

Given 2 integer arrays X and Y of same size. Consider both arrays as vectors and print the sum of maximum scalar product (Dot product) of 2 vectors.

Sample input 1:

4

1 2 3 4

5 6 7 8

Sample output 1:

70

Explanation :

$$(8*4 + 7*3 + 6*2 + 1*5) = 70$$

Sample input 2:

4

-1 -2 -3 -4

5 6 -7 -8

Sample output 2:

37

Explanation :

$$(-4*-8 + -3*-7 + -2*5 + -1*6) = 37$$

Solution:

C

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

```
#include <limits.h>
```

```
// SpecialSort function sorts negative numbers in array1 in ascending order
```

## Talent Battle 100 Days Coding Series

// and positive numbers and zero in descending order

```
void swap(int *x, int *y)
```

```
{
```

```
    int temp = *x;
```

```
    *x = *y;
```

```
    *y = temp;
```

```
}
```

```
void SpecialSort(int *vec1,int n)
```

```
{
```

```
    for(int i = 0 ; i < n-1 ; i++)
```

```
    {
```

```
        for(int k = 0 ; k < n-1-i ; k++)
```

```
        {
```

```
            if(vec1[k] < vec1[k+1])
```

```
            {
```

```
                swap(&vec1[k],&vec1[k+1]);
```

```
            }
```

```
        }
```

```
    }
```

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

```
    {
```

```
        printf("%d ",vec1[i]);
```

```
    }
```

```
    printf("\n");
```

```
    int idx=0;
```

## Talent Battle 100 Days Coding Series

```
while(vec1[idx] >=0)
{
    idx++;
}
int start = idx,end = n-1;
while(start<end)
{
    swap(&vec1[start],&vec1[end]);
    start++;end--;
}
for(int i = 0 ; i<n ; i++)
{
    printf("%d ",vec1[i]);
}
printf("\n\n");
}
```

// Find min product and move the elements to left side of both arrays

```
int MaximumScalarProduct(int *vec1,int *vec2,int n)
{
    int max,sop=0,id1,id2;
    for(int i = 0 ; i<n ; i++)
    {
        max = INT_MIN;
        for(int j = i ; j<n ; j++)
```

## Talent Battle 100 Days Coding Series

```
{
    if((vec1[i]*vec2[j]) > max)
    {
        max = vec1[i]*vec2[j];
        id1 = i; id2 = j;
    }
}

sop = sop + max;
swap(&vec1[i],&vec1[id1]);
swap(&vec2[i],&vec2[id2]);
for(int i = 0 ; i<n ; i++)
{
    printf("%d ",vec1[i]);
}
printf("\n");
for(int i = 0 ; i<n ; i++)
{
    printf("%d ",vec2[i]);
}

printf("\n\n");
}

return sop;
}

int main()
{
```

## Talent Battle 100 Days Coding Series

```
int n;    scanf("%d",&n);

int vec1[n];

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

{

    scanf("%d",&vec1[i]);

}

int vec2[n];

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

{

    scanf("%d",&vec2[i]);


}

SpecialSort(vec1,n);

printf("%d",MaximumScalarProduct(vec1,vec2,n));

return 0;

}
```

The logo for TalentBattle is a shield-shaped emblem. It features a stylized figure in the center, possibly representing a warrior or a person in motion, rendered in a light blue or white color. The shield is divided into several colored sections: a purple top section, a green middle section, and a yellow bottom section. The word 'TalentBattle' is written in a large, light blue, sans-serif font across the bottom of the shield.

### C++

```
#include <bits/stdc++.h>
```

```
#include <limits.h>
```

```
using namespace std;
```

```
void swap(int *x, int *y)
```

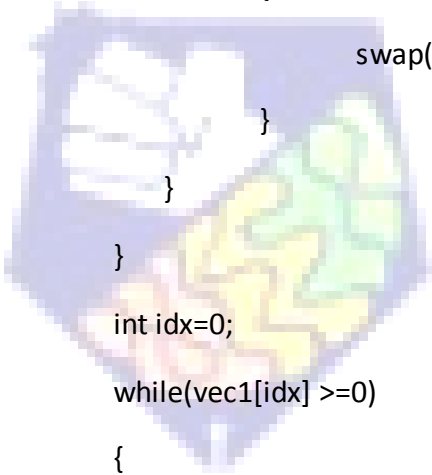
```
{
```

```
    int temp = *x;
```

```
    *x = *y;
```

## Talent Battle 100 Days Coding Series

```
*y = temp;
}
void SpecialSort(int *vec1,int n)
{
    for(int i = 0 ; i < n-1 ; i++)
    {
        for(int k = 0 ; k<n-1-i ; k++)
        {
            if(vec1[k] < vec1[k+1])
            {
                swap(&vec1[k],&vec1[k+1]);
            }
        }
    }
    int idx=0;
    while(vec1[idx] >=0)
    {
        idx++;
    }
    int start = idx,end = n-1;
    while(start<end)
    {
        swap(&vec1[start],&vec1[end]);
        start++;end--;
    }
}
```



TalentBattle

## Talent Battle 100 Days Coding Series

```
}

int MaximumScalarProduct(int *vec1,int *vec2,int n)
{
    int max,sop=0,id1,id2;
    for(int i = 0 ; i<n ; i++)
    {
        max = INT_MIN;
        for(int j = i ; j<n ; j++)
        {
            if((vec1[i]*vec2[j]) > max)
            {
                max = vec1[i]*vec2[j];
                id1 = i; id2 = j;
            }
        }
        sop = sop + max;
        swap(&vec1[i],&vec1[id1]);
        swap(&vec2[i],&vec2[id2]);
    }
    return sop;
}

int main()
{
    int n;    cin>>n;
```

## Talent Battle 100 Days Coding Series

```
int vec1[n];
for(int i = 0 ; i<n ; i++)
{
    cin>>vec1[i];
}

int vec2[n];
for(int i = 0 ; i<n ; i++)
{
    cin>>vec2[i];
}

SpecialSort(vec1,n);
cout<<MaximumScalarProduct(vec1,vec2,n);
return 0;
}
```

### JAVA

```
import java.util.Arrays;
import java.util.Collections;
import java.util.Scanner;

class Main
{
    public static void main (String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
```



## Talent Battle 100 Days Coding Series

```
int arr1[] = new int[n];
for(int i = 0 ; i<n ; i++)
{
    arr1[i] = sc.nextInt();
}

int arr2[] = new int[n];
for(int i = 0 ; i<n ; i++)
{
    arr2[i] = sc.nextInt();
}
```



```
Arrays.sort(arr1);
```

```
Arrays.sort(arr2);
```

```
int product = 0;
```

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

```
    product += arr1[i]*arr2[i];
```

```
System.out.print(product);
```

```
}
```

```
}
```

### Python

```
n = int(input())
```

```
arr1 = list(map(int,input().split(' ')))
```

## Talent Battle 100 Days Coding Series

```
arr2 = list(map(int,input().split(' ')))  
arr1.sort()  
arr2.sort()  
product = 0  
for i in range(n):  
    product += arr1[i]*arr2[i]  
  
print(product)
```



TalentBattle