet is $(A_i - A_j) \cdot A_k$. You need to find the **maximum** value among all possible ordered triplets.

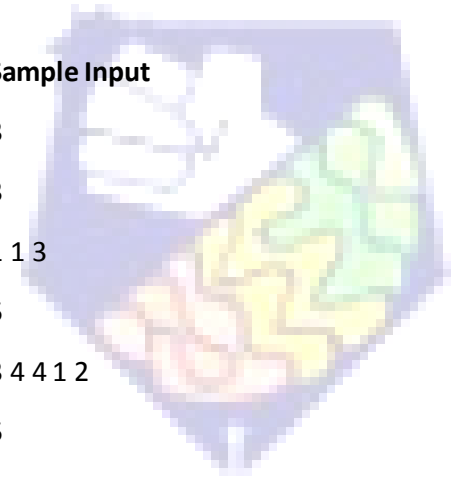
Note: Two ordered triplets (a,b,c) and (d,e,f) are only equal when $a=d$ **and** $b=e$ **and** $c=f$. As an example, (1,2,3) and (2,3,1) are two different ordered triplets.

Input Format

- The first line of the input contains a single integer T - the number of test cases. The test cases then follow.
- The first line of each test case contains an integer N.
- The second line of each test case contains N space-separated integers $A_1, A_2, \ldots, A_N$.

Output Format

For each test case, output the maximum value among all different ordered triplets.



TalentBattle

Sample Input

3

3

1 1 3

5

3 4 4 1 2

5

23 17 21 18 19

Sample Output

2

12

126

C

```
#include <stdio.h>
```

```
int main(void) {
        long long int tcase;
        scanf("%lld", &tcase);
        while(tcase--){
          long long int length, i, j, swap;
          scanf("%lld", &length);
          long long int a[length], min = 10000000,  max = 0, secMax = 0;
          for(i=0; i<length; i++){
            scanf("%lld", &a[i]);
            if(min>a[i]){
              min = a[i];
            }
            if(max<a[i]){
              max = a[i];
            }
          }
          for(i=0; i<length; i++){
            if(a[i]==max){
              a[i] = 0;
              break;
            }
          }
          for(i=0; i<length; i++){
            if(secMax<a[i]){
              secMax = a[i];
            }
          }
          printf("%lld\n", (max-min)*secMax);
        }
```

TalentBattle

```
    return 0;
}
```

C++

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int main() {
    int t;cin>>t;
    while(t--){
      long long int n;cin>>n;vector<int> v;
      for(int i=0;i<n;i++){
          long long int num;cin>>num;v.push_back(num);
      }
      sort(v.begin(),v.end());
      long long int ans = (long long int)(v[n-1]-v[0])*v[n-2];
      cout<<ans<<'\n';
    }
    return 0;
}
```

Java

```
import java.util.*;
import java.lang.*;
import java.io.*;
```

```
Class Main
{
        public static void main (String[] args) throws java.lang.Exception
        {
           BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    int t=Integer.parseInt(br.readLine());
    while(t-->0)
    {
       int n=Integer.parseInt(br.readLine());
       String s=br.readLine();
       String sr[]=s.split(" ");int ar[]=new int[n];
       for (int i=0;i<n ;i++)
         ar[i]=Integer.parseInt(sr[i]);
       Arrays.sort(ar);long cout=Integer.MIN_VALUE;
       cout=(long)(ar[n-1]-ar[0])*ar[n-2];
       System.out.println(cout);
    }
        }
}
```

TalentBattle

Python

```
t=int(input())
while(t>0):
   t=t-1
   n=int(input())
   a=input().split()
   for i in range(0,n):
      a[i]=int(a[i])
   a.sort()
```

```
a1=(a[-2]-a[0])*a[-1]
a2=(a[-1]-a[0])*a[-2]
print(max(a1,a2))
```


TalentBattle