

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

Sample input 1:

4

1 2 3 4

5 6 7 8

Sample output 1:

60

Explanation :

$$(4*5 + 3*6 + 2*7 + 1*8) = 60$$

Sample input 2:

4

-1 -2 -3 -4

5 6 -7 -8

Sample output 2:

-17

Explanation :

$$(-1*-8 + -2*-7 + -3*6 + -4*5) = -17$$

Solution:

C

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

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

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

```
// 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]);
```

```
            }
```

```
        }
```

```
    }
```

Talent Battle 100 Days Coding Series

```
for(int i = 0 ; i<n ; i++)
{
    printf("%d ",vec1[i]);
}
printf("\n");
int idx=0;
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 MinimumScalarProduct(int *vec1,int *vec2,int n)
```

Talent Battle 100 Days Coding Series

```
{
    int min,sop=0,id1,id2;
    for(int i = 0 ; i<n ; i++)
    {
        min = INT_MAX;
        for(int j = i ; j<n ; j++)
        {
            if((vec1[i]*vec2[j]) < min)
            {
                min = vec1[i]*vec2[j];
                id1 = i; id2 = j;
            }
        }
        sop = sop + min;
        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]);
    }
}
```

Talent Battle 100 Days Coding Series

```
printf("\n\n");
}
return sop;
}
int main()
{
    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",MinimumScalarProduct(vec1,vec2,n));
    return 0;
}
```

C++

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

```
using namespace std;
```

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

```
// and positive numbers and zero in descending order
```

```
void SpecialSort(int vec1[],int n)
```

```
{
```

```
    int idx=0;
```

```
    sort(vec1,vec1+n);
```

```
    while(vec1[idx] < 0)
```

```
    {
```

```
        idx++;
```

```
    }
```

```
    int start = idx,end = n-1;
```

```
    while(start<end)
```

```
    {
```

```
        swap(vec1[start],vec1[end]);
```

```
        start++;end--;
```

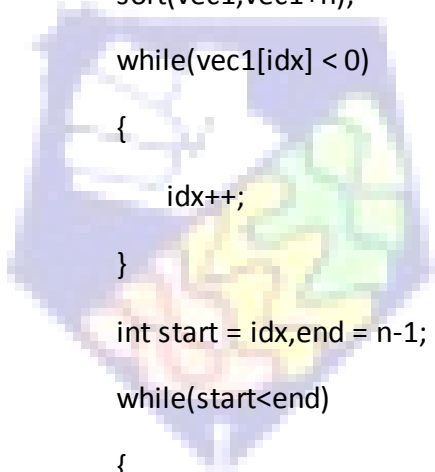
```
    }
```

```
}
```

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

```
int MinimumScalarProduct(int vec1[],int vec2[],int n)
```

```
{
```



TalentBattle

Talent Battle 100 Days Coding Series

```
int min,sop=0,id1,id2;
for(int i = 0 ; i<n ; i++)
{
    min = INT_MAX;
    for(int j = i ; j<n ; j++)
    {
        if((vec1[i]*vec2[j]) < min)
        {
            min = vec1[i]*vec2[j];
            id1 = i; id2 = j;
        }
    }
    sop = sop + min;
    swap(vec1[i],vec1[id1]);
    swap(vec2[i],vec2[id2]);
}
return sop;
}

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



TalentBattle

Talent Battle 100 Days Coding Series

```
}  
int vec2[n];  
for(int i = 0 ; i<n ; i++)  
{  
    cin>>vec2[i];  
}  
SpecialSort(vec1,n);  
cout<<MinimumScalarProduct(vec1,vec2,n);  
return 0;  
}
```



TalentBattle

Talent Battle 100 Days Coding Series

JAVA

```
import java.util.*;

import java.lang.*;

import java.io.*;

class Main

{

    static void swap(int arr[],int start, int end)

    {

        int temp = arr[start];

        arr[start] = arr[end];

        arr[end] = temp;

    }

    // SpecialSort function sorts negative numbers in array1 in ascending
    order
    // and positive numbers and zero in descending order
    static void SpecialSort(int vec1[],int n)

    {

        Arrays.sort(vec1);

        int idx=0;

        while((idx<n) && (vec1[idx] < 0))

        {

            idx++;

        }

        int start = idx,end = n-1;

        while(start<end)

        {
```

Talent Battle 100 Days Coding Series

```
        swap(vec1,start,end);;
        start++;end--;
    }
}

// Find min product and move the elements to left side of both arrays
static int MinimumScalarProduct(int vec1[], int vec2[], int n)
{
    int min,sop=0;
    int id1=0,id2=0;
    for(int i = 0 ; i<n ; i++)
    {
        min = Integer.MAX_VALUE;
        for(int j = i ; j<n ; j++)
        {
            if((vec1[i]*vec2[j]) < min)
            {
                min = vec1[i]*vec2[j];
                id1 = i; id2 = j;
            }
        }
        sop = sop + min;
        swap(vec1,i,id1);
        swap(vec2,i,id2);
    }
}
```

Talent Battle 100 Days Coding Series

```
        return sop;
    }

    public static void main(String[] args) throws java.lang.Exception
    {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int vec1[] = new int[n];
        for(int i = 0 ; i<n ; i++)
        {
            vec1[i] = sc.nextInt();
        }
        int vec2[] = new int[n];
        for(int i = 0 ; i<n ; i++)
        {
            vec2[i] = sc.nextInt();
        }
        SpecialSort(vec1,n);
        System.out.print(MinimumScalarProduct(vec1,vec2,n));
    }
}
```

Python

```
def swap(vec1, pos1, pos2):  
    vec1[pos1], vec1[pos2] = vec1[pos2], vec1[pos1]
```

```
def SpeecialSort(vec1,n):  
    vec1.sort()  
    idx=0  
    while idx<n and vec1[idx] < 0 :  
        idx=idx+1
```

```
    start = idx  
    end = n-1  
    while(start<end):  
        swap(vec1, start, end)  
        start = start + 1  
        end = end + 1
```

```
def MinimumScalarProduct(vec1,vec2,n):  
    sop=0  
    for i in range(0,n):  
        min = 2147483647  
        for j in range(i,n):  
            if(vec1[i]*vec2[j]) < min :  
                min = vec1[i]*vec2[j]  
            id1 = i
```

Talent Battle 100 Days Coding Series

```
        id2 = j
    sop = sop + min
    swap(vec1,i,id1)
    swap(vec2,i,id2)
return sop
```

```
n = int(input())
vec1 = list(map(int,input().split(' ')))
vec2 = list(map(int,input().split(' ')))

SpecialSort(vec1,n)
print(MinimumScalarProduct(vec1,vec2,n))
```



TalentBattle