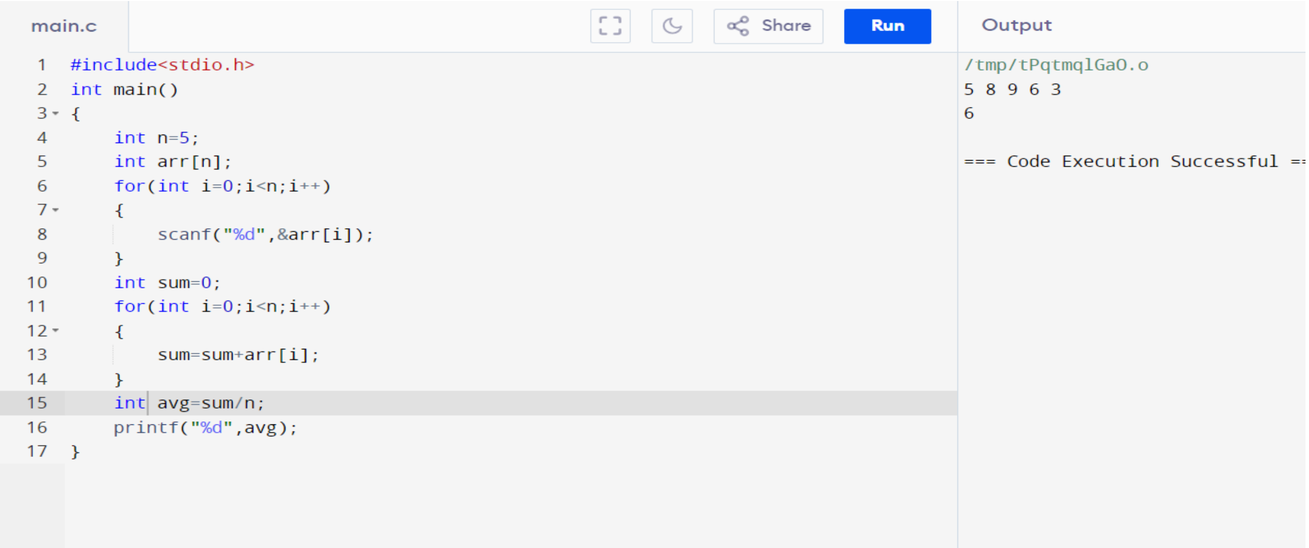
ARRAYS
LEVEL 3

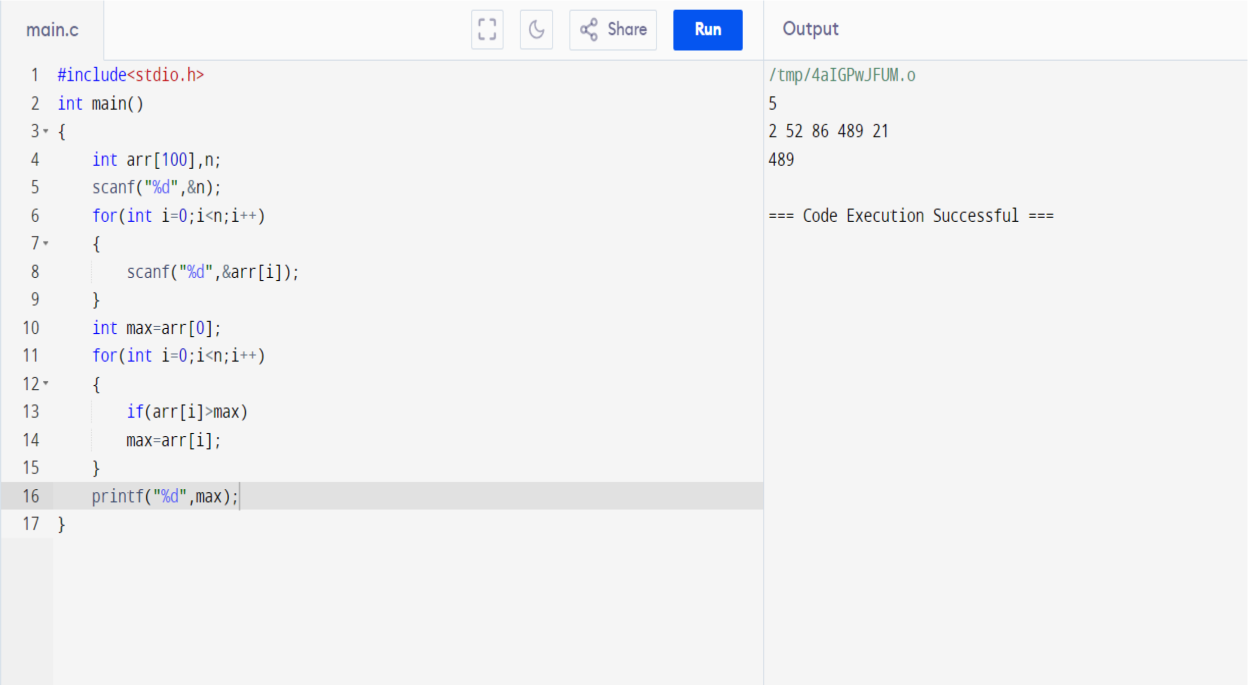
1.AVERAGE OF AN ARRAY



```c
#include<stdio.h>
int main()
{
    int n=5;
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int sum=0;
    for(int i=0;i<n;i++)
    {
        sum=sum+arr[i];
    }
    int avg=sum/n;
    printf("%d",avg);
}
```

```
/tmp/tPqtmqlGa0.o
5 8 9 6 3
6

=== Code Execution Successful ==
```

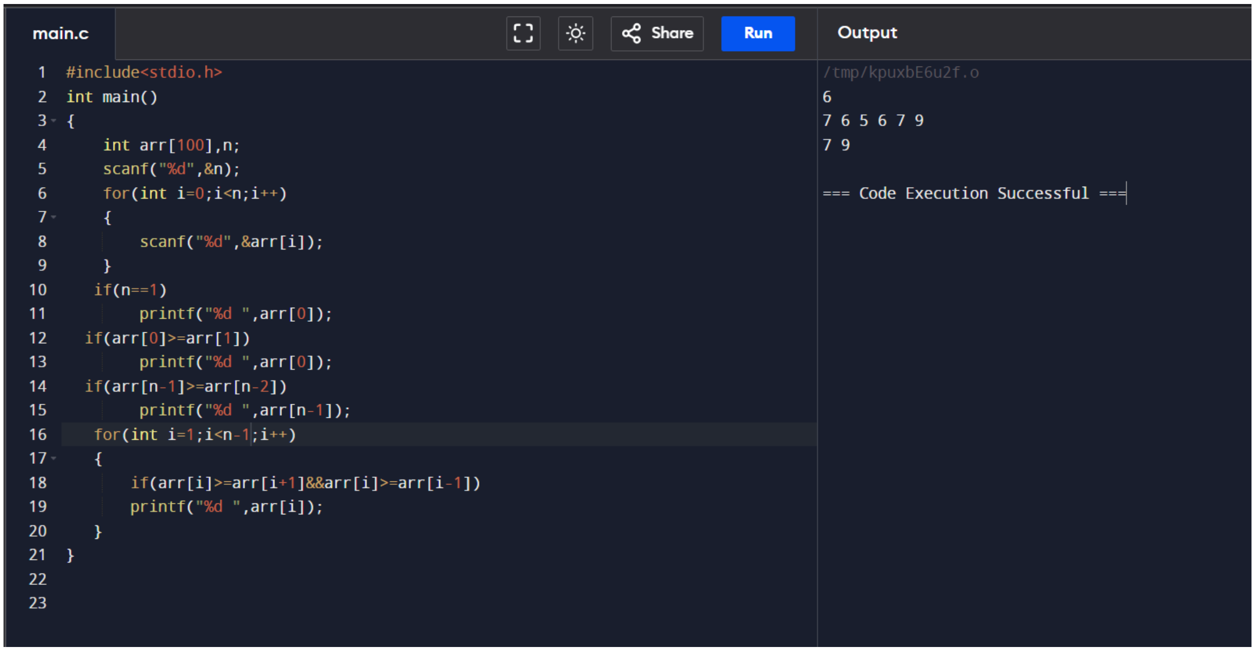
2.FINDING MAXIMUM / LARGEST NUMBER



```c
#include<stdio.h>
int main()
{
    int arr[100],n;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int max=arr[0];
    for(int i=0;i<n;i++)
    {
        if(arr[i]>max)
        max=arr[i];
    }
    printf("%d",max);
}
```

```
/tmp/4aIGPwJFUM.o
5
2 52 86 489 21
489

=== Code Execution Successful ===
```

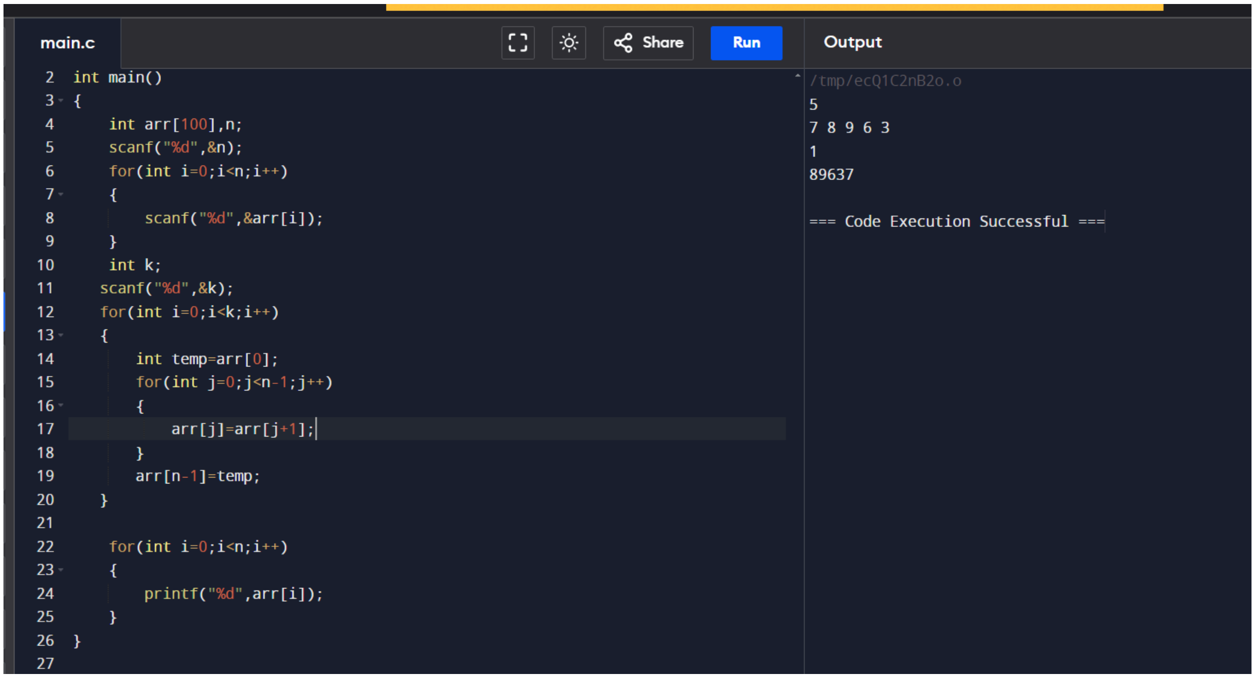
3.PEAK ELEMENT



```c
#include<stdio.h>
int main()
{
    int arr[100],n;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
   if(n==1)
        printf("%d ",arr[0]);
  if(arr[0]>=arr[1])
        printf("%d ",arr[0]);
  if(arr[n-1]>=arr[n-2])
        printf("%d ",arr[n-1]);
   for(int i=1;i<n-1;i++)
   {
       if(arr[i]>=arr[i+1]&&arr[i]>=arr[i-1])
       printf("%d ",arr[i]);
   }
}


```

```
/tmp/kpuxbE6u2f.o
6
7 6 5 6 7 9
7 9

=== Code Execution Successful ===
```

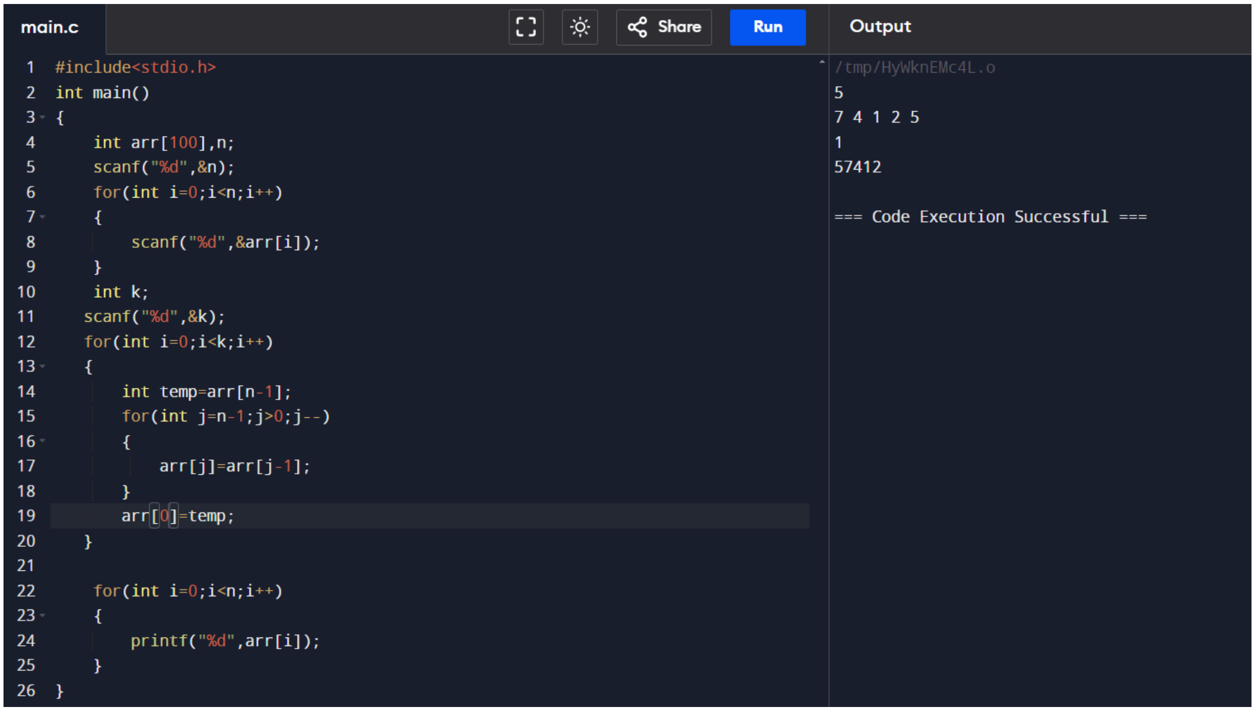
4.LEFT ROTATION



```c
int main()
{
    int arr[100],n;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int k;
   scanf("%d",&k);
   for(int i=0;i<k;i++)
   {
       int temp=arr[0];
       for(int j=0;j<n-1;j++)
       {
           arr[j]=arr[j+1];
       }
       arr[n-1]=temp;
   }

    for(int i=0;i<n;i++)
    {
        printf("%d",arr[i]);
    }
}

```

```
/tmp/ecQ1C2nB2o.o
5
7 8 9 6 3
1
89637

=== Code Execution Successful ===
```

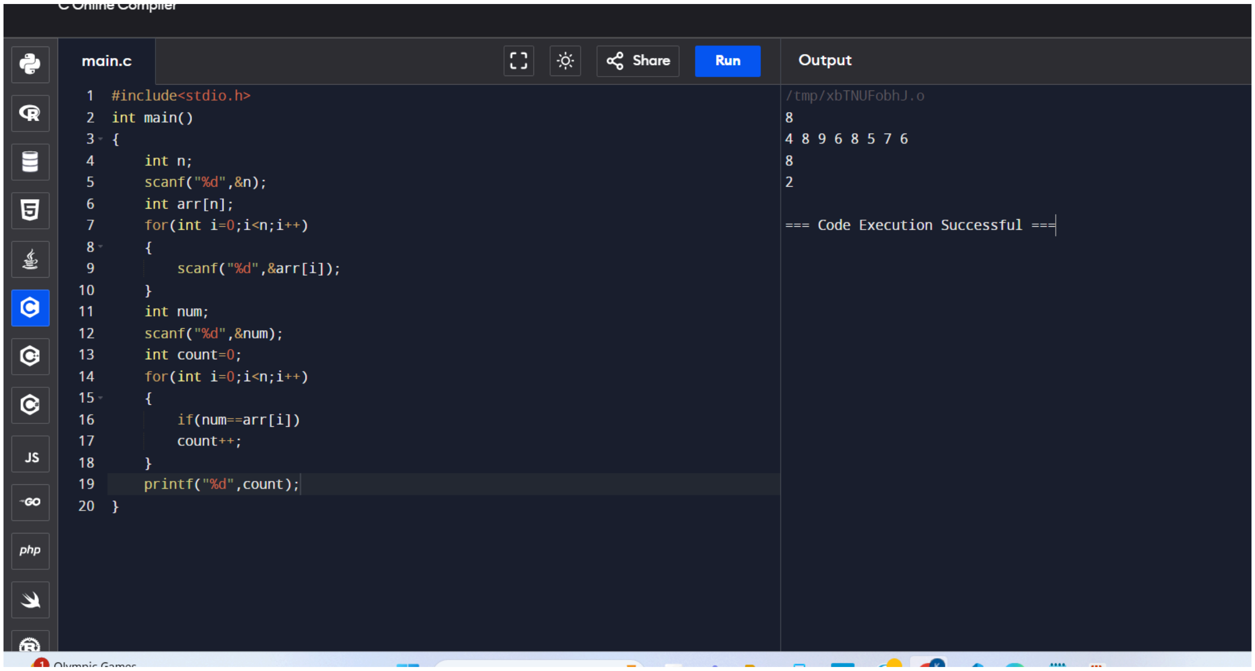
5.RIGHT ROTATION



```c
#include<stdio.h>
int main()
{
    int arr[100],n;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int k;
   scanf("%d",&k);
   for(int i=0;i<k;i++)
   {
       int temp=arr[n-1];
       for(int j=n-1;j>0;j--)
       {
           arr[j]=arr[j-1];
       }
       arr[0]=temp;
   }

    for(int i=0;i<n;i++)
    {
        printf("%d",arr[i]);
    }
}
```

```
/tmp/HyWknEMc4L.o
5
7 4 1 2 5
1
57412

=== Code Execution Successful ===
```

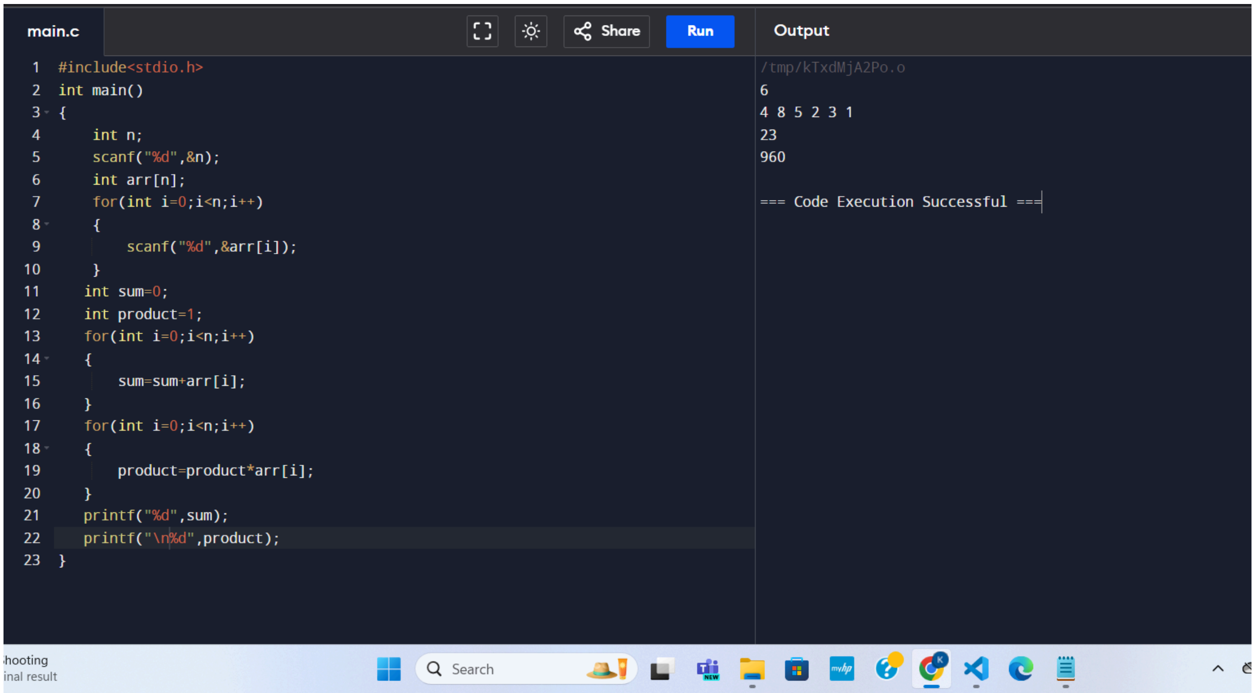
6.NUMBER OF OCCURENCE OF A NUMBER



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int num;
    scanf("%d",&num);
    int count=0;
    for(int i=0;i<n;i++)
    {
        if(num==arr[i])
        count++;
    }
    printf("%d",count);
}
```

```
/tmp/xbTNUFobhJ.o
8
4 8 9 6 8 5 7 6
8
2

=== Code Execution Successful ===
```

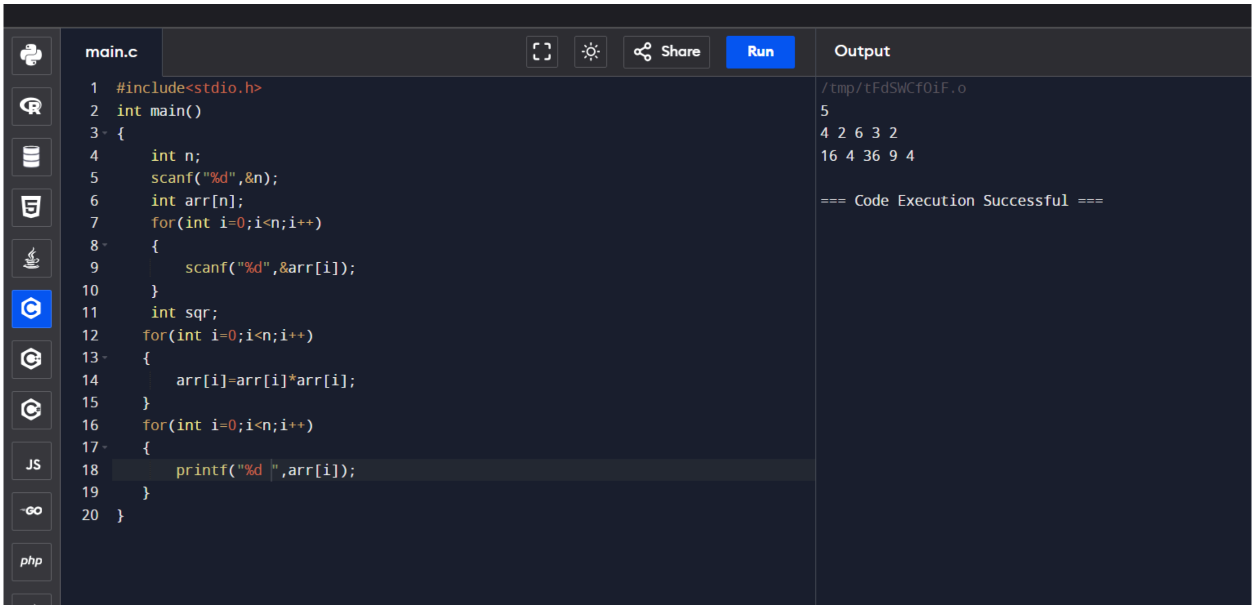
7.SUM AND PRODUCT OF A NUMBER



```c
#include<stdio.h>
int main()
{
     int n;
     scanf("%d",&n);
     int arr[n];
     for(int i=0;i<n;i++)
     {
         scanf("%d",&arr[i]);
     }
    int sum=0;
    int product=1;
    for(int i=0;i<n;i++)
    {
        sum=sum+arr[i];
    }
    for(int i=0;i<n;i++)
    {
        product=product*arr[i];
    }
    printf("%d",sum);
    printf("\n%d",product);
}
```

```
/tmp/kTxdMjA2Po.o
6
4 8 5 2 3 1
23
960

=== Code Execution Successful ===
```

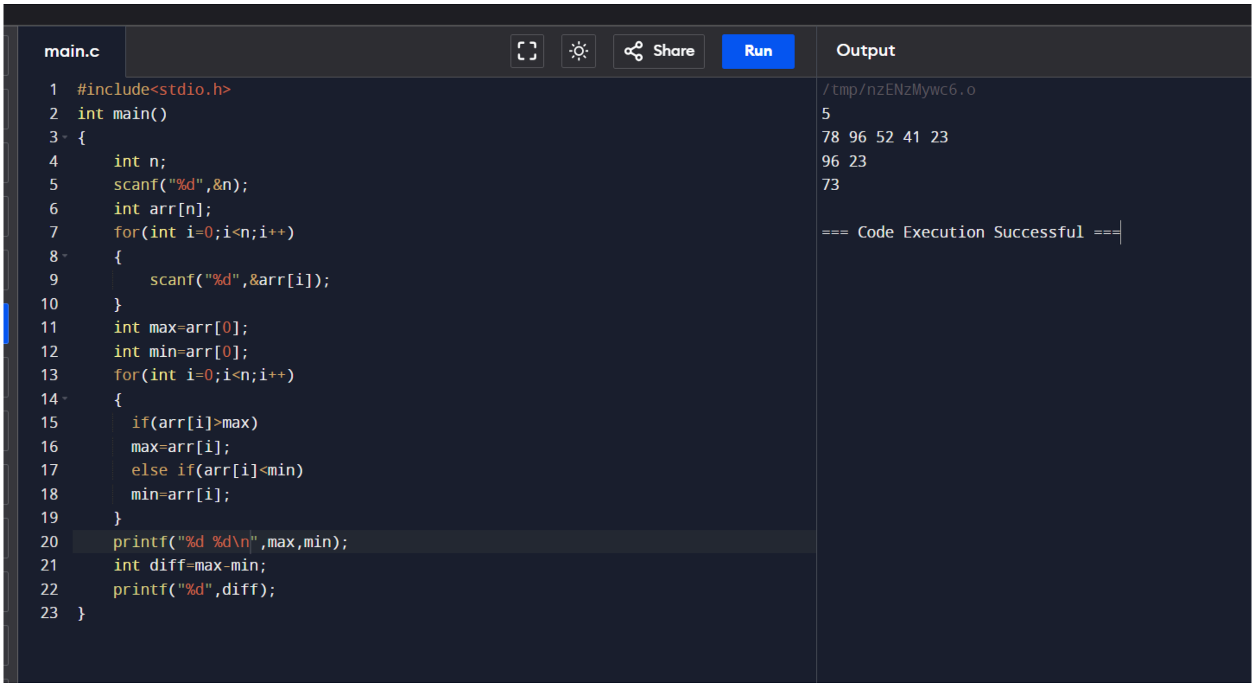
8.SQUARE OF ARRAY ELEMENTS



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int sqr;
   for(int i=0;i<n;i++)
   {
      arr[i]=arr[i]*arr[i];
   }
   for(int i=0;i<n;i++)
   {
      printf("%d ",arr[i]);
   }
}
```

```
/tmp/tFdSWCf0iF.o
5
4 2 6 3 2
16 4 36 9 4

=== Code Execution Successful ===
```

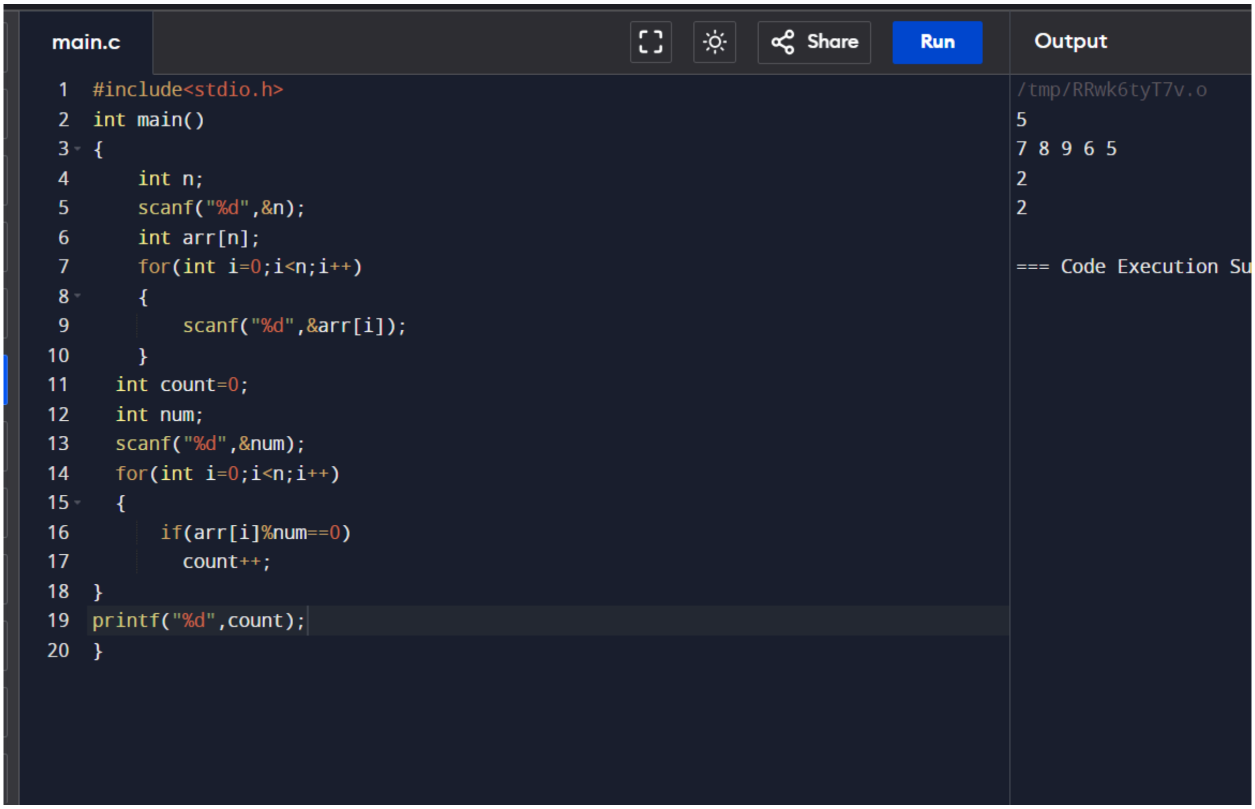
9.DIFFERENCE BETWEEN MAXIMUM AND MINIMUM NUMBER



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int max=arr[0];
    int min=arr[0];
    for(int i=0;i<n;i++)
    {
      if(arr[i]>max)
      max=arr[i];
      else if(arr[i]<min)
      min=arr[i];
    }
    printf("%d %d\n",max,min);
    int diff=max-min;
    printf("%d",diff);
}
```

```
/tmp/nzENzMywc6.o
5
78 96 52 41 23
96 23
73

=== Code Execution Successful ===
```

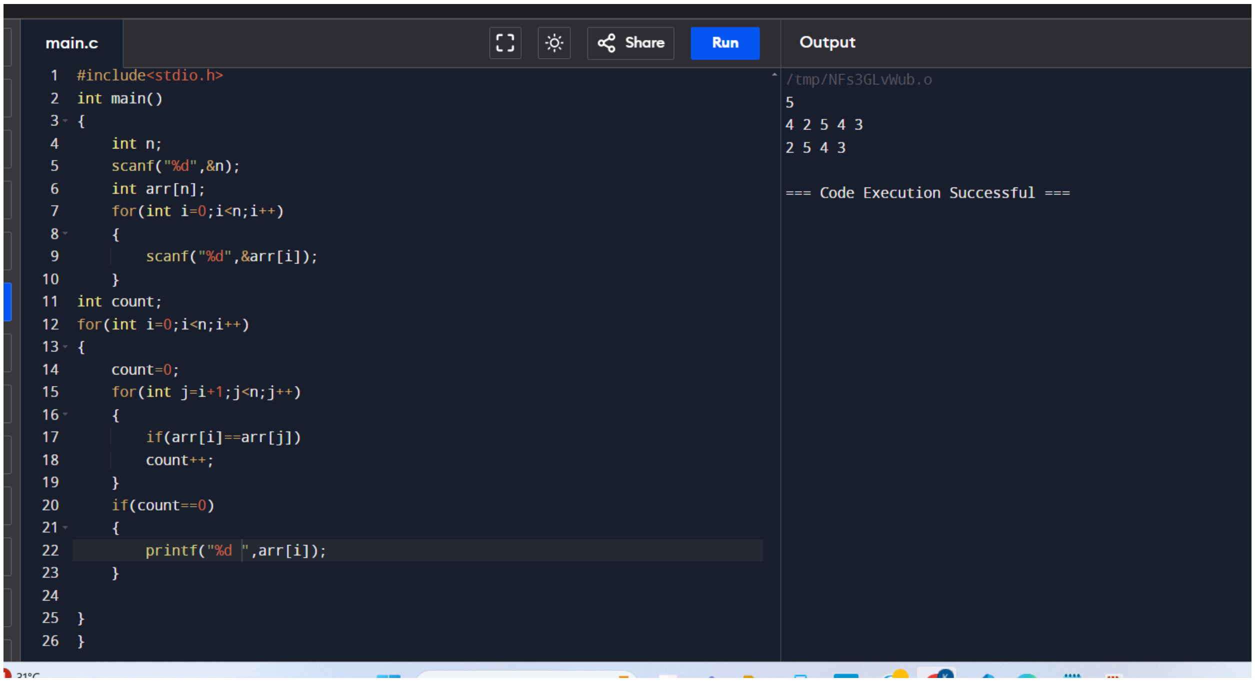
10.NO OF ELEMENTS DIVISIBLE BY A NUMBER



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
  int count=0;
  int num;
  scanf("%d",&num);
  for(int i=0;i<n;i++)
  {
      if(arr[i]%num==0)
        count++;
}
printf("%d",count);
}
```

```
/tmp/RRwk6tyT7v.o
5
7 8 9 6 5
2
2


=== Code Execution Su
```

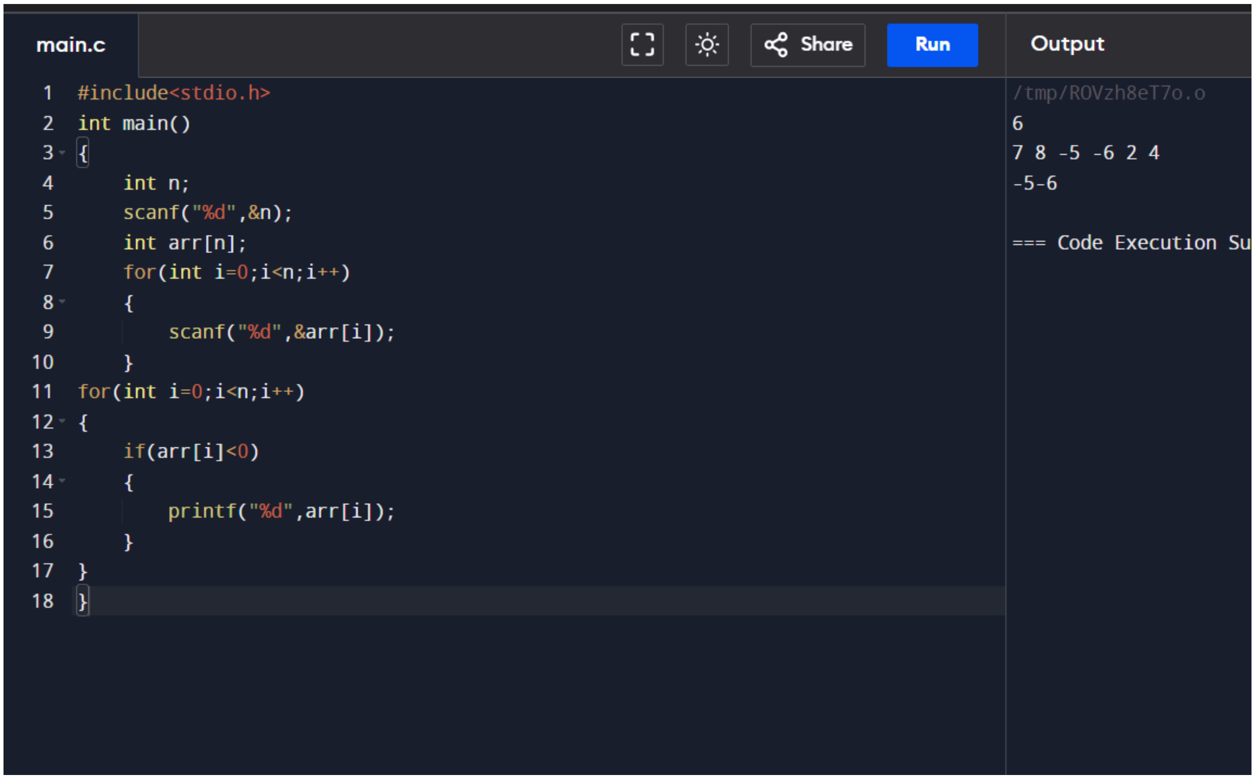
11.REMOVING DUPLICATE ELEMENTS



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
int count;
for(int i=0;i<n;i++)
{
    count=0;
    for(int j=i+1;j<n;j++)
    {
        if(arr[i]==arr[j])
        count++;
    }
    if(count==0)
    {
        printf("%d ",arr[i]);
    }

}
}
```

```
/tmp/NFs3GLvWub.o
5
4 2 5 4 3
2 5 4 3

=== Code Execution Successful ===
```

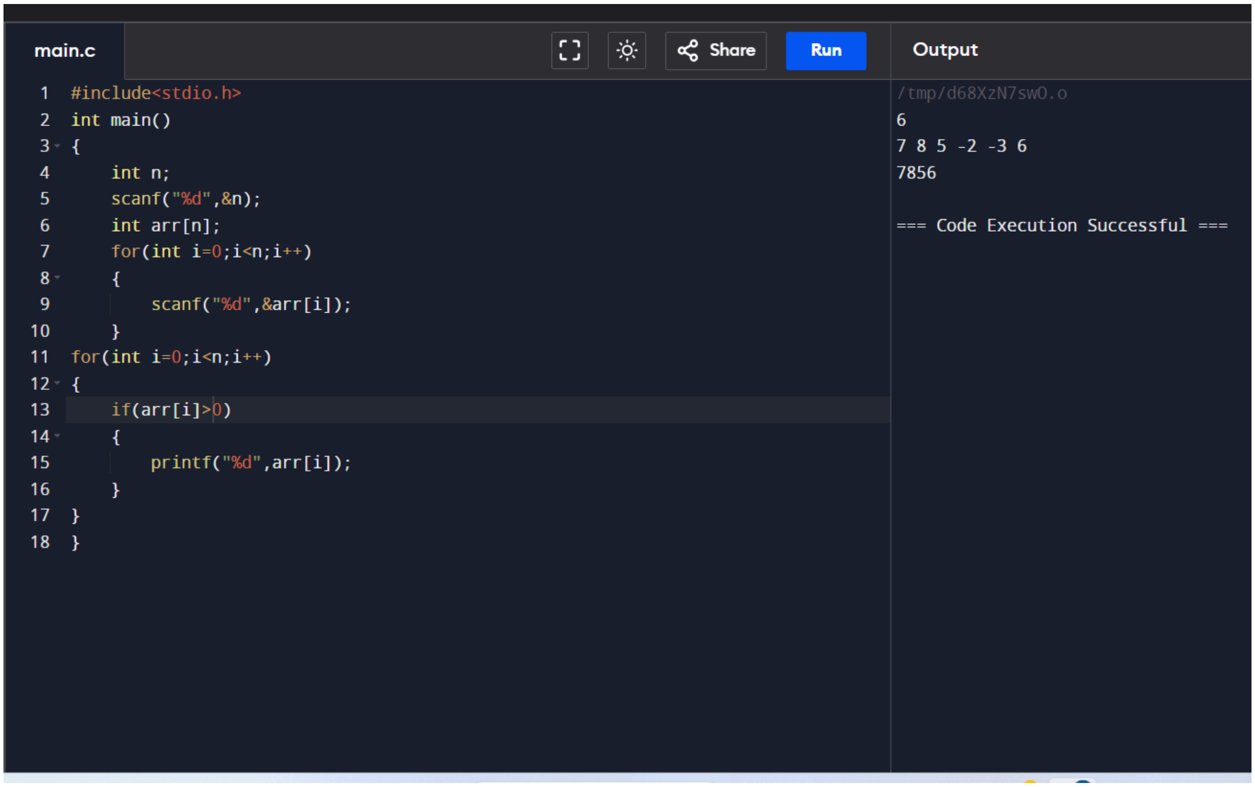
12. NEGATIVE ELEMENTS



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
for(int i=0;i<n;i++)
{
    if(arr[i]<0)
    {
        printf("%d",arr[i]);
    }
}
}
```

```
/tmp/ROVzh8eT7o.o
6
7 8 -5 -6 2 4
-5-6

=== Code Execution Su
```

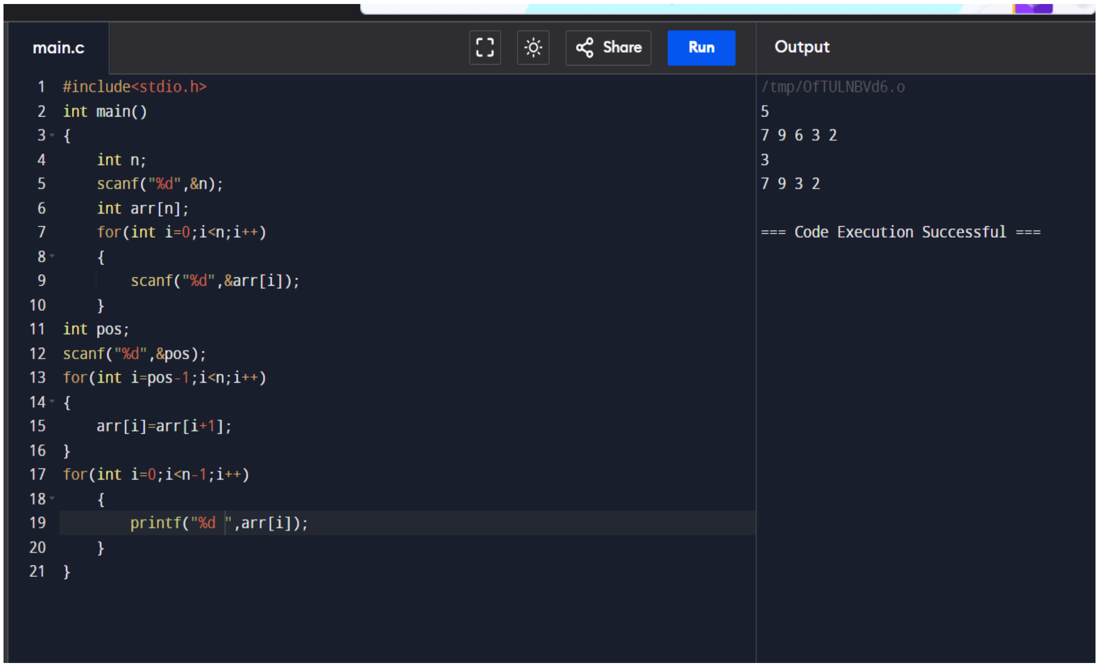
13.POSITIVE ELEMENTS



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
for(int i=0;i<n;i++)
{
    if(arr[i]>0)
    {
        printf("%d",arr[i]);
    }
}
}
```

```
/tmp/d68XzN7sw0.o
6
7 8 5 -2 -3 6
7856

=== Code Execution Successful ===
```

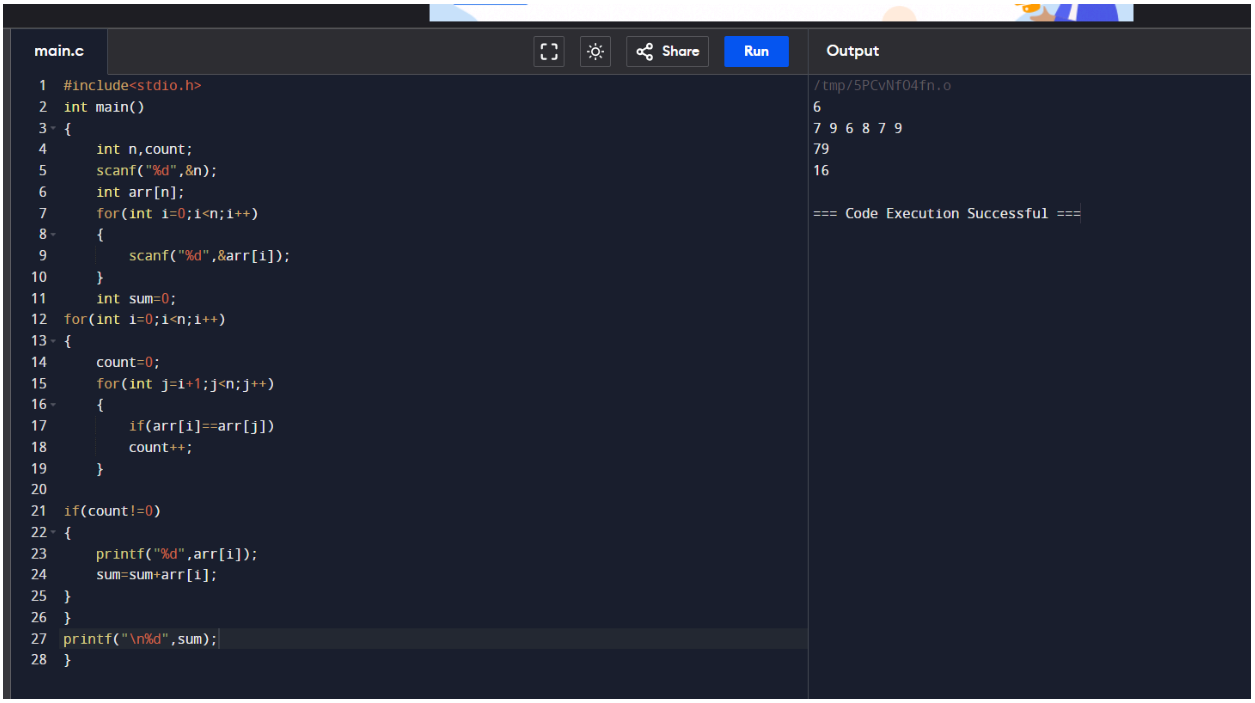
14.DELETING AN ELEMENT AT A POSITION IN A ARRAY



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
int pos;
scanf("%d",&pos);
for(int i=pos-1;i<n;i++)
{
    arr[i]=arr[i+1];
}
for(int i=0;i<n-1;i++)
    {
        printf("%d ",arr[i]);
    }
}
```

```
/tmp/OfTULNBVd6.o
5
7 9 6 3 2
3
7 9 3 2

=== Code Execution Successful ===
```

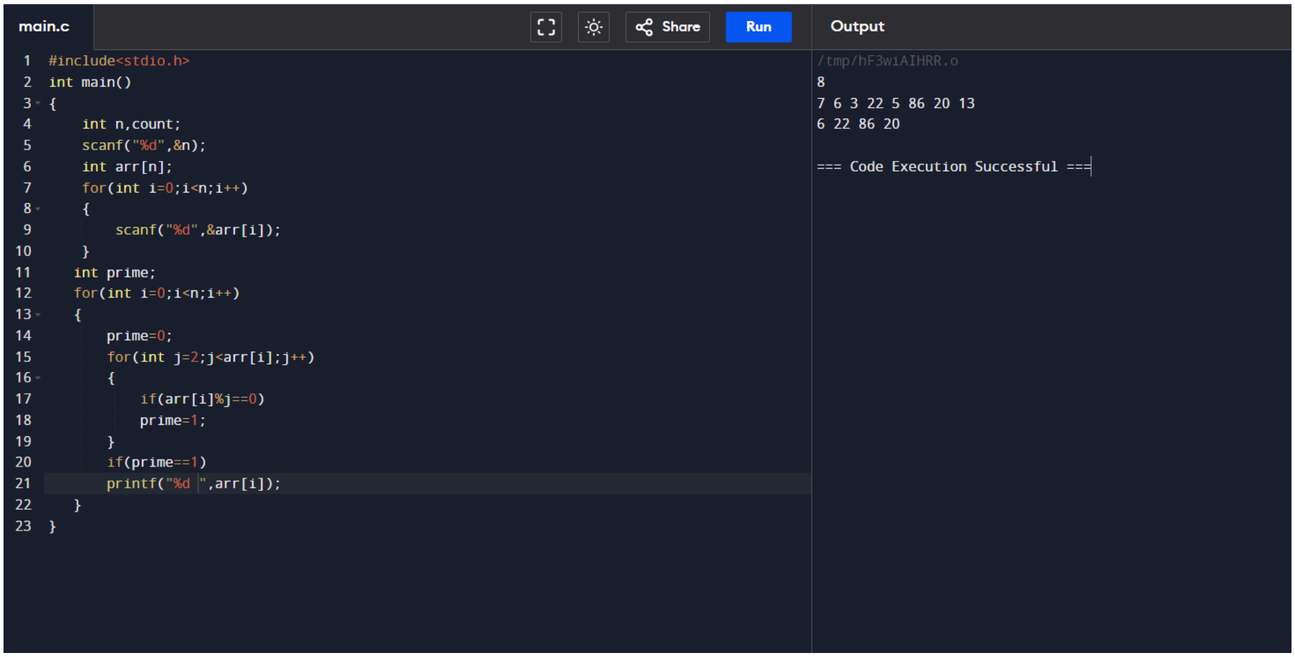
15.SUM OF DUPLICATE ELEMENTS



```
#include<stdio.h>
int main()
{
    int n,count;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int sum=0;
for(int i=0;i<n;i++)
{
    count=0;
    for(int j=i+1;j<n;j++)
    {
        if(arr[i]==arr[j])
        count++;
    }

if(count!=0)
{
    printf("%d",arr[i]);
    sum=sum+arr[i];
}
}
printf("\n%d",sum);
}
```

```
/tmp/5PCvNfO4fn.o
6
7 9 6 8 7 9
79
16

=== Code Execution Successful ===
```

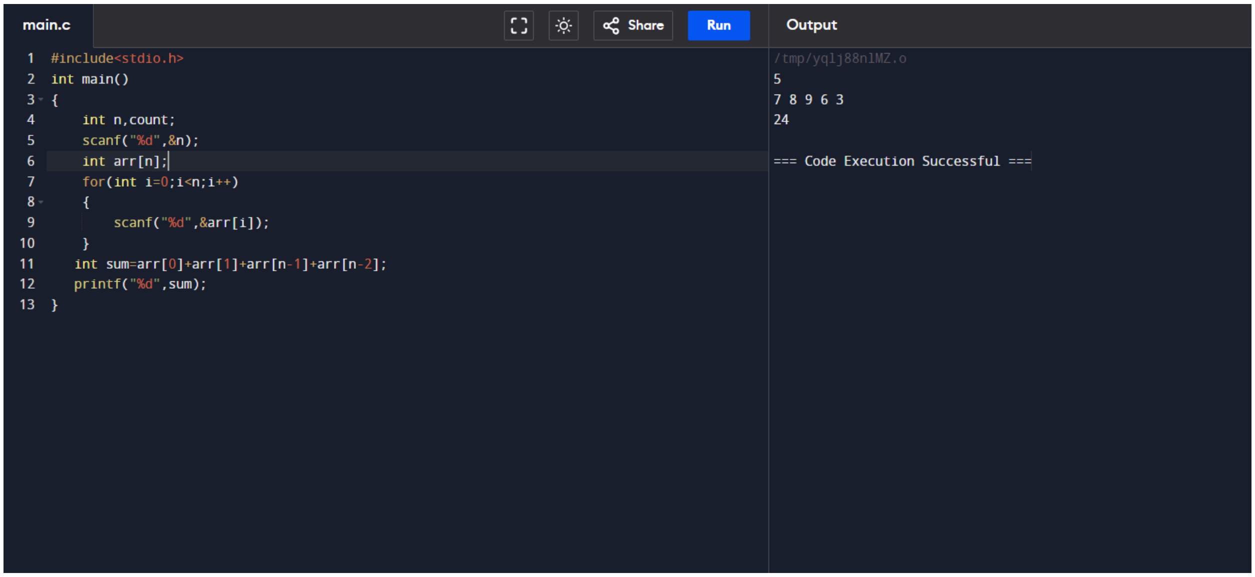
16. NON PRIME NUMBERS



```
#include<stdio.h>
int main()
{
    int n,count;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
   int prime;
   for(int i=0;i<n;i++)
   {
       prime=0;
       for(int j=2;j<arr[i];j++)
       {
           if(arr[i]%j==0)
           prime=1;
       }
       if(prime==1)
       printf("%d ",arr[i]);
   }
}
```

```
/tmp/hF3wiAIHRR.o
8
7 6 3 22 5 86 20 13
6 22 86 20

=== Code Execution Successful ===
```

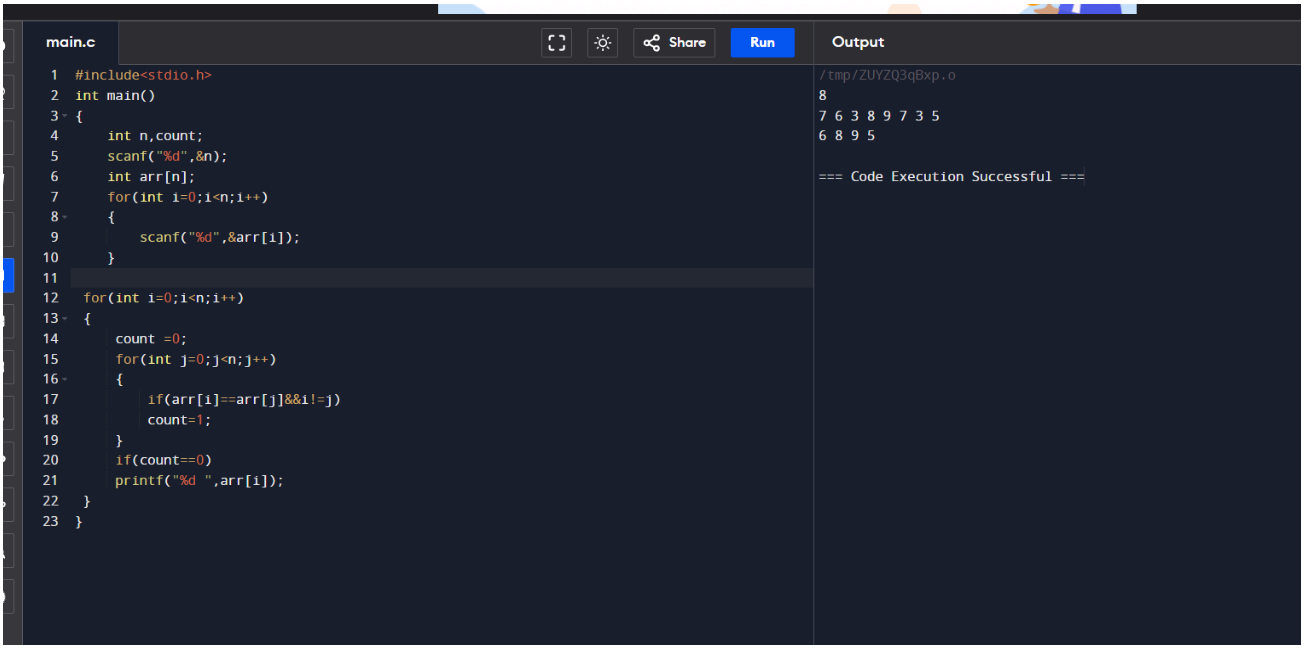
17.SUM OF 1ST ,2ND , LAST AND LAST BEFORE NUMBER



```c
#include<stdio.h>
int main()
{
    int n,count;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
   int sum=arr[0]+arr[1]+arr[n-1]+arr[n-2];
   printf("%d",sum);
}
```

```
/tmp/yqlj88n1MZ.o
5
7 8 9 6 3
24

=== Code Execution Successful ===
```

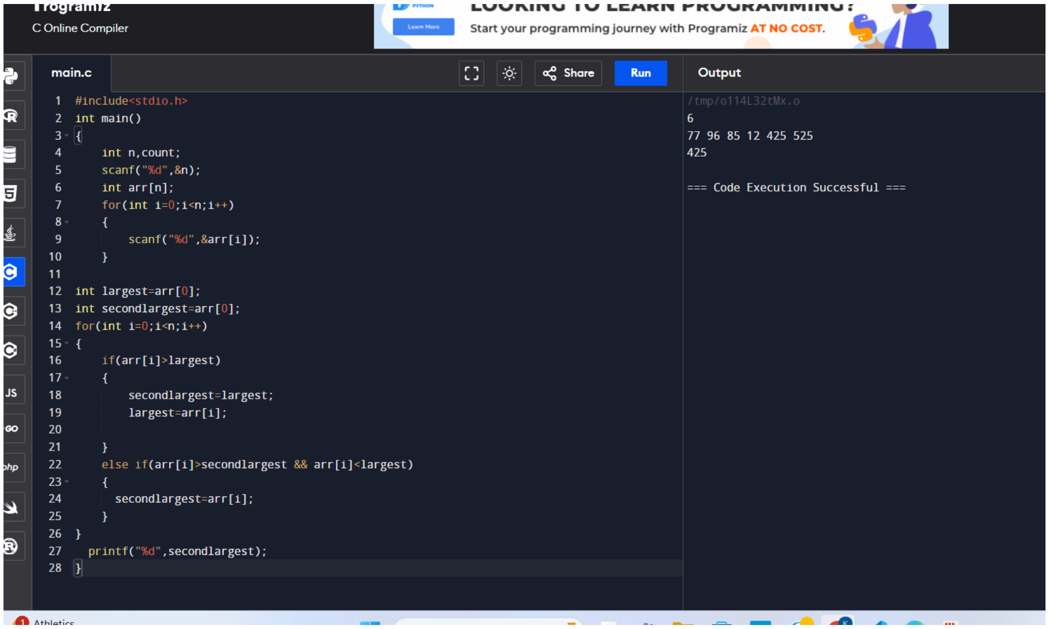
18.PRINTING UNIQUE NUMBERS



```c
#include<stdio.h>
int main()
{
    int n,count;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }

 for(int i=0;i<n;i++)
 {
     count =0;
     for(int j=0;j<n;j++)
     {
         if(arr[i]==arr[j]&&i!=j)
         count=1;
     }
     if(count==0)
     printf("%d ",arr[i]);
 }
}
```

```
/tmp/ZUYZQ3qBxp.o
8
7 6 3 8 9 7 3 5
6 8 9 5

=== Code Execution Successful ===
```

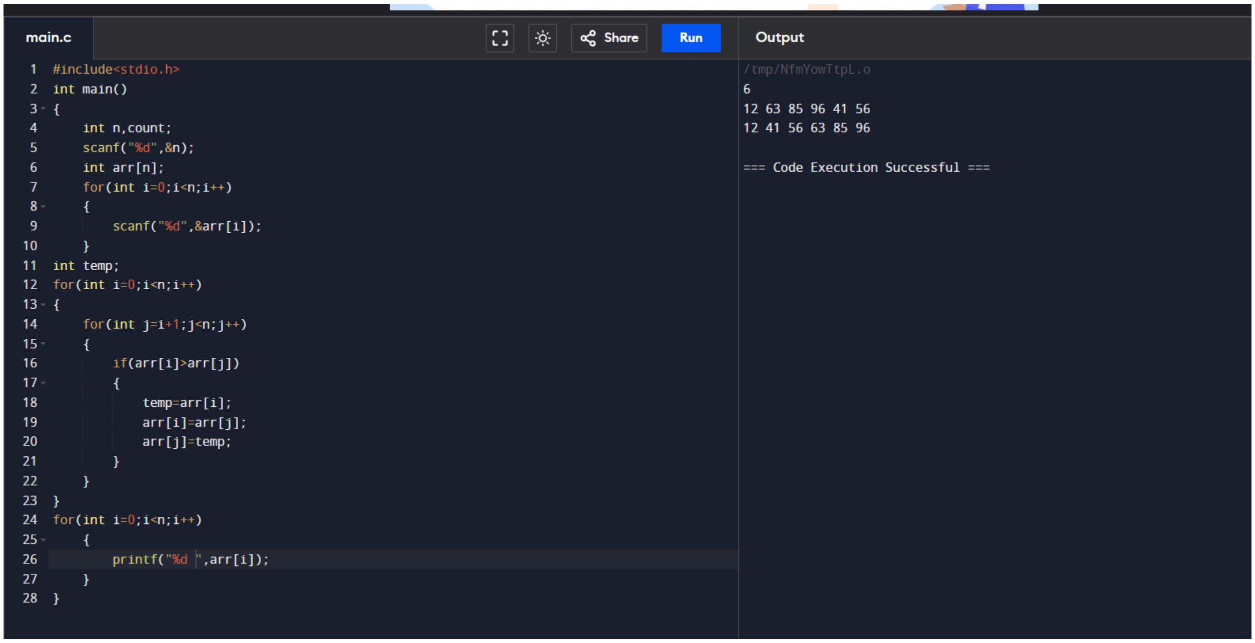
19. Second largest number


Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
main.c
Share
Run
Output

```c
#include<stdio.h>
int main()
{
    int n,count;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }

int largest=arr[0];
int secondlargest=arr[0];
for(int i=0;i<n;i++)
{
    if(arr[i]>largest)
    {
        secondlargest=largest;
        largest=arr[i];

    }
    else if(arr[i]>secondlargest && arr[i]<largest)
    {
      secondlargest=arr[i];
    }
}
  printf("%d",secondlargest);
}
```

```
/tmp/o114L32tMx.o
6
77 96 85 12 425 525
425

=== Code Execution Successful ===
```

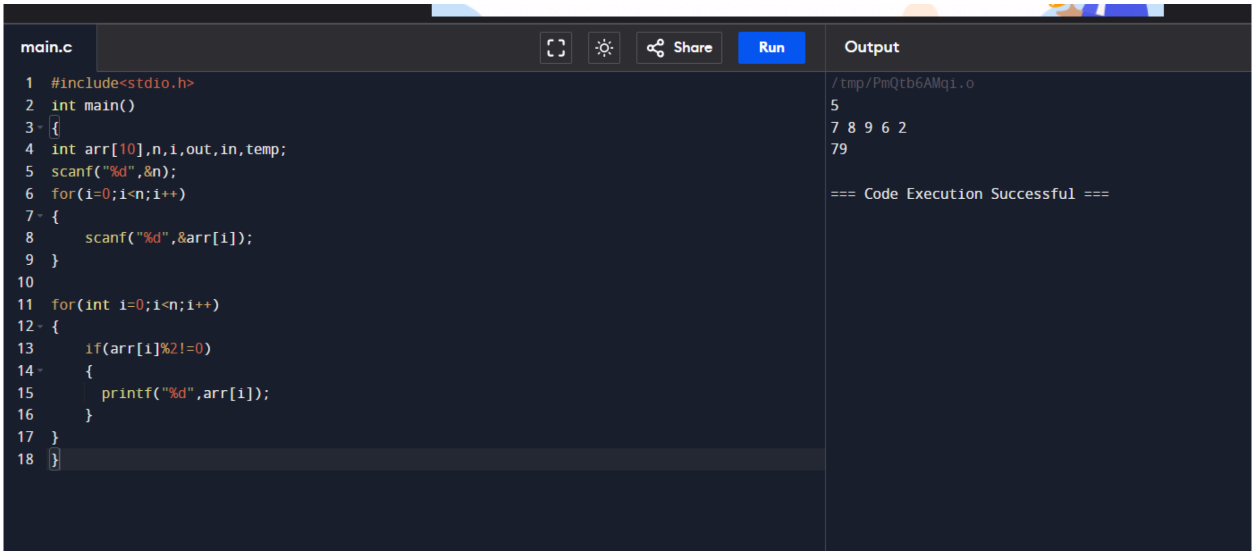
20.ASCENDING ORDER



```c
#include<stdio.h>
int main()
{
    int n,count;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
int temp;
for(int i=0;i<n;i++)
{
    for(int j=i+1;j<n;j++)
    {
        if(arr[i]>arr[j])
        {
            temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
        }
    }
}
for(int i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
}
```

```
/tmp/NfmYowTtpL.o
6
12 63 85 96 41 56
12 41 56 63 85 96

=== Code Execution Successful ===
```

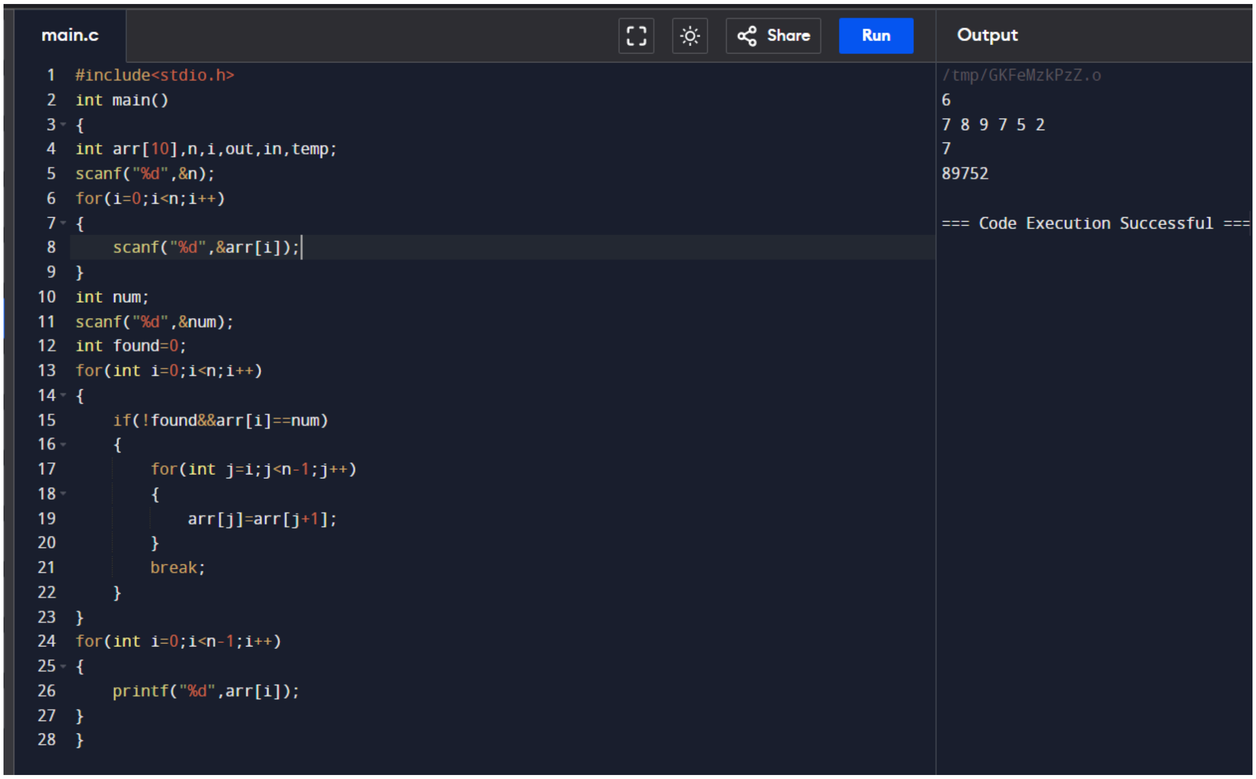
21.REMOVING EVEN NUMBERS



```c
#include<stdio.h>
int main()
{
int arr[10],n,i,out,in,temp;
scanf("%d",&n);
for(i=0;i<n;i++)
{
    scanf("%d",&arr[i]);
}

for(int i=0;i<n;i++)
{
    if(arr[i]%2!=0)
    {
      printf("%d",arr[i]);
    }
}
}
```

```
/tmp/PmQtb6AMqi.o
5
7 8 9 6 2
79

=== Code Execution Successful ===
```

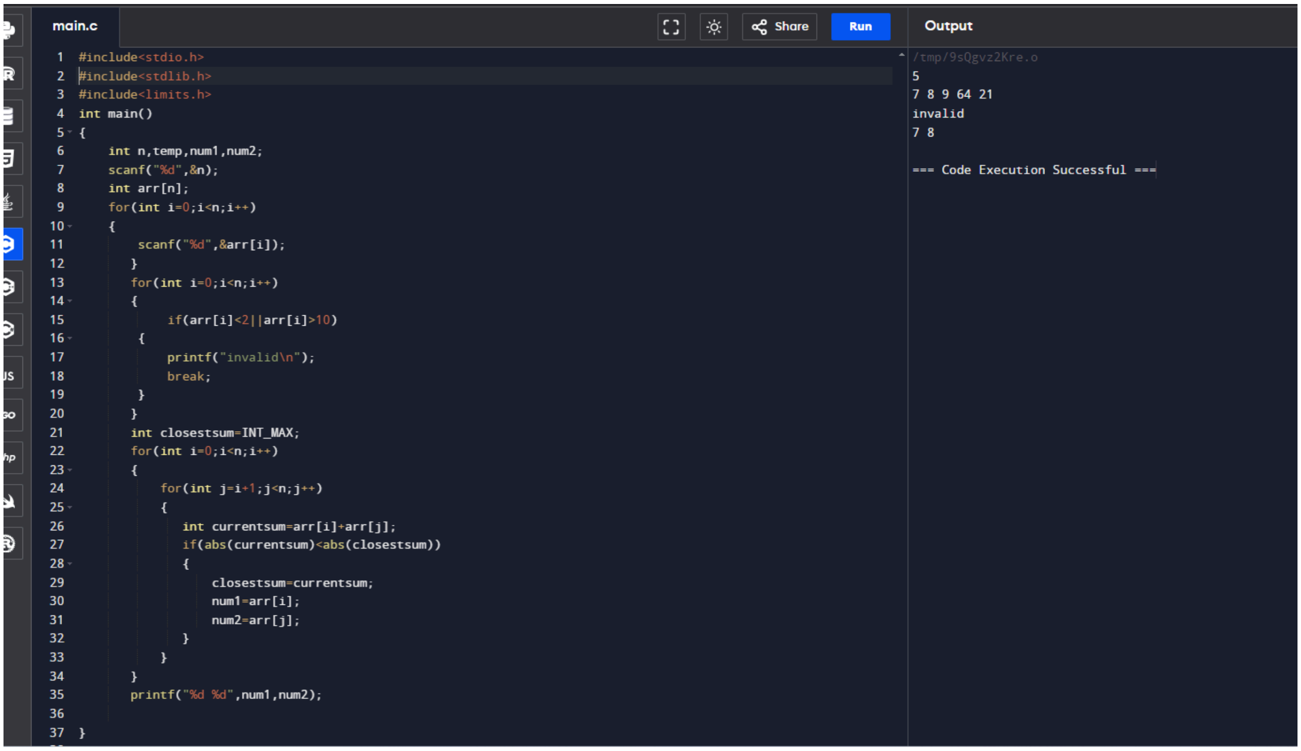
22.REMOVING FIRST OCCURRENCE OF A NUMBER



```c
#include<stdio.h>
int main()
{
int arr[10],n,i,out,in,temp;
scanf("%d",&n);
for(i=0;i<n;i++)
{
    scanf("%d",&arr[i]);
}
int num;
scanf("%d",&num);
int found=0;
for(int i=0;i<n;i++)
{
    if(!found&&arr[i]==num)
    {
        for(int j=i;j<n-1;j++)
        {
            arr[j]=arr[j+1];
        }
        break;
    }
}
for(int i=0;i<n-1;i++)
{
    printf("%d",arr[i]);
}
}
```

```
/tmp/GKFeMzkPzZ.o
6
7 8 9 7 5 2
7
89752

=== Code Execution Successful ===
```

23. Array Input range 2-10 if exceeds print "Invalid". print two elements which is the sum of two elements closest to 0(zero). Input: [-1,-10,8,2] output: [-1,2]. Explanation -1+2=-1 is the closest to 0 than other combinations.

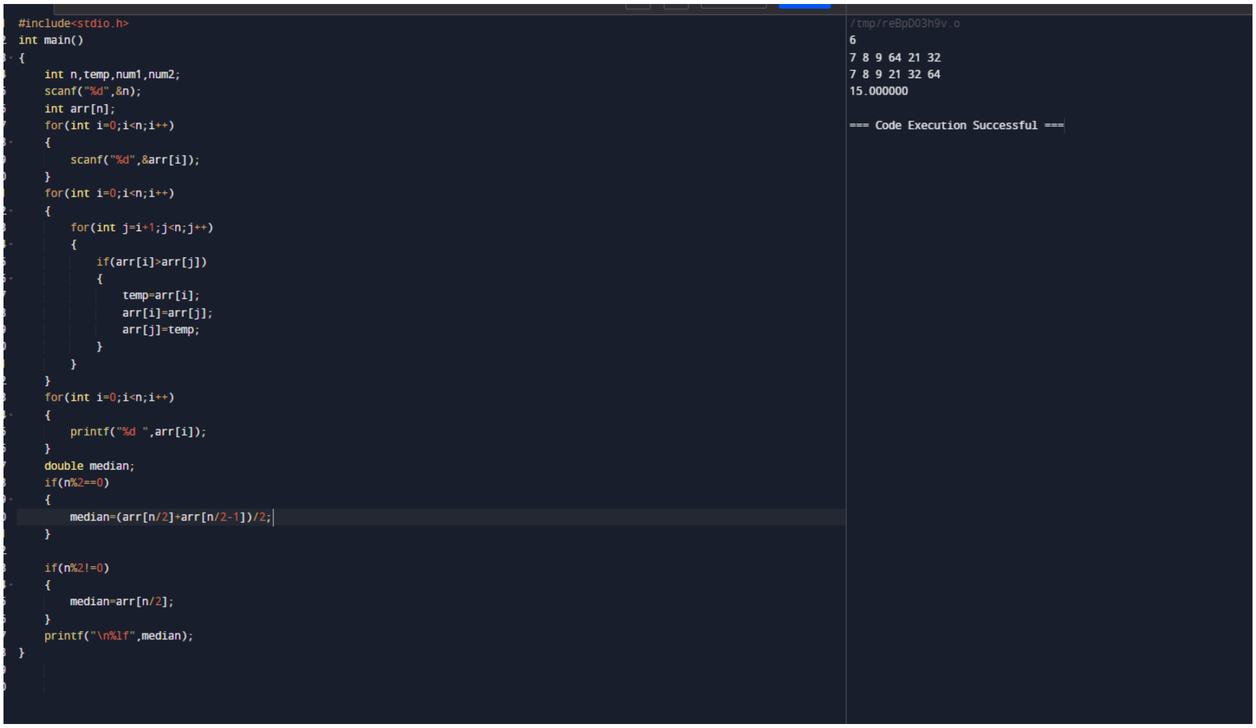


```c
#include<stdio.h>
#include<stdlib.h>
#include<limits.h>
int main()
{
    int n,temp,num1,num2;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
       }
       for(int i=0;i<n;i++)
       {
            if(arr[i]<2||arr[i]>10)
        {
            printf("invalid\n");
            break;
        }
       }
       int closestsum=INT_MAX;
       for(int i=0;i<n;i++)
       {
           for(int j=i+1;j<n;j++)
           {
              int currentsum=arr[i]+arr[j];
              if(abs(currentsum)<abs(closestsum))
              {
                  closestsum=currentsum;
                  num1=arr[i];
                  num2=arr[j];
              }
           }
       }
       printf("%d %d",num1,num2);

}
```

```
/tmp/9sQgvz2Kre.o
5
7 8 9 64 21
invalid
7 8

=== Code Execution Successful ===
```

2

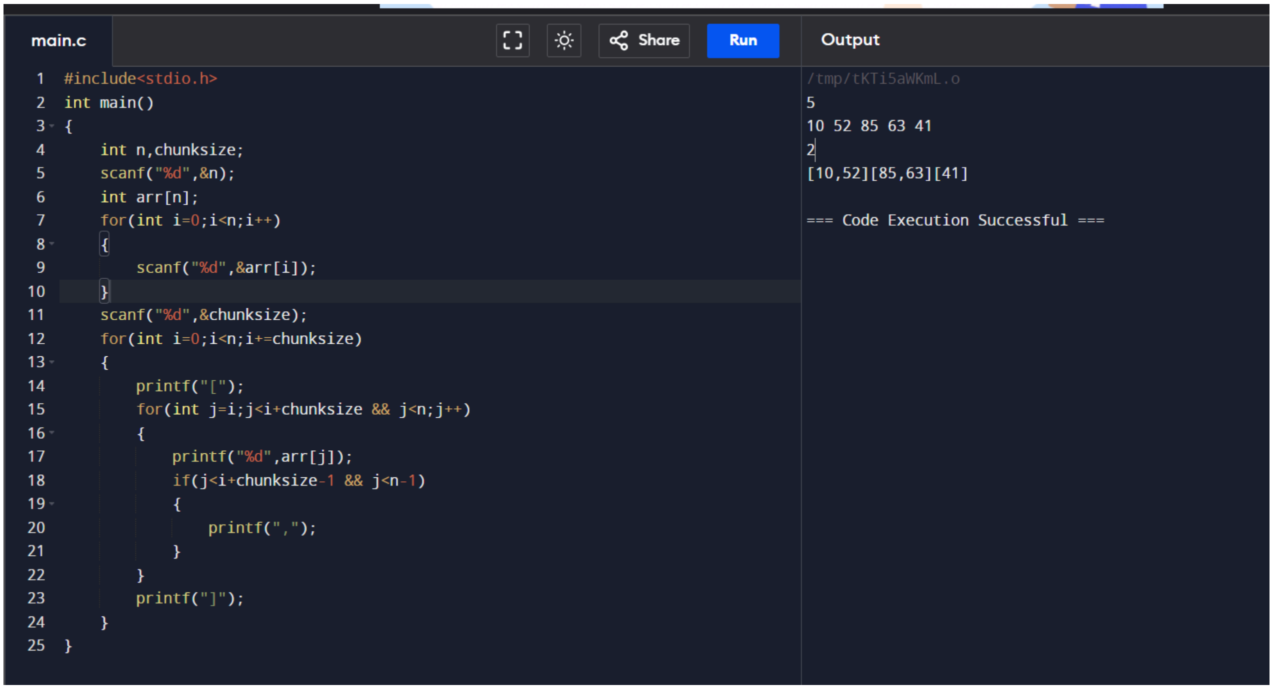
24.MEDIAN



```c
#include<stdio.h>
int main()
{
    int n,temp,num1,num2;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(arr[i]>arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    for(int i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
    double median;
    if(n%2==0)
    {
        median=(arr[n/2]+arr[n/2-1])/2;
    }

    if(n%2!=0)
    {
        median=arr[n/2];
    }
    printf("\n%lf",median);
}
```

```
/tmp/reBpD03h9v.o
6
7 8 9 64 21 32
7 8 9 21 32 64
15.000000

=== Code Execution Successful ===
```

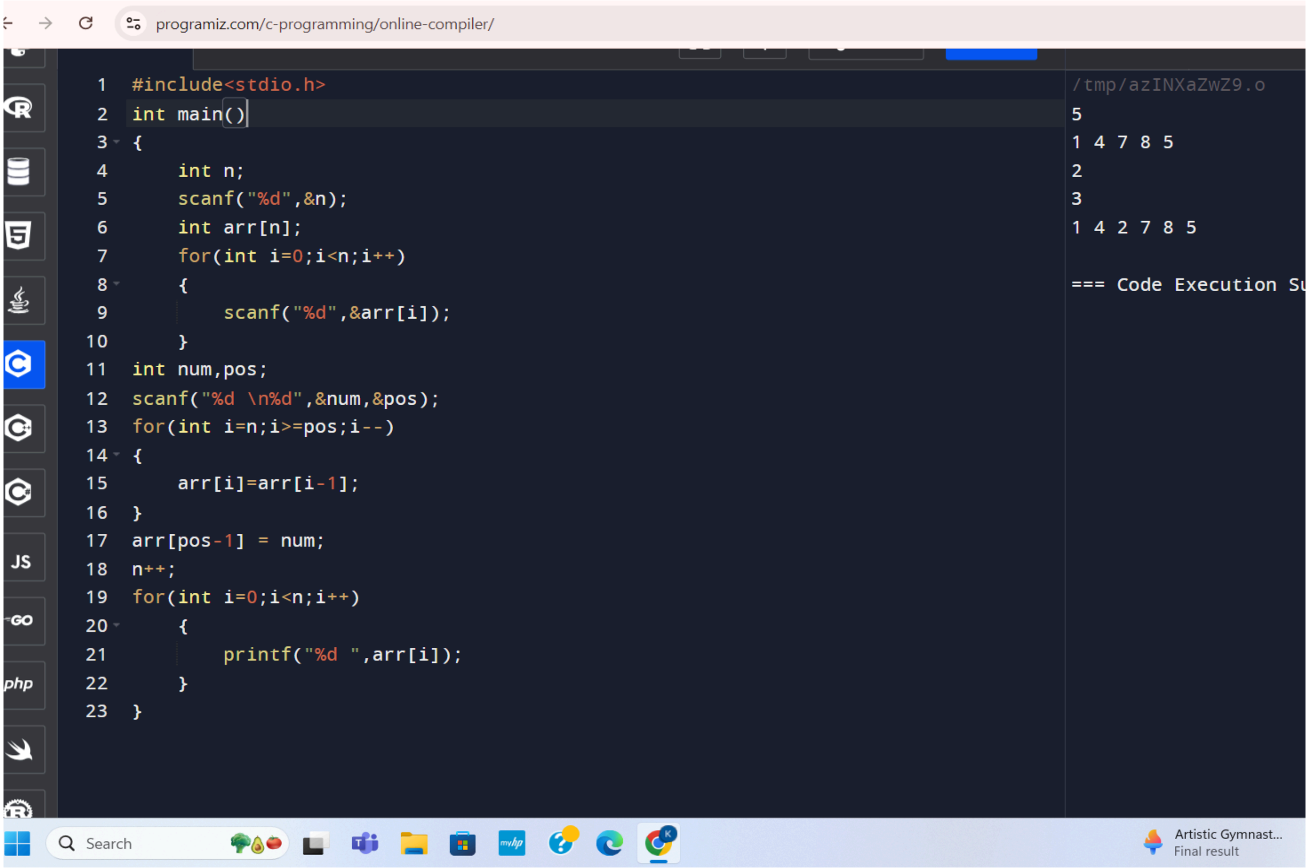
25.CHUNKSIZE



```c
#include<stdio.h>
int main()
{
    int n,chunksize;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    scanf("%d",&chunksize);
    for(int i=0;i<n;i+=chunksize)
    {
        printf("[");
        for(int j=i;j<i+chunksize && j<n;j++)
        {
            printf("%d",arr[j]);
            if(j<i+chunksize-1 && j<n-1)
            {
                printf(",");
            }
        }
        printf("]");
    }
}
```

```
/tmp/tKTi5aWKmL.o
5
10 52 85 63 41
2
[10,52][85,63][41]

=== Code Execution Successful ===
```

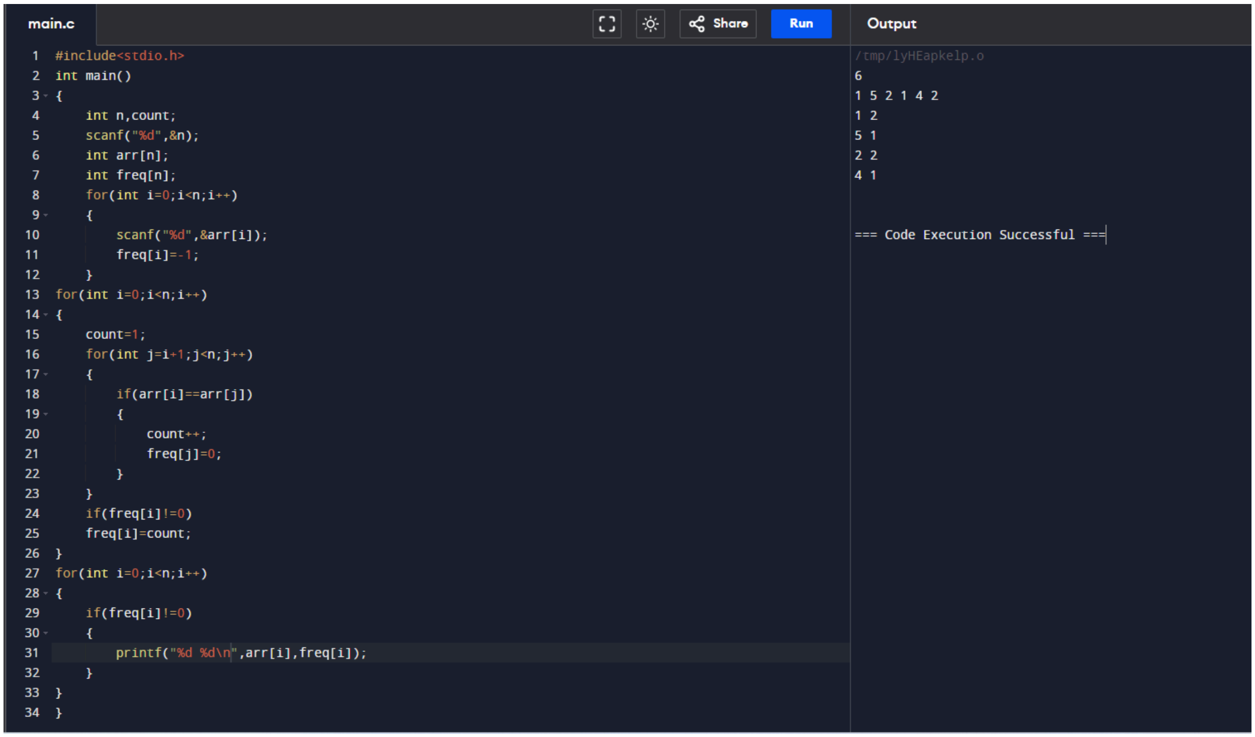
26.INSERTING ELEMENT AT PARTICULAR POSITION



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
int num,pos;
scanf("%d \n%d",&num,&pos);
for(int i=n;i>=pos;i--)
{
    arr[i]=arr[i-1];
}
arr[pos-1] = num;
n++;
for(int i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
}
```

```
/tmp/azINXaZwZ9.o
5
1 4 7 8 5
2
3
1 4 2 7 8 5

=== Code Execution Su
```

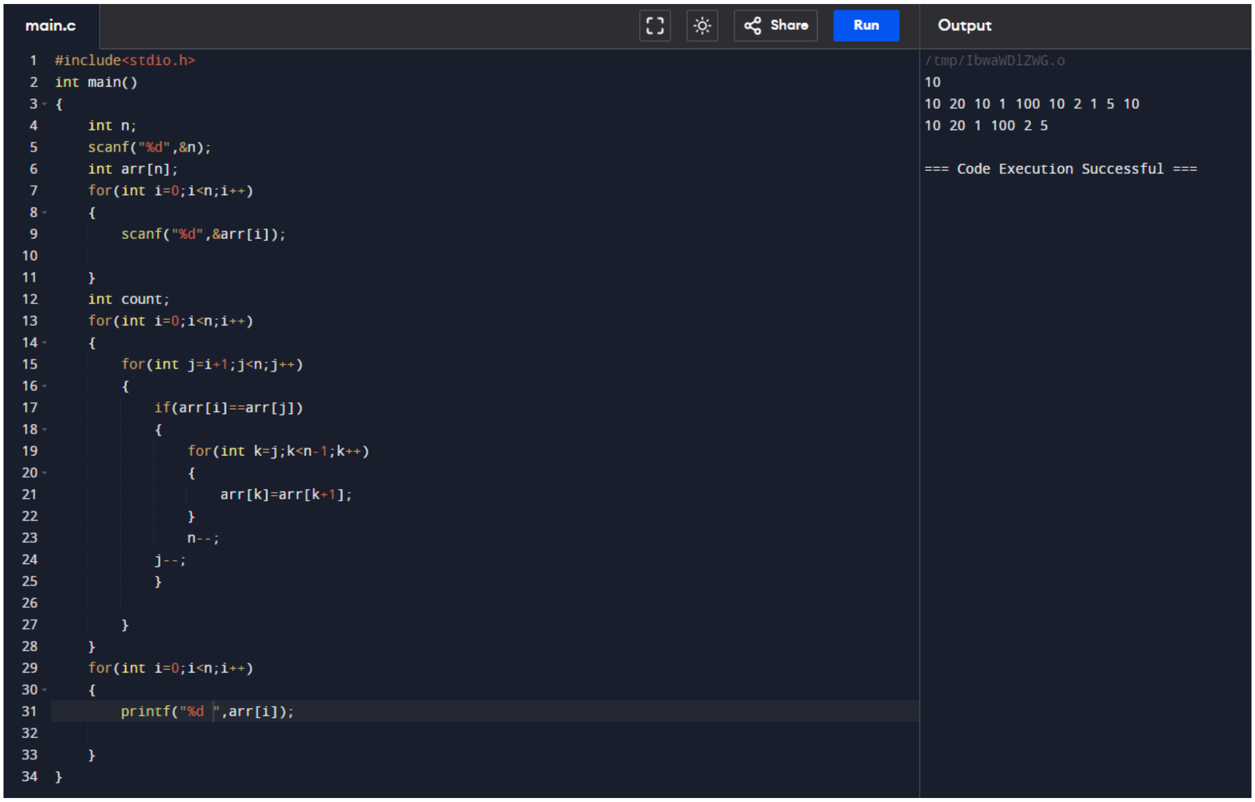
27.FIND FREQUENCY OF ELEMENTS



```c
#include<stdio.h>
int main()
{
    int n,count;
    scanf("%d",&n);
    int arr[n];
    int freq[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
        freq[i]=-1;
    }
for(int i=0;i<n;i++)
{
    count=1;
    for(int j=i+1;j<n;j++)
    {
        if(arr[i]==arr[j])
        {
            count++;
            freq[j]=0;
        }
    }
    if(freq[i]!=0)
    freq[i]=count;
}
for(int i=0;i<n;i++)
{
    if(freq[i]!=0)
    {
        printf("%d %d\n",arr[i],freq[i]);
    }
}
}
```

```
/tmp/lyHEapkelp.o
6
1 5 2 1 4 2
1 2
5 1
2 2
4 1


=== Code Execution Successful ===
```

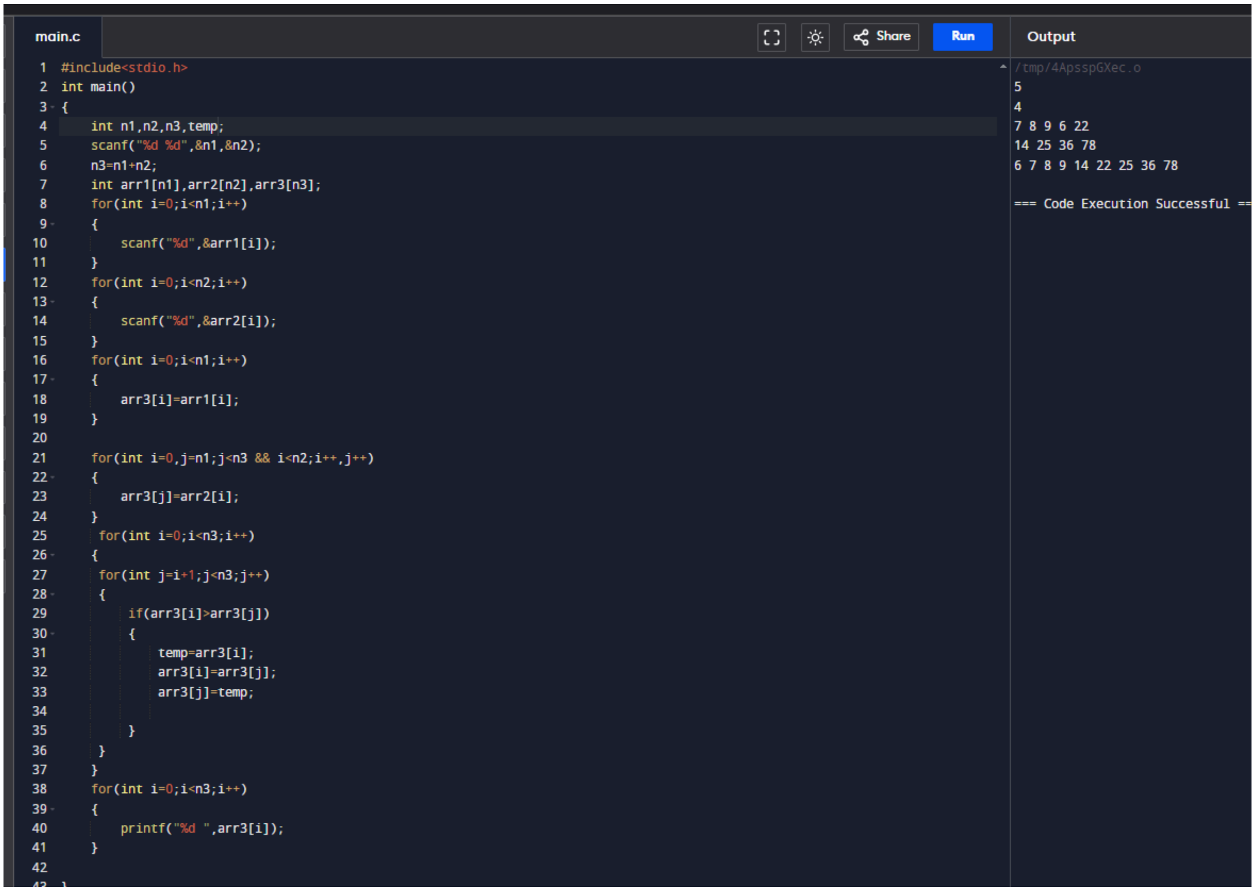
28.DELETING DUPLICATE ELEMENTS(PRINTING 1ST AND DELETING REST ALL)



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);

    }
    int count;
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(arr[i]==arr[j])
            {
                for(int k=j;k<n-1;k++)
                {
                    arr[k]=arr[k+1];
                }
                n--;
            j--;
            }

        }
    }
    for(int i=0;i<n;i++)
    {
        printf("%d ",arr[i]);

    }
}
```

```
/tmp/IbwaWDlZWG.o
10
10 20 10 1 100 10 2 1 5 10
10 20 1 100 2 5

=== Code Execution Successful ===
```

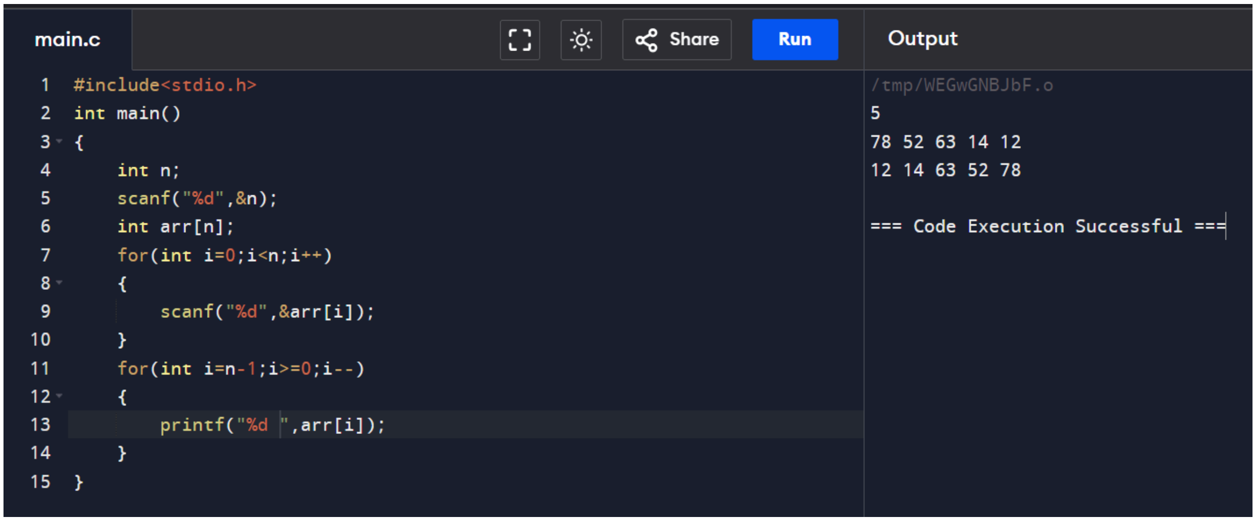
29.MERGE 2 ARRAYS AND ASCENDING ORDER



```c
#include<stdio.h>
int main()
{
    int n1,n2,n3,temp;
    scanf("%d %d",&n1,&n2);
    n3=n1+n2;
    int arr1[n1],arr2[n2],arr3[n3];
    for(int i=0;i<n1;i++)
    {
        scanf("%d",&arr1[i]);
    }
    for(int i=0;i<n2;i++)
    {
        scanf("%d",&arr2[i]);
    }
    for(int i=0;i<n1;i++)
    {
        arr3[i]=arr1[i];
    }

    for(int i=0,j=n1;j<n3 && i<n2;i++,j++)
    {
        arr3[j]=arr2[i];
    }
     for(int i=0;i<n3;i++)
    {
     for(int j=i+1;j<n3;j++)
     {
         if(arr3[i]>arr3[j])
         {
             temp=arr3[i];
             arr3[i]=arr3[j];
             arr3[j]=temp;

         }
     }
    }
    for(int i=0;i<n3;i++)
    {
        printf("%d ",arr3[i]);
    }

}
```

```
/tmp/4ApsspGXec.o
5
4
7 8 9 6 22
14 25 36 78
6 7 8 9 14 22 25 36 78

=== Code Execution Successful ==
```

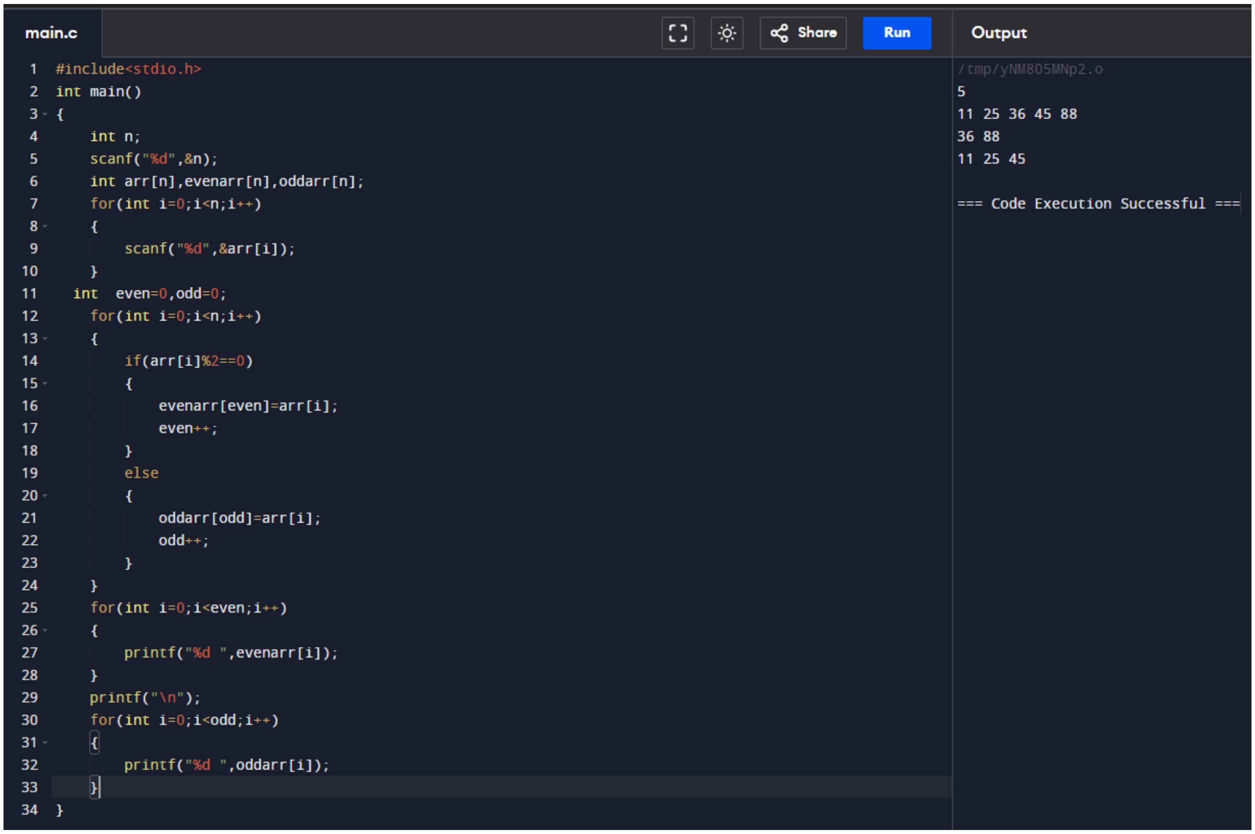
30.REVERSE ARRAY ELEMENTS



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(int i=n-1;i>=0;i--)
    {
        printf("%d ",arr[i]);
    }
}
```

```
/tmp/WEGwGNBJbF.o
5
78 52 63 14 12
12 14 63 52 78

=== Code Execution Successful ===
```

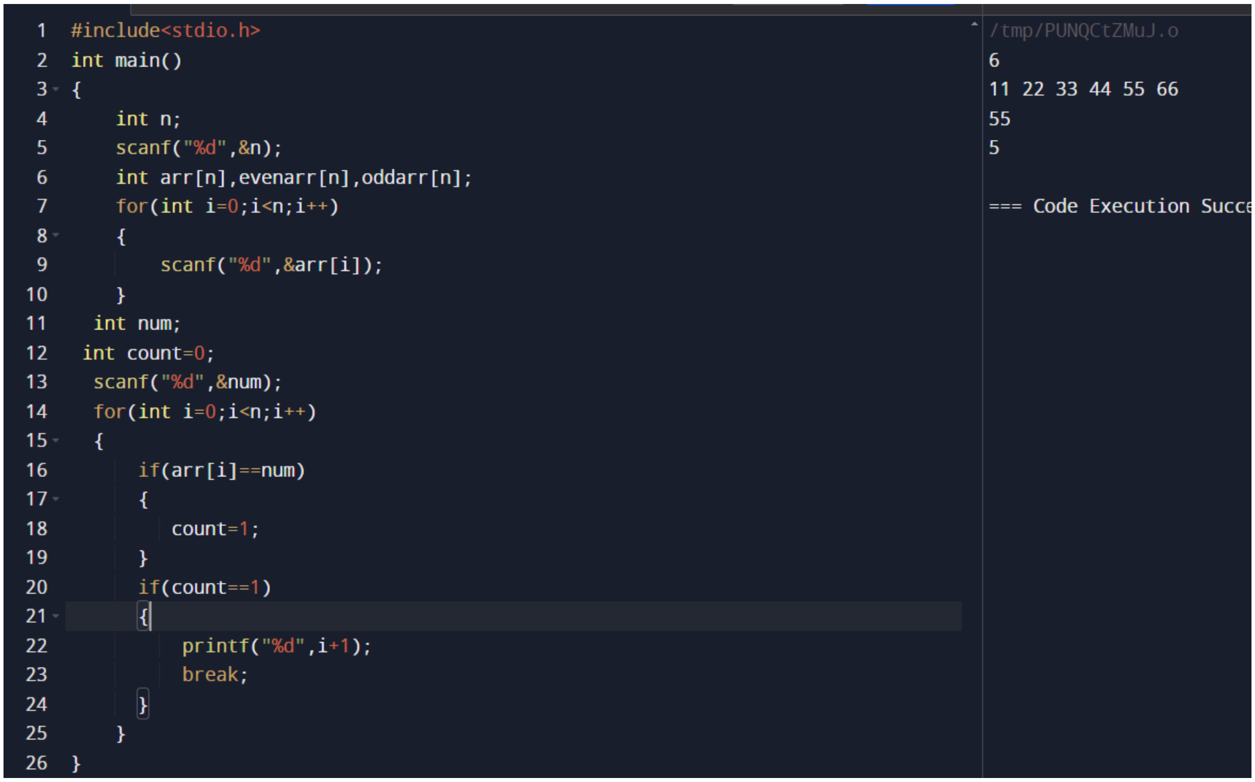
31.PRINTING EVEN AND ODD NUMBERS IN SEPARATE 2 ARRAYS



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n],evenarr[n],oddarr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
  int  even=0,odd=0;
    for(int i=0;i<n;i++)
    {
        if(arr[i]%2==0)
        {
            evenarr[even]=arr[i];
            even++;
        }
        else
        {
            oddarr[odd]=arr[i];
            odd++;
        }
    }
    for(int i=0;i<even;i++)
    {
        printf("%d ",evenarr[i]);
    }
    printf("\n");
    for(int i=0;i<odd;i++)
    {
        printf("%d ",oddarr[i]);
    }
}
```

```
/tmp/yNM805MNp2.o
5
11 25 36 45 88
36 88
11 25 45

=== Code Execution Successful ===
```

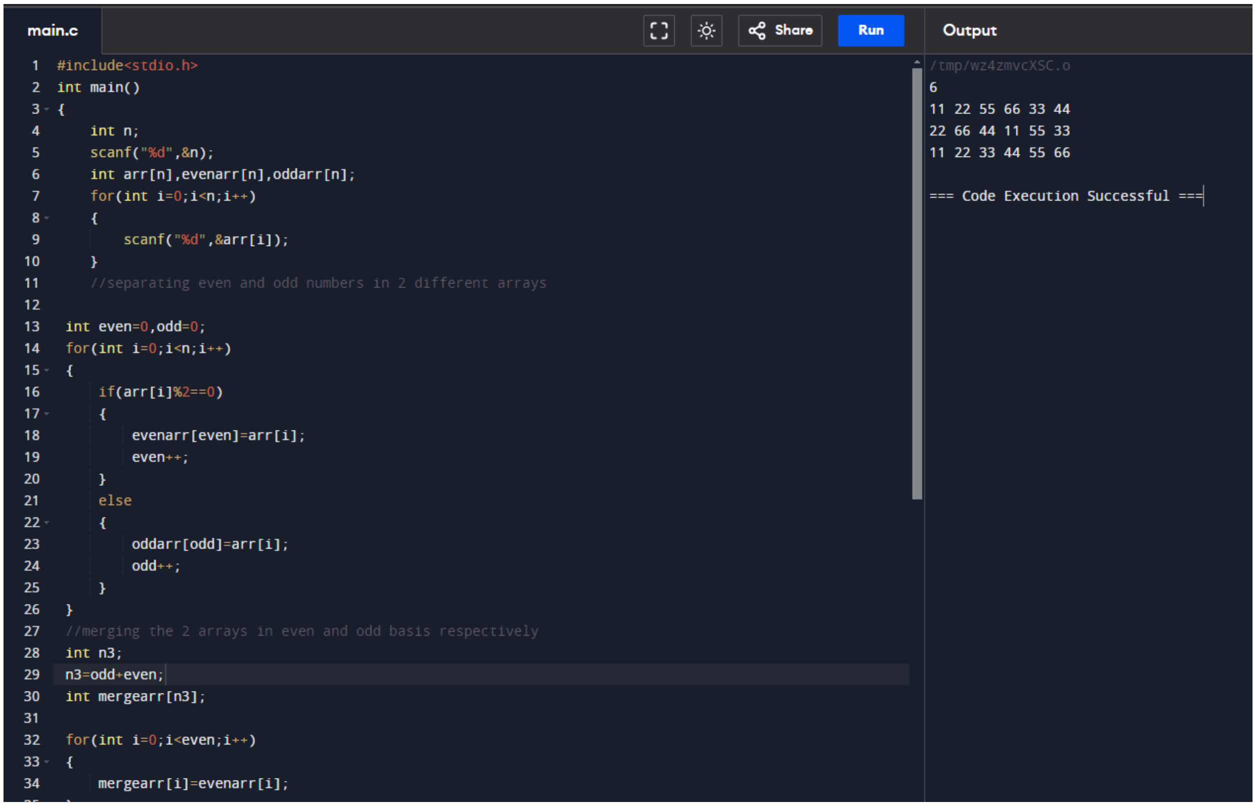
32.SEARCHING AN ELEMENT AND PRINTING ITS POSITION



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n],evenarr[n],oddarr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
  int num;
 int count=0;
  scanf("%d",&num);
  for(int i=0;i<n;i++)
  {
      if(arr[i]==num)
      {
         count=1;
      }
      if(count==1)
      {
          printf("%d",i+1);
          break;
      }
    }
}
```

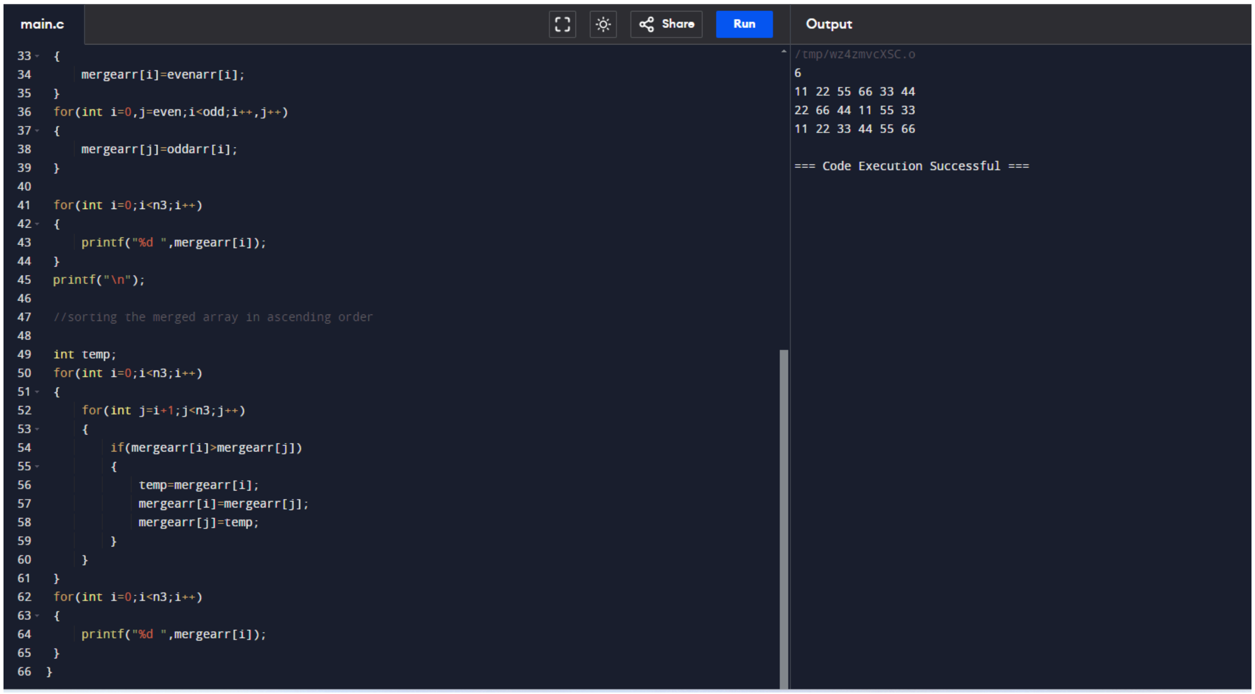
```
/tmp/PUNQCtZMuJ.o
6
11 22 33 44 55 66
55
5

=== Code Execution Succe
```

33.



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n],evenarr[n],oddarr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    //separating even and odd numbers in 2 different arrays

 int even=0,odd=0;
 for(int i=0;i<n;i++)
 {
     if(arr[i]%2==0)
     {
         evenarr[even]=arr[i];
         even++;
     }
     else
     {
         oddarr[odd]=arr[i];
         odd++;
     }
 }
 //merging the 2 arrays in even and odd basis respectively
 int n3;
 n3=odd+even;
 int mergearr[n3];

 for(int i=0;i<even;i++)
 {
     mergearr[i]=evenarr[i];
```

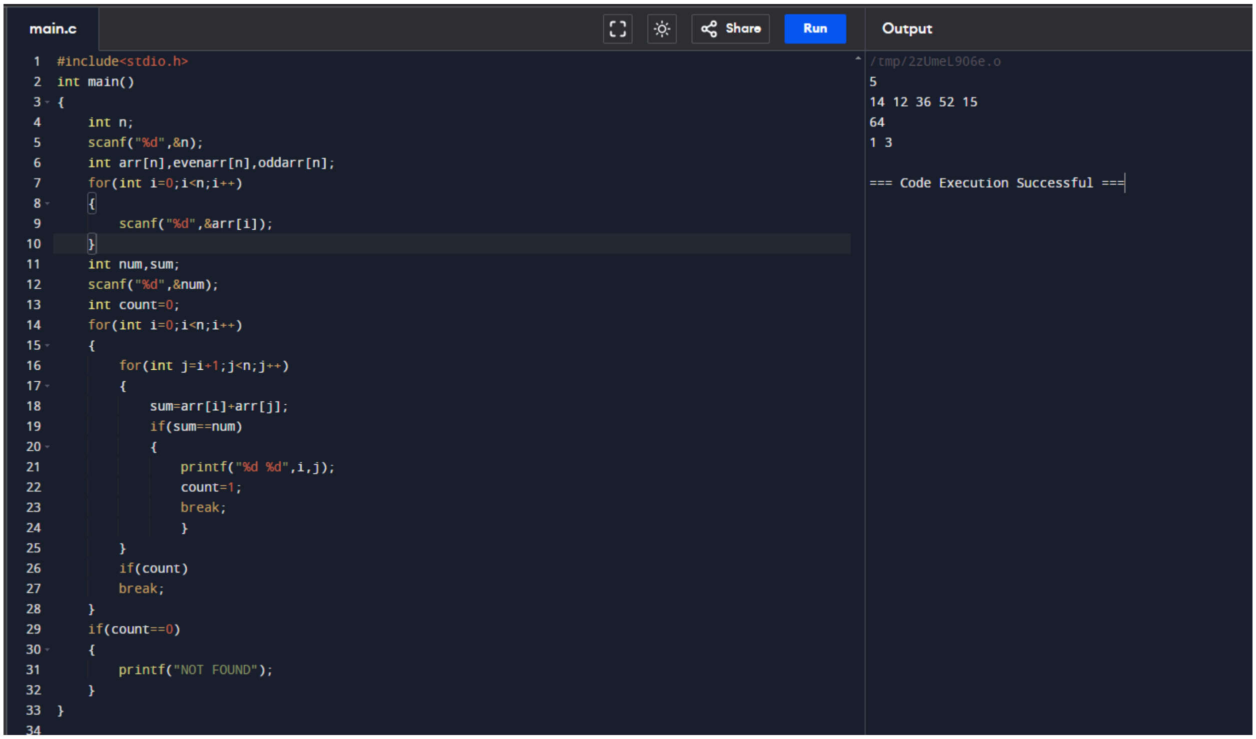
```
/tmp/wz4zmvcXSC.o
6
11 22 55 66 33 44
22 66 44 11 55 33
11 22 33 44 55 66

=== Code Execution Successful ===
```



```c
 {
     mergearr[i]=evenarr[i];
 }
 for(int i=0,j=even;i<odd;i++,j++)
 {
     mergearr[j]=oddarr[i];
 }

 for(int i=0;i<n3;i++)
 {
     printf("%d ",mergearr[i]);
 }
 printf("\n");

 //sorting the merged array in ascending order

 int temp;
 for(int i=0;i<n3;i++)
 {
     for(int j=i+1;j<n3;j++)
     {
         if(mergearr[i]>mergearr[j])
         {
             temp=mergearr[i];
             mergearr[i]=mergearr[j];
             mergearr[j]=temp;
         }
     }
 }
 for(int i=0;i<n3;i++)
 {
     printf("%d ",mergearr[i]);
 }
}
```

```
/tmp/wz4zmvcXSC.o
6
11 22 55 66 33 44
22 66 44 11 55 33
11 22 33 44 55 66

=== Code Execution Successful ===
```

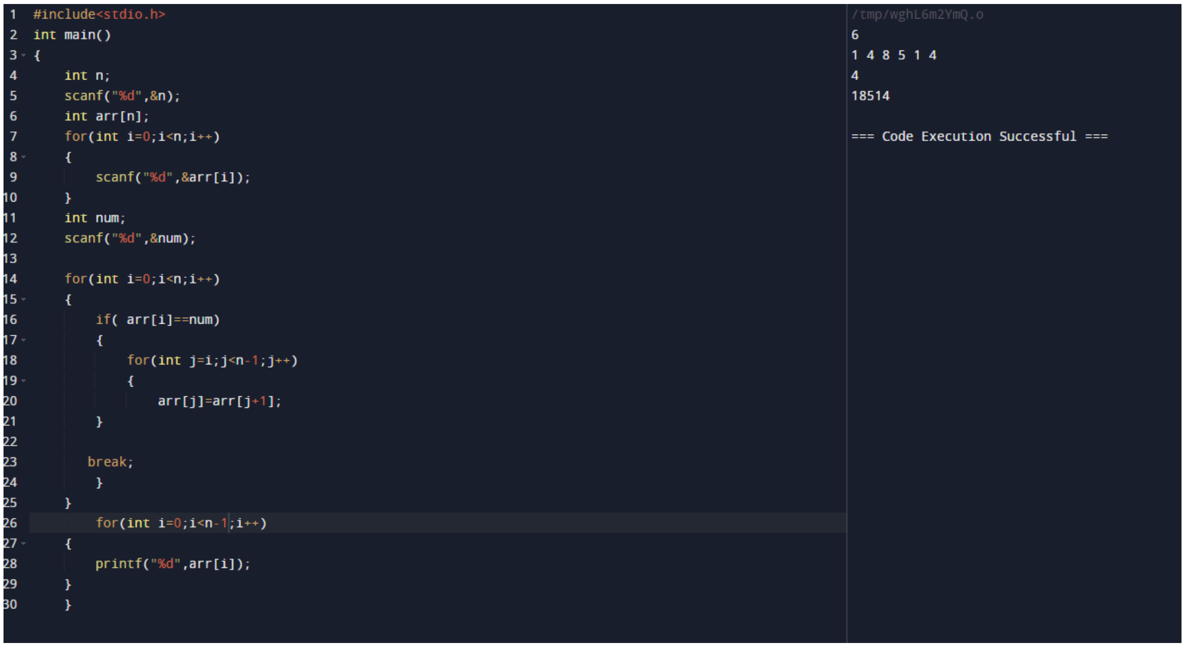
34.Write a program in C to find a pair with a given sum in the array.



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n],evenarr[n],oddarr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int num,sum;
    scanf("%d",&num);
    int count=0;
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            sum=arr[i]+arr[j];
            if(sum==num)
            {
                printf("%d %d",i,j);
                count=1;
                break;
                }
        }
        if(count)
        break;
    }
    if(count==0)
    {
        printf("NOT FOUND");
    }
}
```

```
/tmp/2zUmeL906e.o
5
14 12 36 52 15
64
1 3

=== Code Execution Successful ===
```

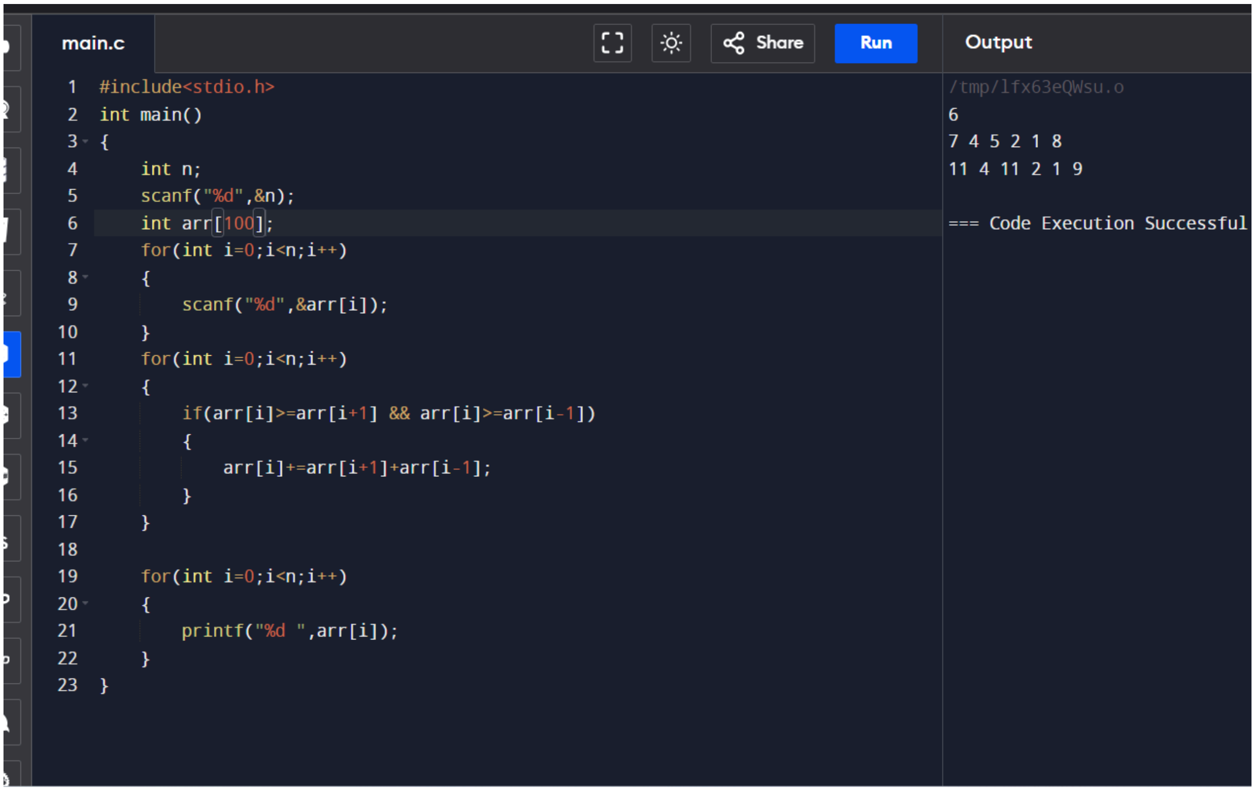
35.removing 1st occurrence of a number



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int num;
    scanf("%d",&num);

    for(int i=0;i<n;i++)
    {
        if( arr[i]==num)
        {
            for(int j=i;j<n-1;j++)
            {
                arr[j]=arr[j+1];
        }

       break;
        }
    }
        for(int i=0;i<n-1;i++)
    {
        printf("%d",arr[i]);
    }
    }
```

```
/tmp/wghL6m2YmQ.o
6
1 4 8 5 1 4
4
18514

=== Code Execution Successful ===
```

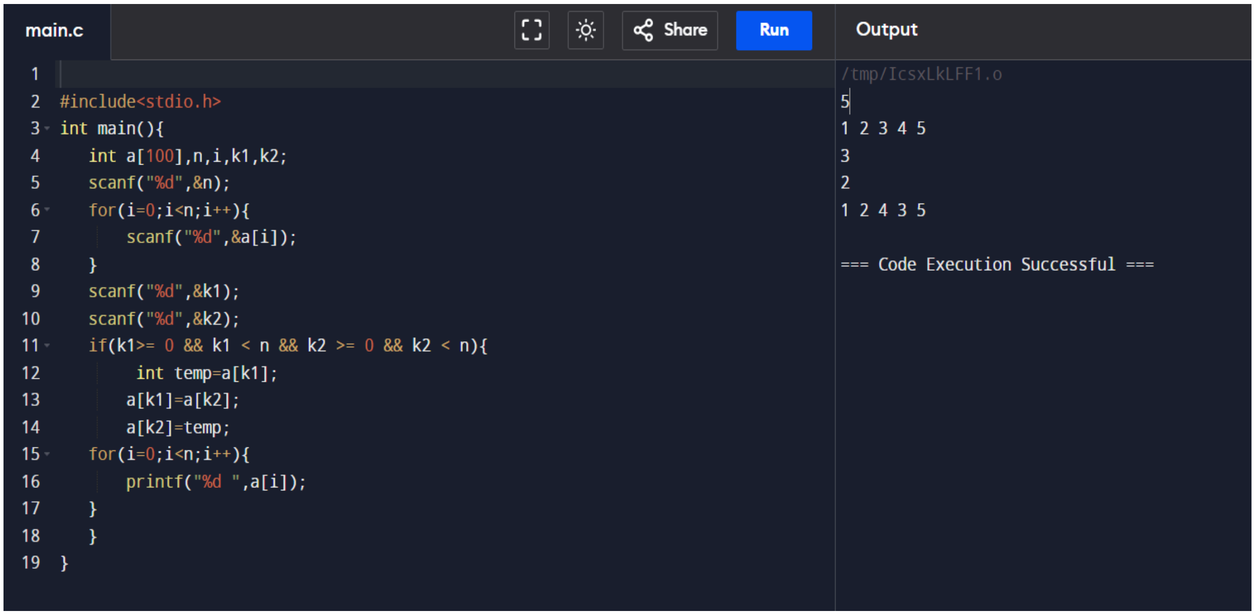
36.SUM OF NEIGHBOR PEAK ELEMENTS



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[100];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(int i=0;i<n;i++)
    {
        if(arr[i]>=arr[i+1] && arr[i]>=arr[i-1])
        {
            arr[i]+=arr[i+1]+arr[i-1];
        }
    }

    for(int i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
}
```

```
/tmp/1fx63eQWsu.o
6
7 4 5 2 1 8
11 4 11 2 1 9

=== Code Execution Successful
```

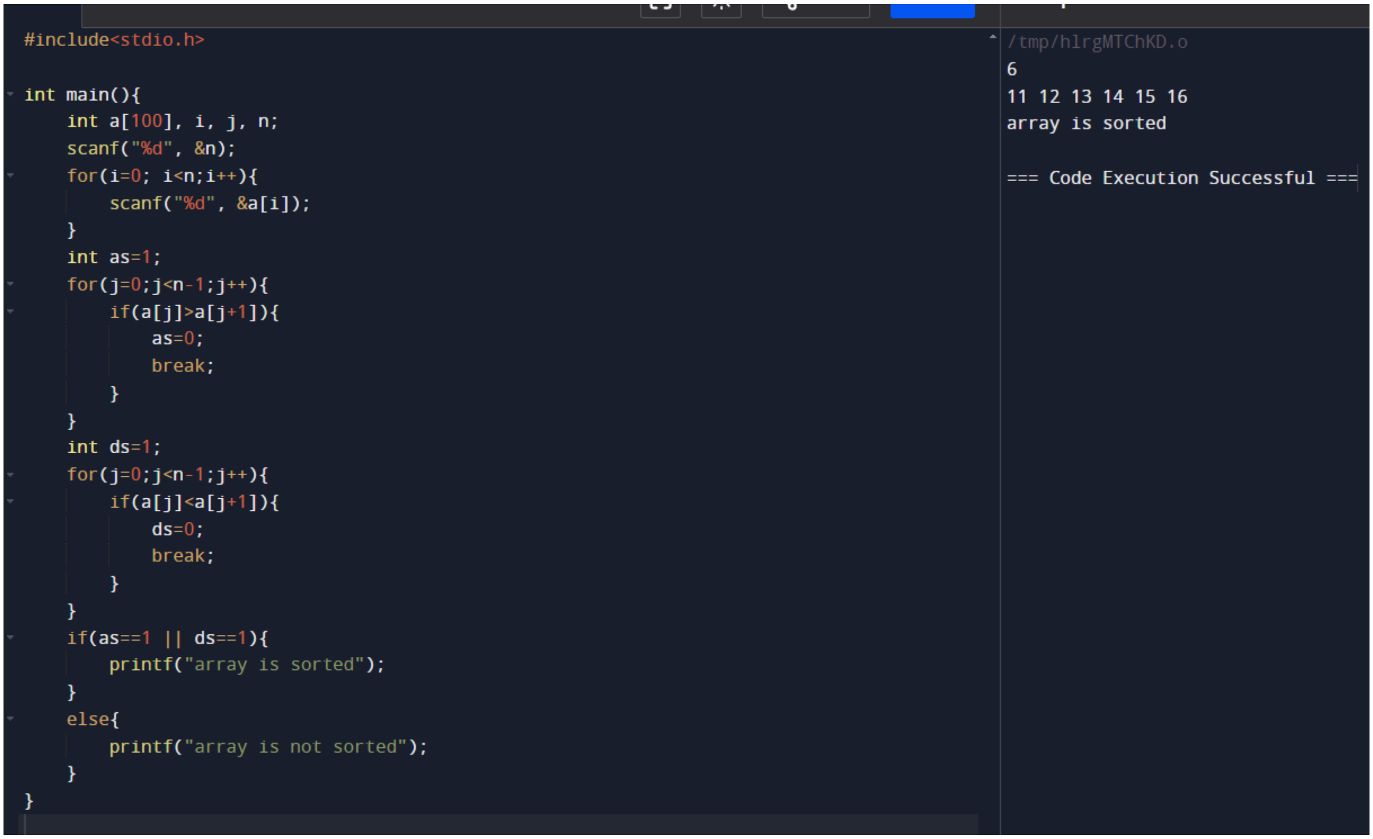
37. Swapping TWO elements:



```c

#include<stdio.h>
int main(){
    int a[100],n,i,k1,k2;
    scanf("%d",&n);
    for(i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    scanf("%d",&k1);
    scanf("%d",&k2);
    if(k1>= 0 && k1 < n && k2 >= 0 && k2 < n){
         int temp=a[k1];
        a[k1]=a[k2];
        a[k2]=temp;
    for(i=0;i<n;i++){
        printf("%d ",a[i]);
    }
    }
}
```

```
/tmp/IcsxLkLFF1.o
5
1 2 3 4 5
3
2
1 2 4 3 5

=== Code Execution Successful ===
```

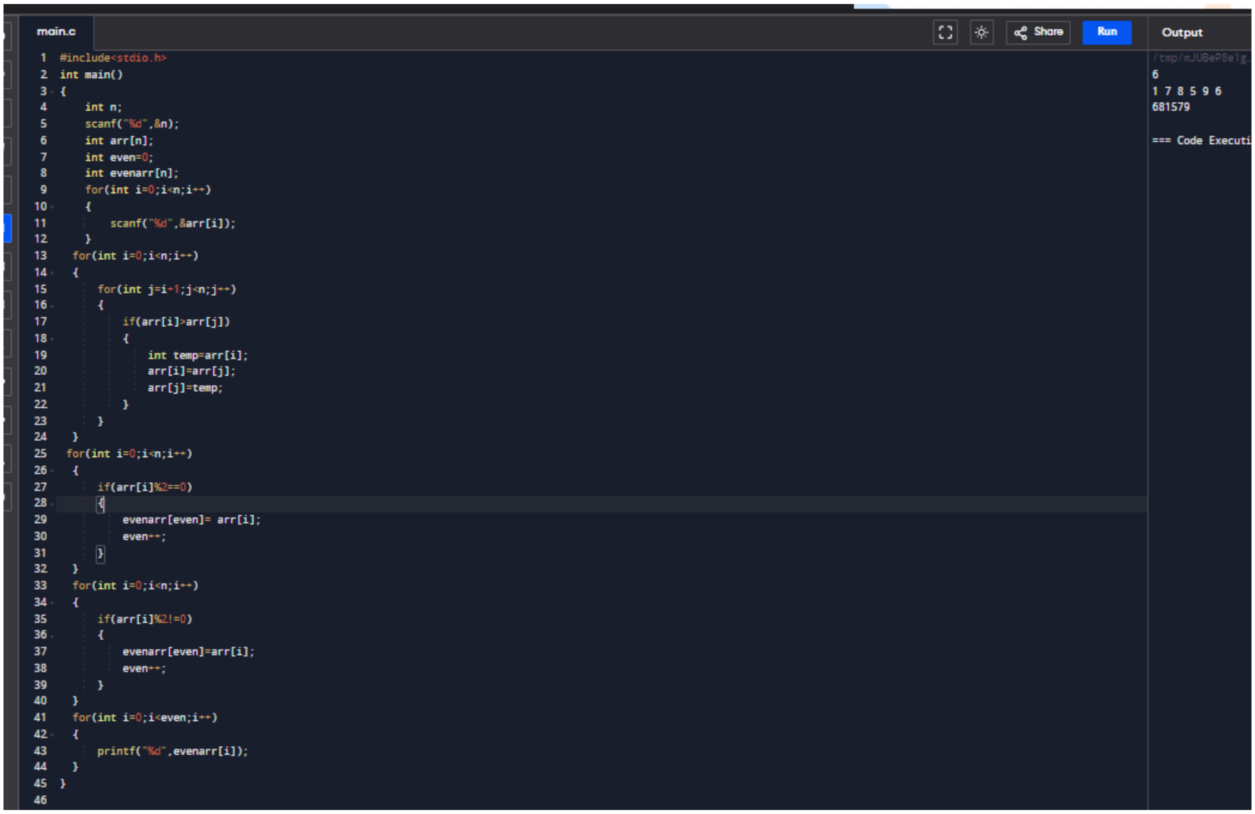
38.Sorted or NOT Sorted:



```c
#include<stdio.h>

int main(){
    int a[100], i, j, n;
    scanf("%d", &n);
    for(i=0; i<n;i++){
        scanf("%d", &a[i]);
    }
    int as=1;
    for(j=0;j<n-1;j++){
        if(a[j]>a[j+1]){
            as=0;
            break;
        }
    }
    int ds=1;
    for(j=0;j<n-1;j++){
        if(a[j]<a[j+1]){
            ds=0;
            break;
        }
    }
    if(as==1 || ds==1){
        printf("array is sorted");
    }
    else{
        printf("array is not sorted");
    }
}
```

```
/tmp/hlrgMTChKD.o
6
11 12 13 14 15 16
array is sorted

=== Code Execution Successful ===
```

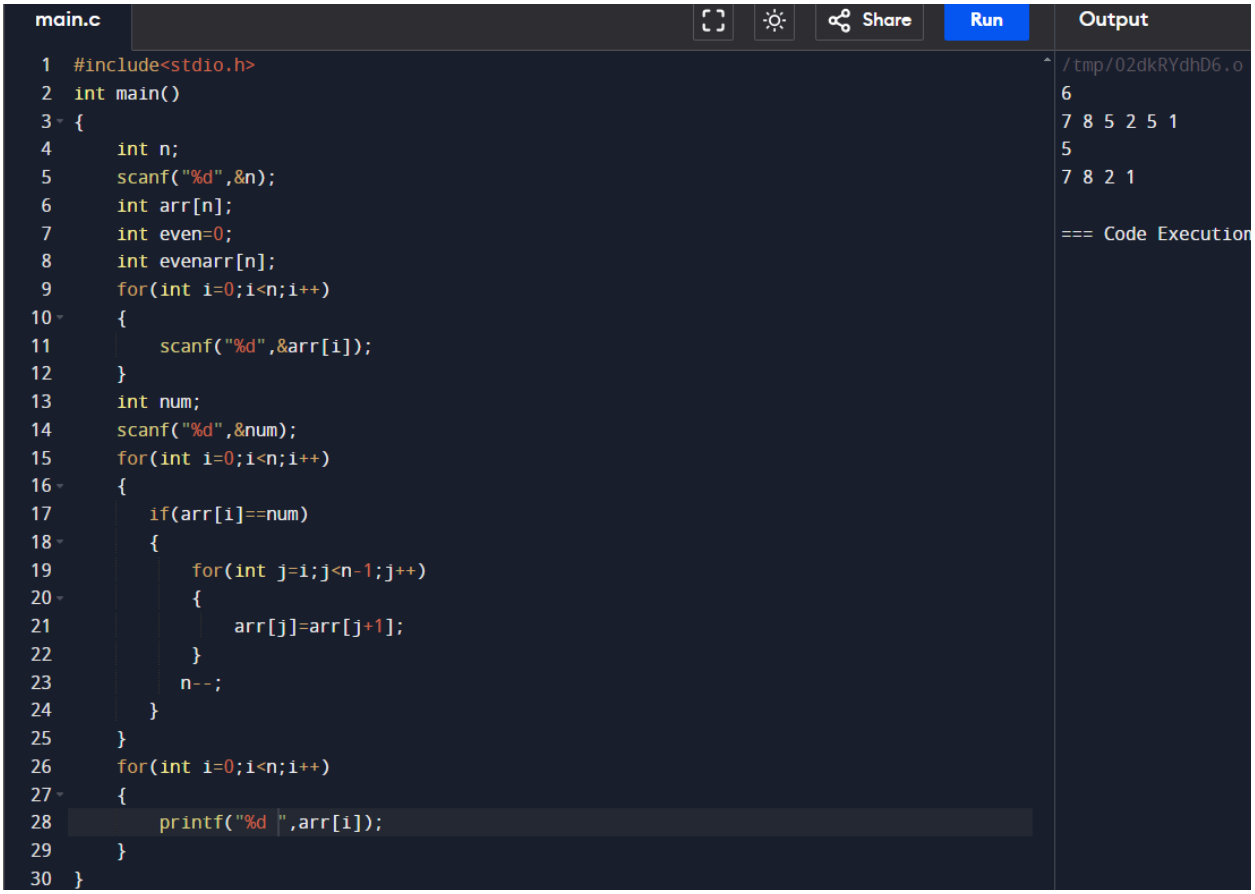
39. **Sort the array in ascending order and print even numbers first and odd numbers next**



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    int even=0;
    int evenarr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
  for(int i=0;i<n;i++)
  {
      for(int j=i+1;j<n;j++)
      {
          if(arr[i]>arr[j])
          {
              int temp=arr[i];
              arr[i]=arr[j];
              arr[j]=temp;
          }
      }
  }
 for(int i=0;i<n;i++)
  {
      if(arr[i]%2==0)
      {
          evenarr[even]= arr[i];
          even++;
      }
  }
  for(int i=0;i<n;i++)
  {
      if(arr[i]%2!=0)
      {
          evenarr[even]=arr[i];
          even++;
      }
  }
  for(int i=0;i<even;i++)
  {
      printf("%d",evenarr[i]);
  }
}
```

```
/tmp/mJUBeP8e1g.
6
1 7 8 5 9 6
681579

=== Code Executi
```

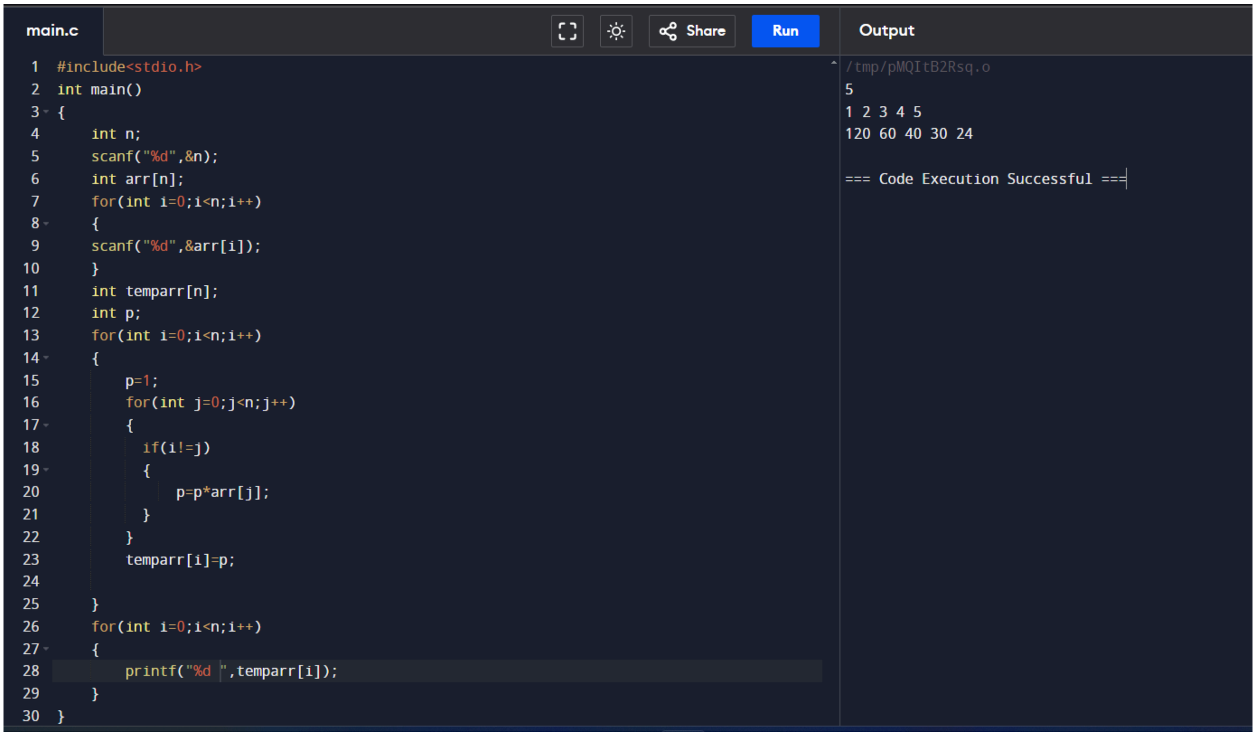
40.Removing the occurrence of the digit is :



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    int even=0;
    int evenarr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int num;
    scanf("%d",&num);
    for(int i=0;i<n;i++)
    {
       if(arr[i]==num)
       {
           for(int j=i;j<n-1;j++)
           {
               arr[j]=arr[j+1];
           }
          n--;
       }
    }
    for(int i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
}
```

```
/tmp/02dkRYdhD6.o
6
7 8 5 2 5 1
5
7 8 2 1

=== Code Execution
```

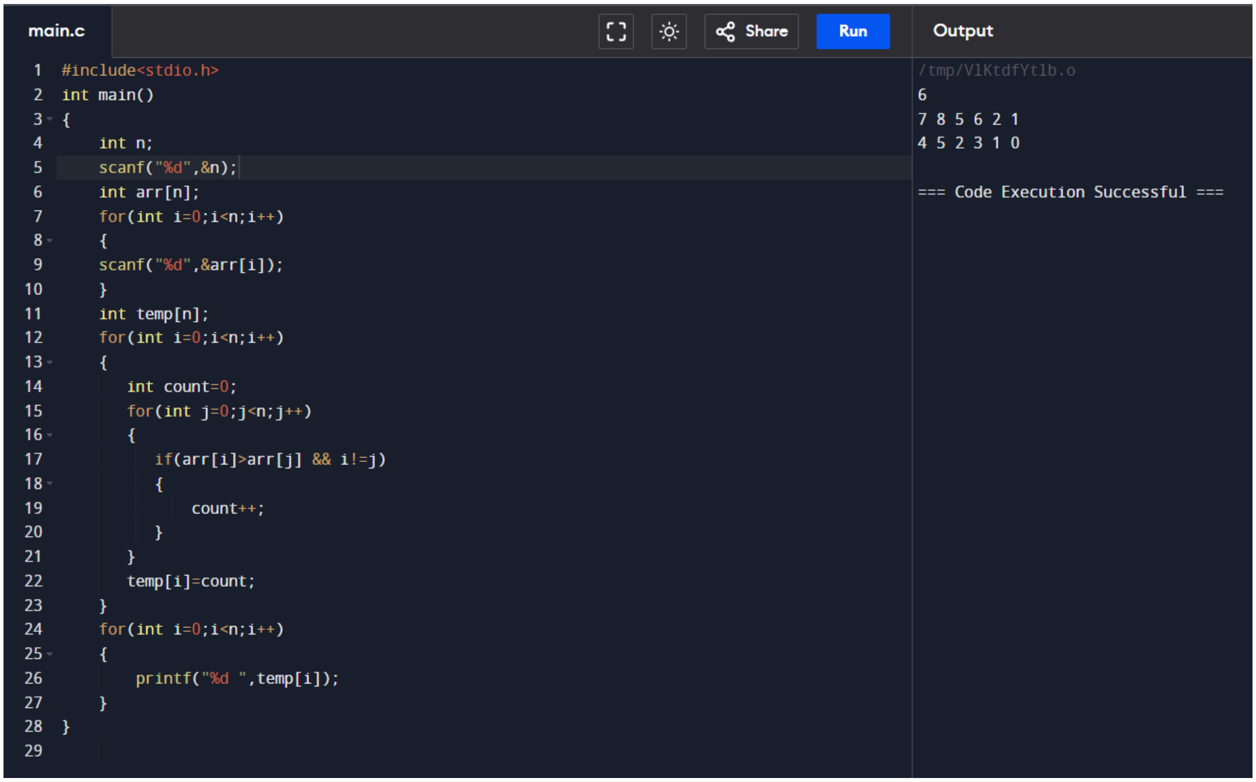
41.Product of an array without multiplying the present index :



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
    scanf("%d",&arr[i]);
    }
    int temparr[n];
    int p;
    for(int i=0;i<n;i++)
    {
        p=1;
        for(int j=0;j<n;j++)
        {
          if(i!=j)
          {
              p=p*arr[j];
          }
        }
        temparr[i]=p;

    }
    for(int i=0;i<n;i++)
    {
        printf("%d ",temparr[i]);
    }
}
```

```
/tmp/pMQItB2Rsq.o
5
1 2 3 4 5
120 60 40 30 24

=== Code Execution Successful ===
```

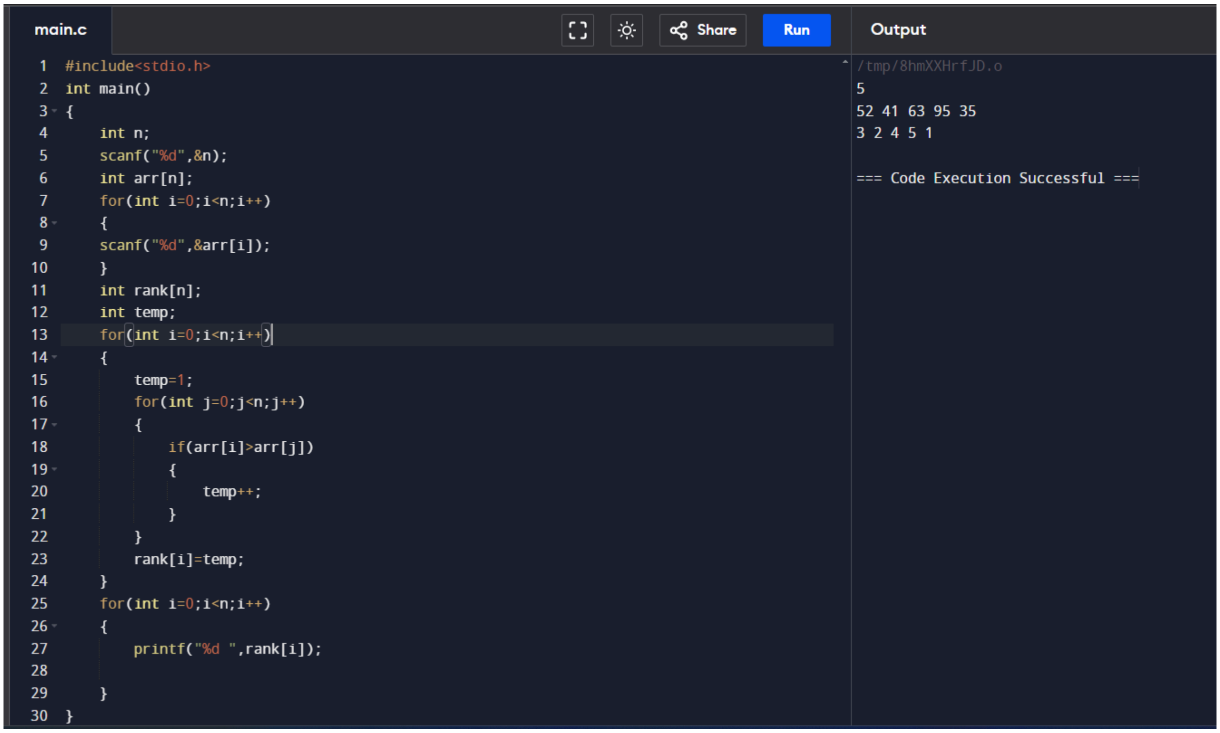
42.**How Many Numbers Are Smaller Than the CurrentNumber Input:nums=[8,1,2,2,3] Output:[4,0,1,1,3]**



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
    scanf("%d",&arr[i]);
    }
    int temp[n];
    for(int i=0;i<n;i++)
    {
       int count=0;
       for(int j=0;j<n;j++)
       {
          if(arr[i]>arr[j] && i!=j)
          {
              count++;
          }
       }
       temp[i]=count;
    }
    for(int i=0;i<n;i++)
    {
        printf("%d ",temp[i]);
    }
}
```

```
/tmp/V1KtdfYtlb.o
6
7 8 5 6 2 1
4 5 2 3 1 0

=== Code Execution Successful ===
```

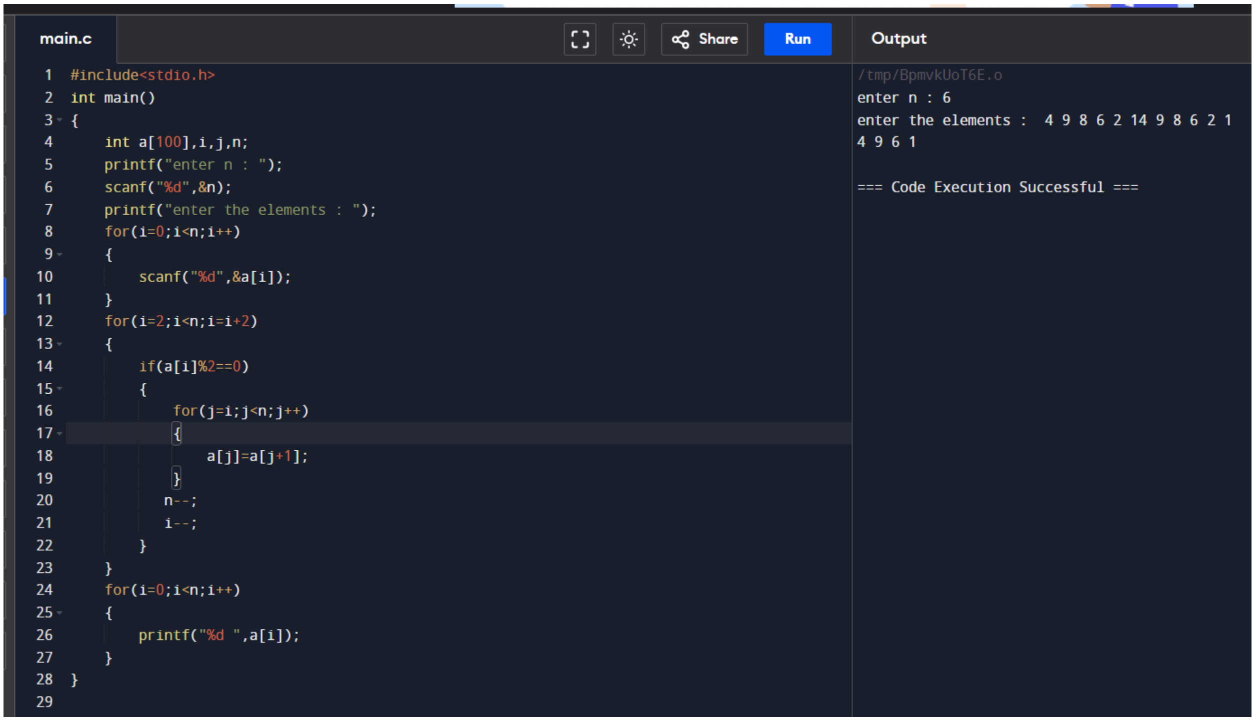
43.RANKING



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
    scanf("%d",&arr[i]);
    }
    int rank[n];
    int temp;
    for(int i=0;i<n;i++)
    {
        temp=1;
        for(int j=0;j<n;j++)
        {
            if(arr[i]>arr[j])
            {
                temp++;
            }
        }
        rank[i]=temp;
    }
    for(int i=0;i<n;i++)
    {
        printf("%d ",rank[i]);

    }
}
```

```
/tmp/8hmXXHrfJD.o
5
52 41 63 95 35
3 2 4 5 1

=== Code Execution Successful ===
```

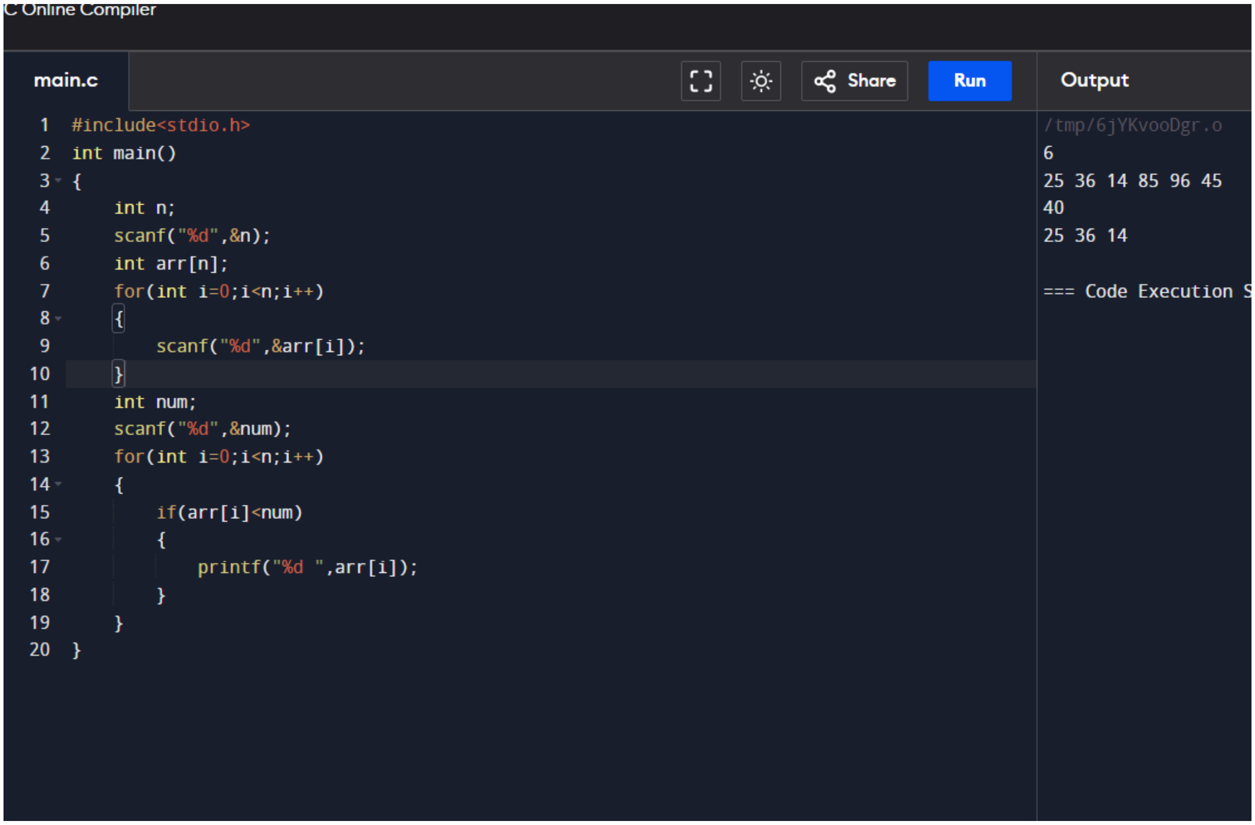
44.REMOVING EVEN NUMBER IN EVEN INDEX



```c
#include<stdio.h>
int main()
{
    int a[100],i,j,n;
    printf("enter n : ");
    scanf("%d",&n);
    printf("enter the elements : ");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=2;i<n;i=i+2)
    {
        if(a[i]%2==0)
        {
            for(j=i;j<n;j++)
            {
                a[j]=a[j+1];
            }
           n--;
           i--;
        }
    }
    for(i=0;i<n;i++)
    {
        printf("%d ",a[i]);
    }
}
```

```
/tmp/BpmvkUoT6E.o
enter n : 6
enter the elements :  4 9 8 6 2 14 9 8 6 2 1
4 9 6 1

=== Code Execution Successful ===
```

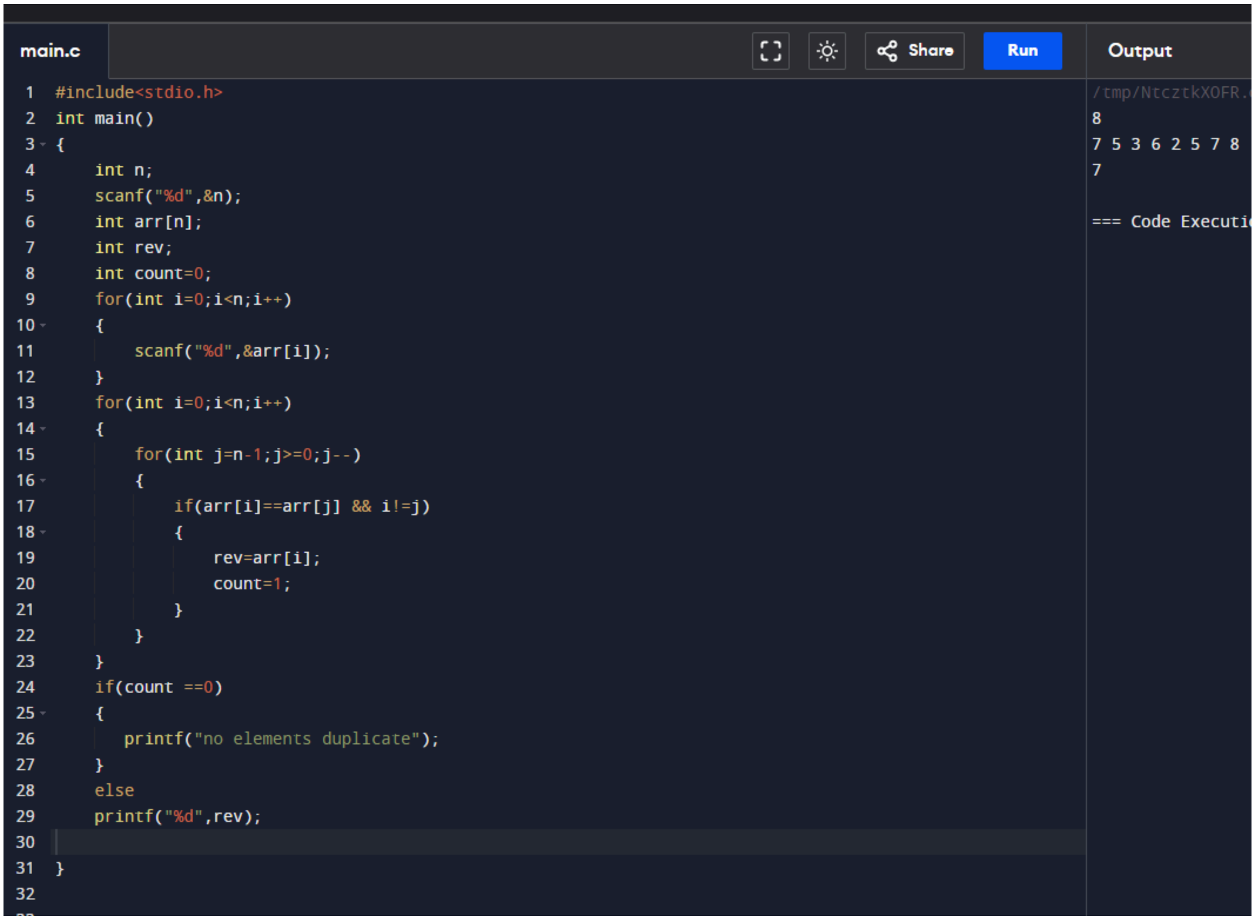
45.REMOVING THE ELEMENTS GREATER THAN THE NUMBER



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int num;
    scanf("%d",&num);
    for(int i=0;i<n;i++)
    {
        if(arr[i]<num)
        {
            printf("%d ",arr[i]);
        }
    }
}
```

```
/tmp/6jYKvooDgr.o
6
25 36 14 85 96 45
40
25 36 14

=== Code Execution S
```

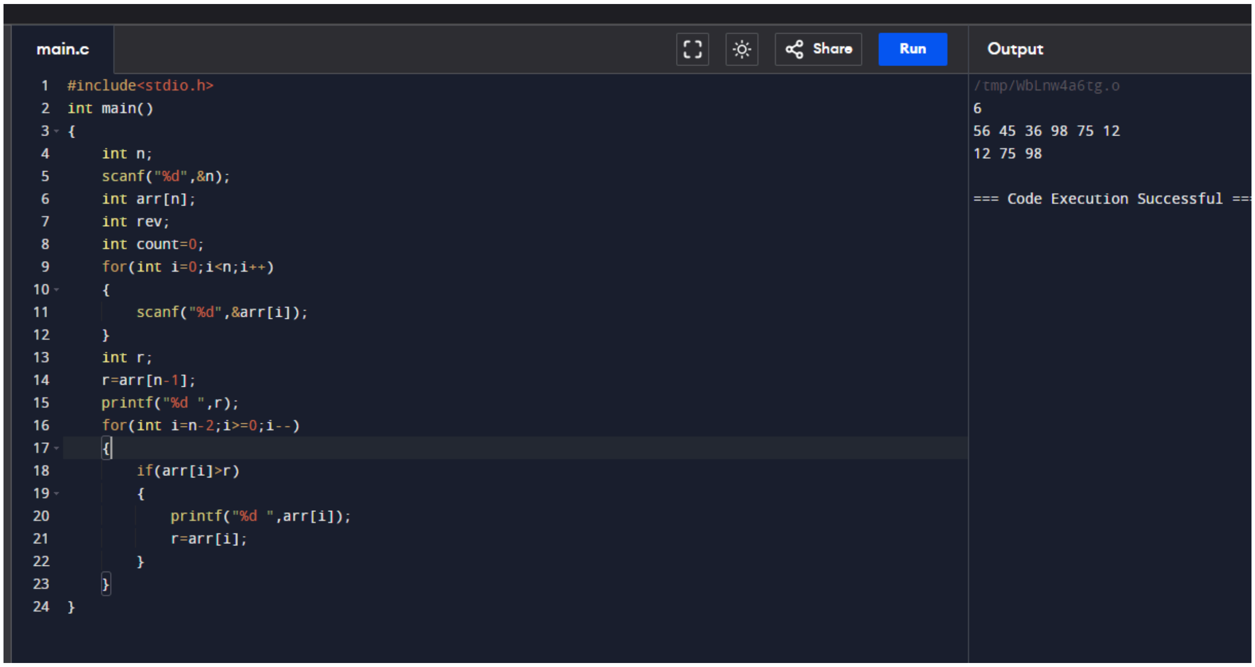
46. **Last repeated element :**



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    int rev;
    int count=0;
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(int i=0;i<n;i++)
    {
        for(int j=n-1;j>=0;j--)
        {
            if(arr[i]==arr[j] && i!=j)
            {
                rev=arr[i];
                count=1;
            }
        }
    }
    if(count ==0)
    {
       printf("no elements duplicate");
    }
    else
    printf("%d",rev);

}

```

```
/tmp/NtcztkXOFR.
8
7 5 3 6 2 5 7 8
7

=== Code Executi
```

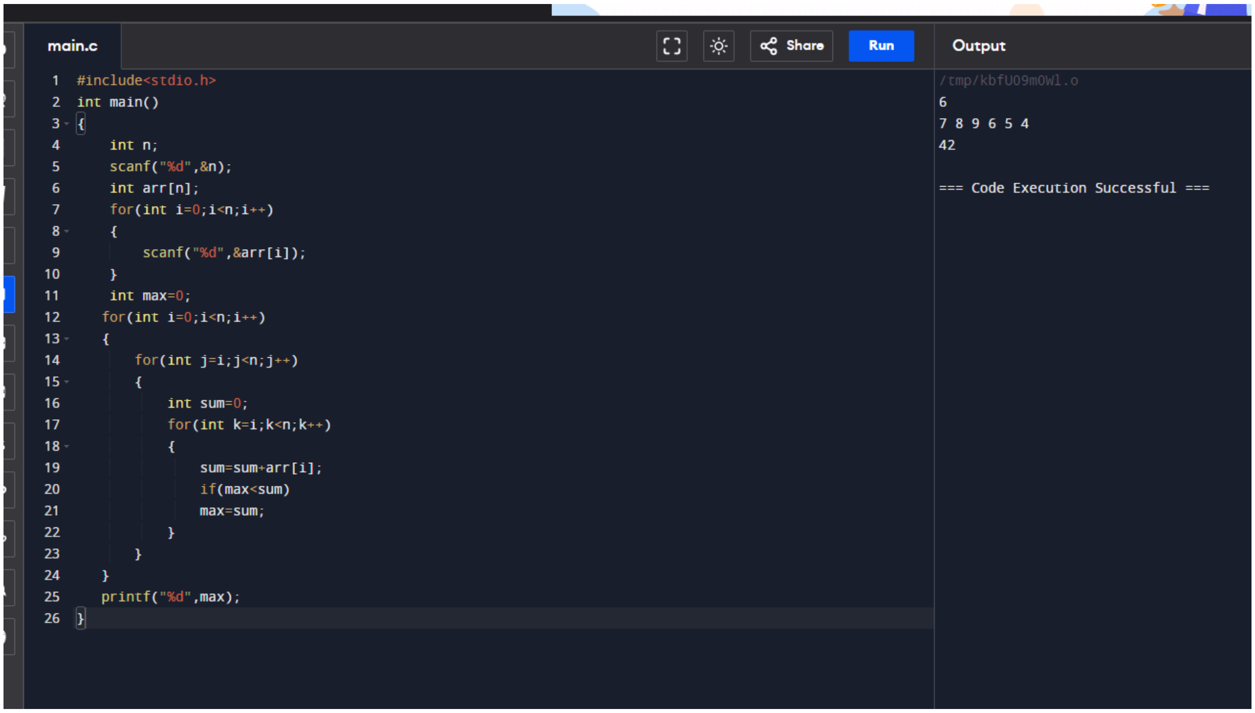
47.Find the element in an array that are greater than all elements to their right and print them in ascending order sample input : 5 arr: 5 3 20 15 8 ouput: 8 15 20



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    int rev;
    int count=0;
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int r;
    r=arr[n-1];
    printf("%d ",r);
    for(int i=n-2;i>=0;i--)
    {
        if(arr[i]>r)
        {
            printf("%d ",arr[i]);
            r=arr[i];
        }
    }
}
```

```
/tmp/WbLnw4a6tg.o
6
56 45 36 98 75 12
12 75 98

=== Code Execution Successful ==
```

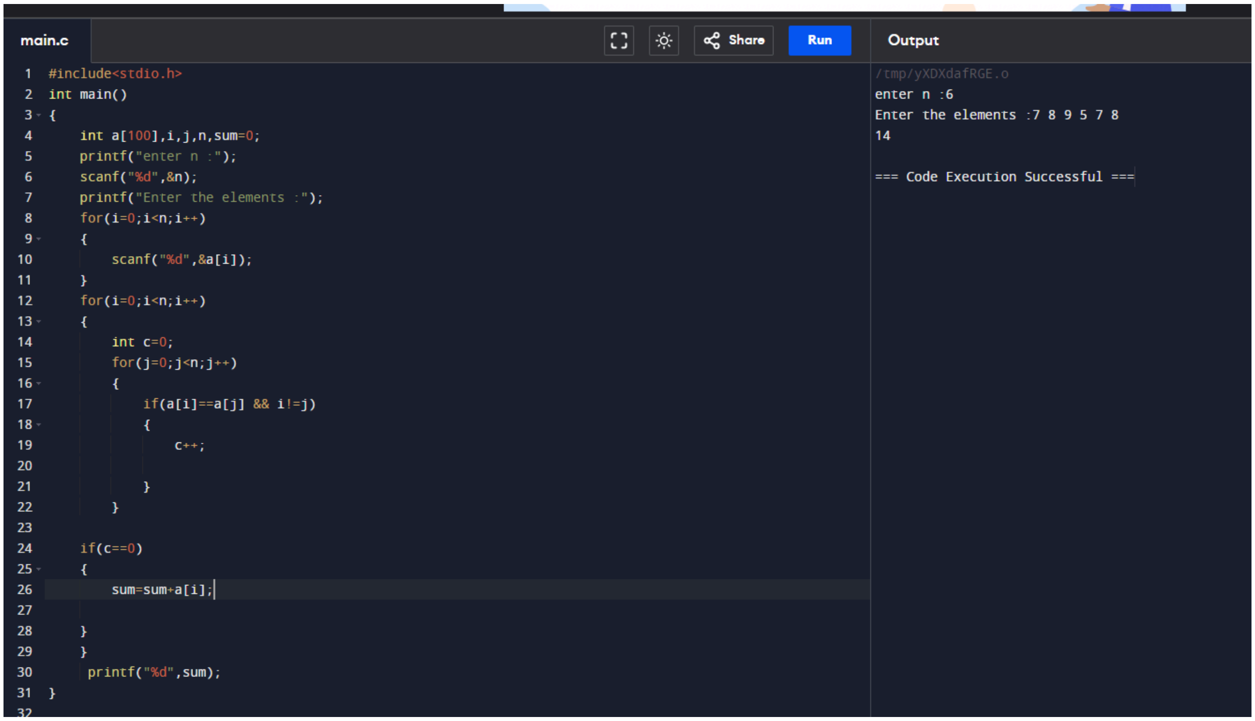
48.MAX sum of array elements



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int max=0;
   for(int i=0;i<n;i++)
   {
       for(int j=i;j<n;j++)
       {
           int sum=0;
           for(int k=i;k<n;k++)
           {
               sum=sum+arr[i];
               if(max<sum)
               max=sum;
           }
       }
   }
   printf("%d",max);
}
```

```
/tmp/kbfU09mOWl.o
6
7 8 9 6 5 4
42

=== Code Execution Successful ===
```

49. **Sum of non repeated numbers : (unique element logic)**

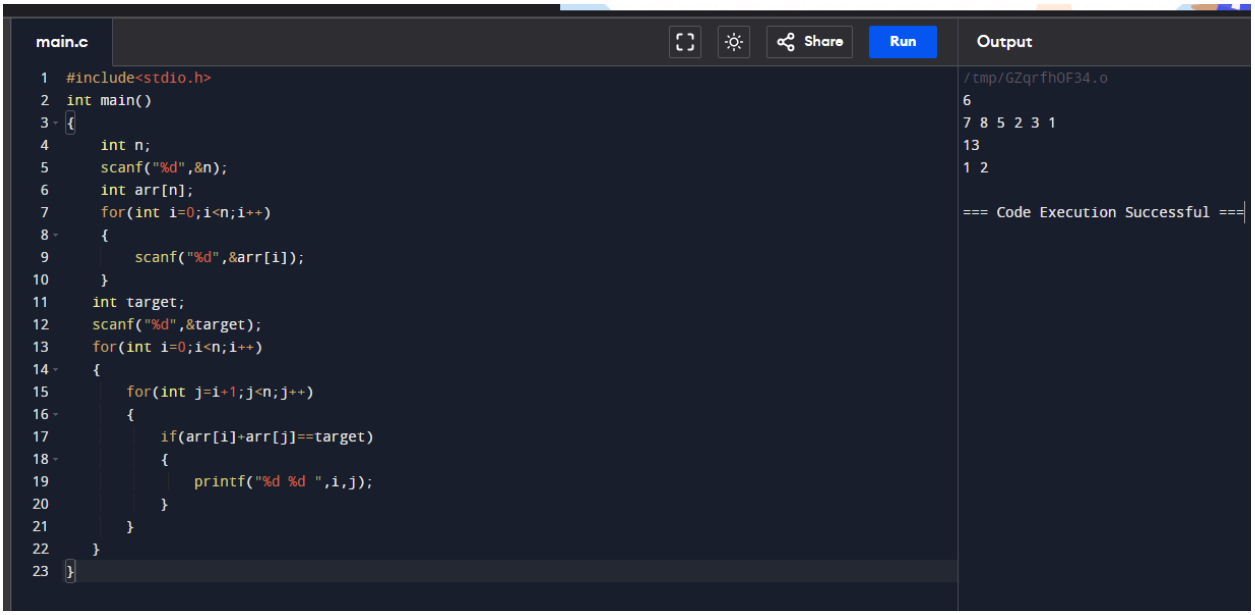


```c
#include<stdio.h>
int main()
{
    int a[100],i,j,n,sum=0;
    printf("enter n :");
    scanf("%d",&n);
    printf("Enter the elements :");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<n;i++)
    {
        int c=0;
        for(j=0;j<n;j++)
        {
            if(a[i]==a[j] && i!=j)
            {
                c++;

            }
        }

    if(c==0)
    {
        sum=sum+a[i];

    }
    }
     printf("%d",sum);
}
```

```
/tmp/yXDXdafRGE.o
enter n :6
Enter the elements :7 8 9 5 7 8
14

=== Code Execution Successful ===
```

50.LEFT CHAR ROTATION

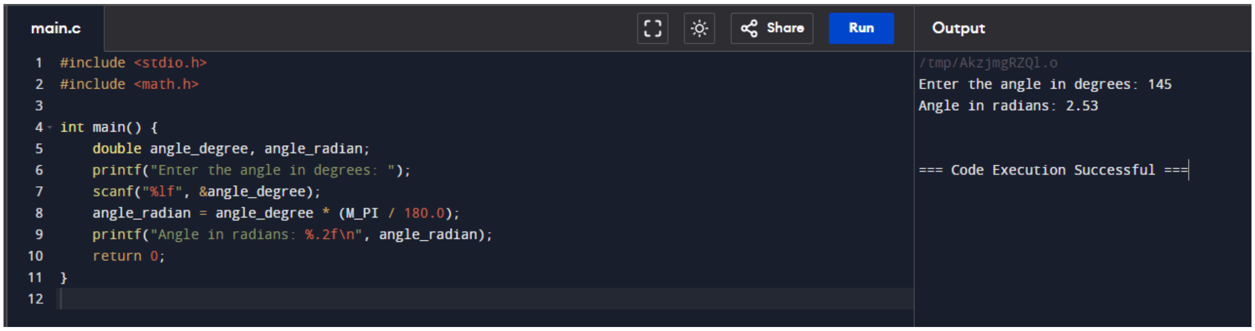
51.**Printing the index of the pairs that gives the target element :**



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    int target;
    scanf("%d",&target);
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(arr[i]+arr[j]==target)
            {
                printf("%d %d ",i,j);
            }
        }
    }
}
```

```
/tmp/GZqrfhOF34.o
6
7 8 5 2 3 1
13
1 2

=== Code Execution Successful ===
```

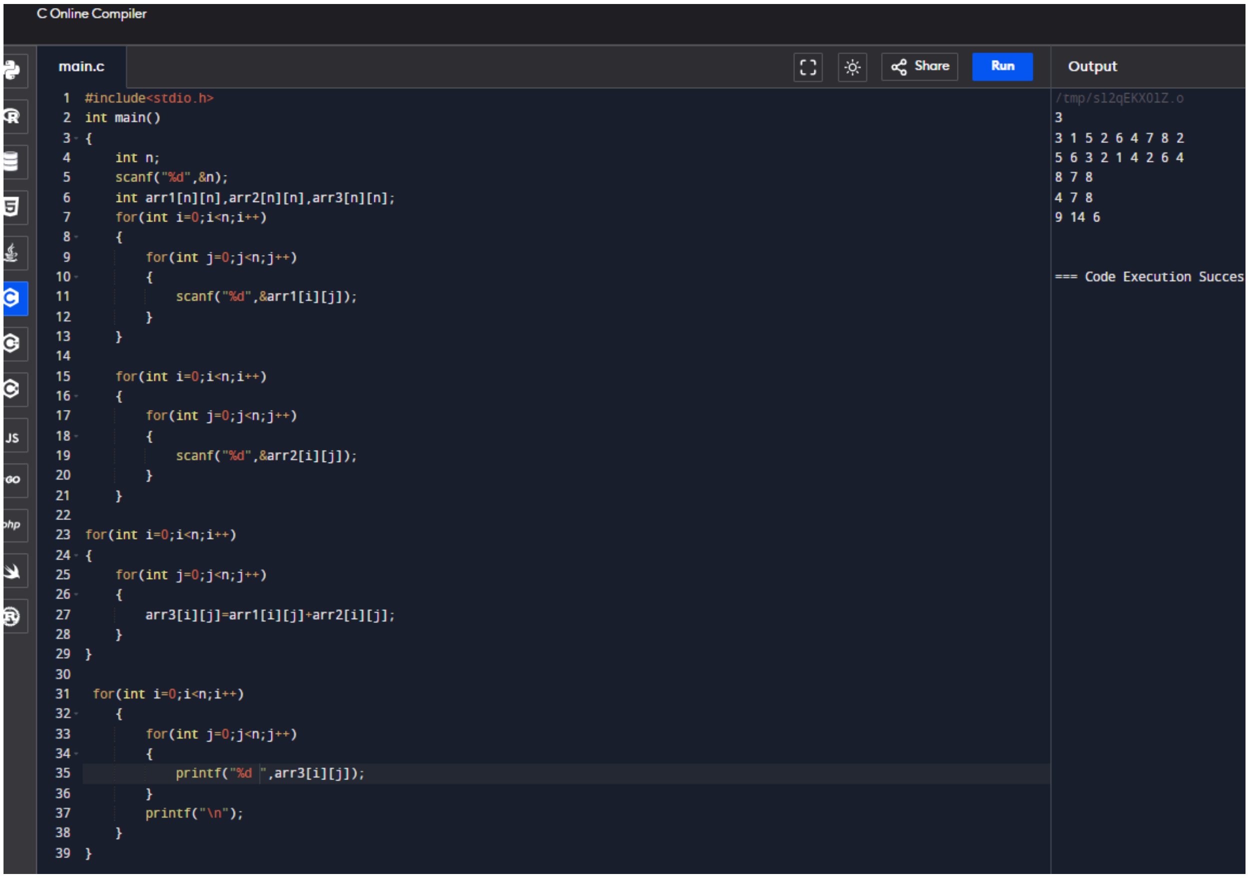
52. **Converting angle to radian :**



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

int main() {
    double angle_degree, angle_radian;
    printf("Enter the angle in degrees: ");
    scanf("%lf", &angle_degree);
    angle_radian = angle_degree * (M_PI / 180.0);
    printf("Angle in radians: %.2f\n", angle_radian);
    return 0;
}
```

```
/tmp/AkzjmgRZQl.o
Enter the angle in degrees: 145
Angle in radians: 2.53


=== Code Execution Successful ===
```

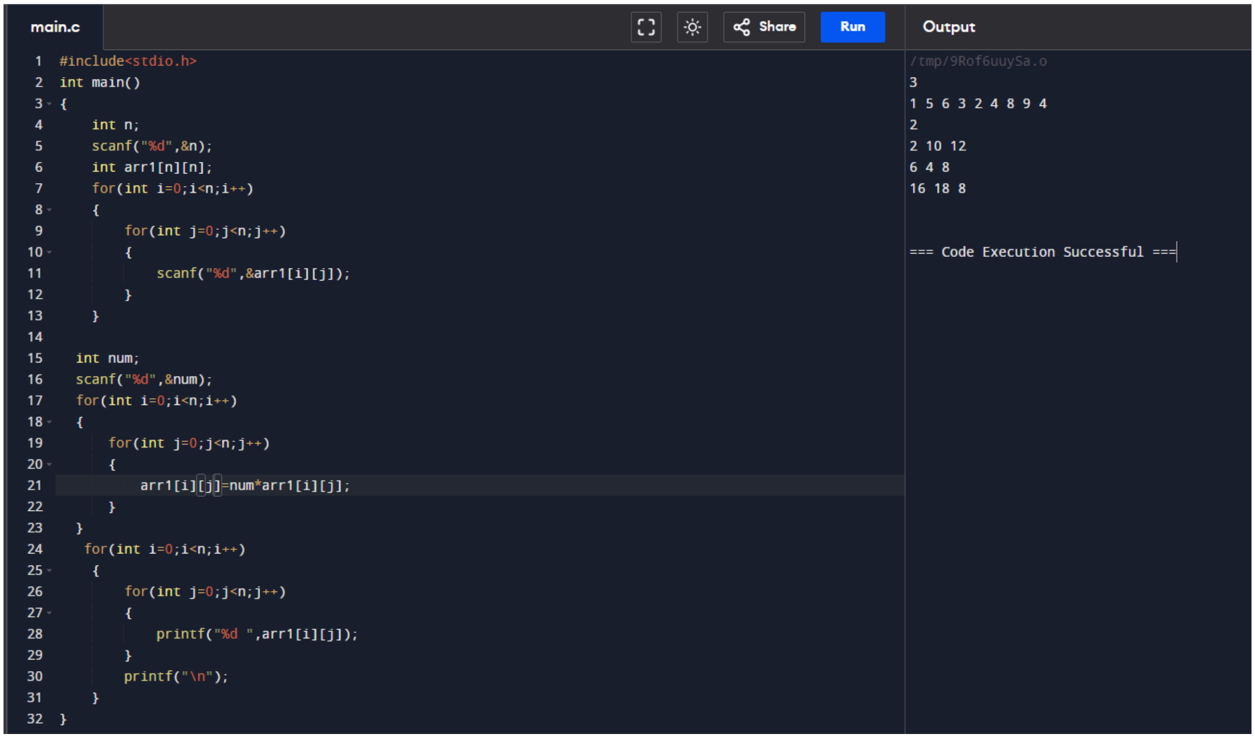
53.SUM OF TWO MATRICES



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr1[n][n],arr2[n][n],arr3[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr1[i][j]);
        }
    }

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr2[i][j]);
        }
    }

for(int i=0;i<n;i++)
{
    for(int j=0;j<n;j++)
    {
        arr3[i][j]=arr1[i][j]+arr2[i][j];
    }
}

 for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("%d ",arr3[i][j]);
        }
        printf("\n");
    }
}
```

```
/tmp/sl2qEKX01Z.o
3
3 1 5 2 6 4 7 8 2
5 6 3 2 1 4 2 6 4
8 7 8
4 7 8
9 14 6


=== Code Execution Succes
```

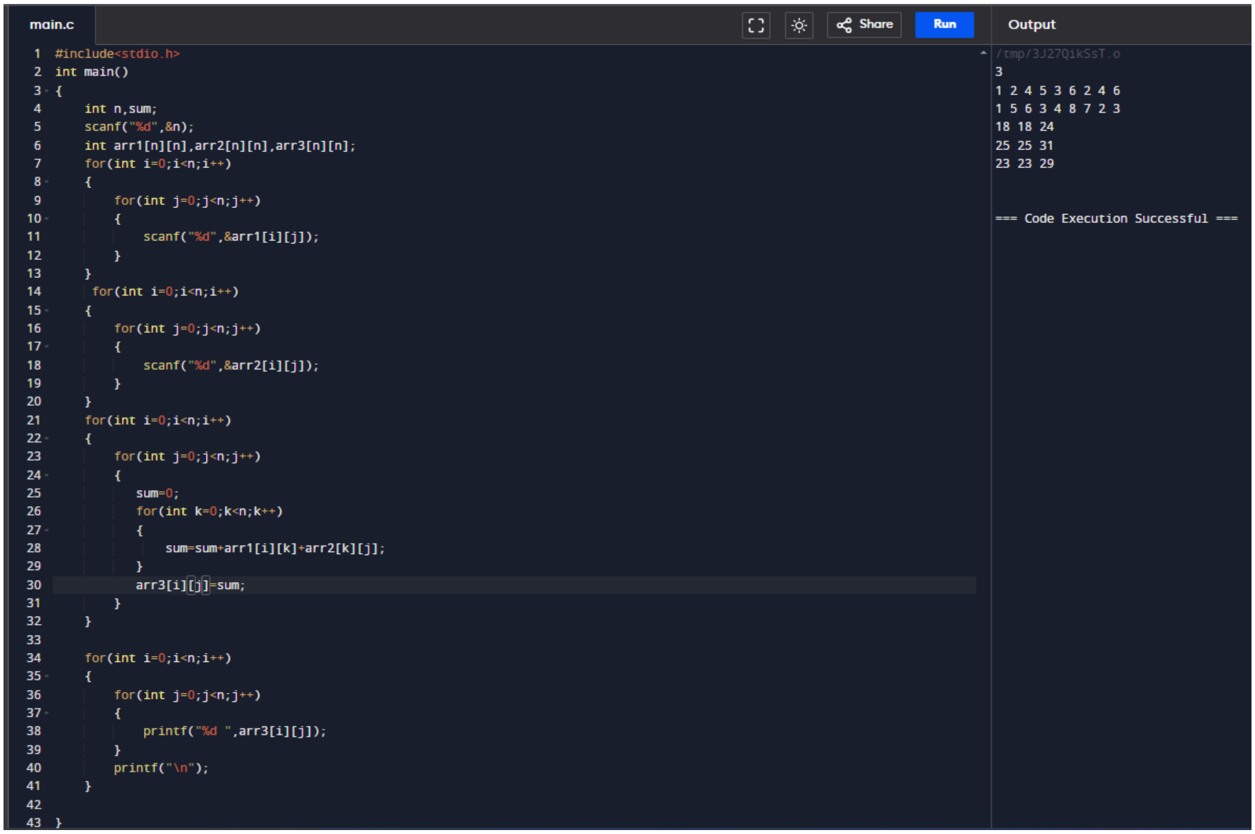
54.SCALAR MULTIPLICATION OF MATRIX



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr1[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr1[i][j]);
        }
    }

  int num;
  scanf("%d",&num);
  for(int i=0;i<n;i++)
  {
      for(int j=0;j<n;j++)
      {
          arr1[i][j]=num*arr1[i][j];
      }
  }
   for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("%d ",arr1[i][j]);
        }
        printf("\n");
    }
}
```

```
/tmp/9Rof6uuySa.o
3
1 5 6 3 2 4 8 9 4
2
2 10 12
6 4 8
16 18 8


=== Code Execution Successful ===
```

55. Matrix multiplication

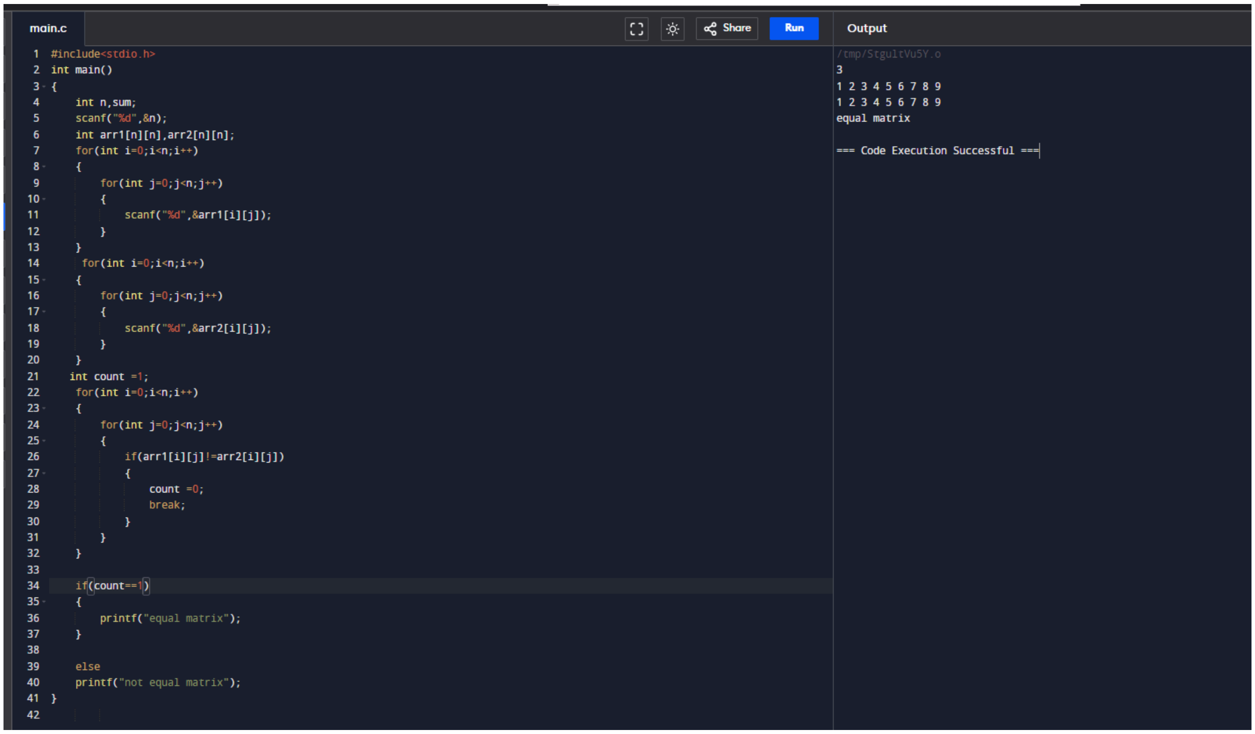


```c
#include<stdio.h>
int main()
{
    int n,sum;
    scanf("%d",&n);
    int arr1[n][n],arr2[n][n],arr3[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr1[i][j]);
        }
    }
     for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr2[i][j]);
        }
    }
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
           sum=0;
           for(int k=0;k<n;k++)
           {
               sum=sum+arr1[i][k]+arr2[k][j];
           }
           arr3[i][j]=sum;
        }
    }

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("%d ",arr3[i][j]);
        }
        printf("\n");
    }

}
```

```
/tmp/3J27QikSsT.o
3
1 2 4 5 3 6 2 4 6
1 5 6 3 4 8 7 2 3
18 18 24
25 25 31
23 23 29


=== Code Execution Successful ===
```

56. Equal or unequal matrix

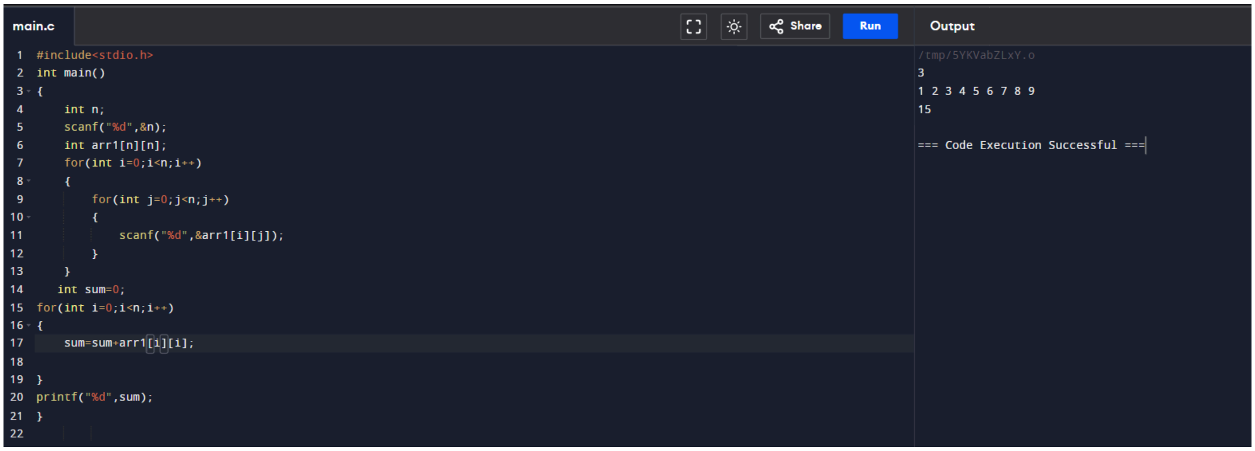
„.



```c
#include<stdio.h>
int main()
{
    int n,sum;
    scanf("%d",&n);
    int arr1[n][n],arr2[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr1[i][j]);
        }
    }
     for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr2[i][j]);
        }
    }
   int count =1;
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(arr1[i][j]!=arr2[i][j])
            {
                count =0;
                break;
            }
        }
    }

    if(count==1)
    {
        printf("equal matrix");
    }

    else
    printf("not equal matrix");
}
```

```
/tmp/StgultVu5Y.o
3
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
equal matrix

=== Code Execution Successful ===
```

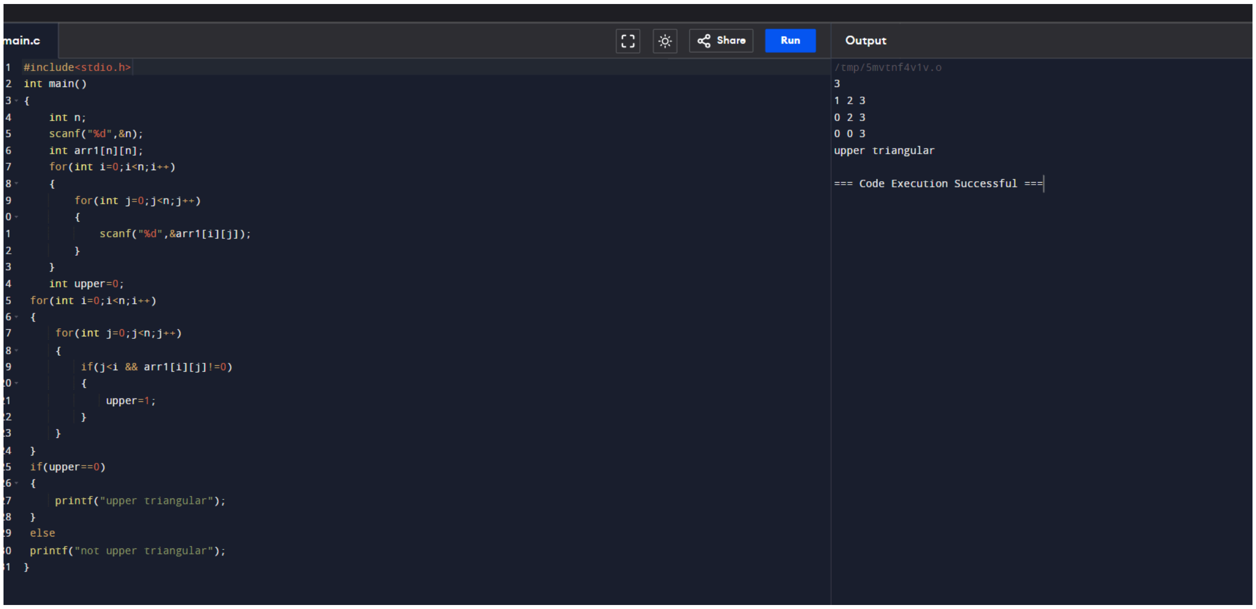
57.SUM OF MAIN DIAGONAL



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr1[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr1[i][j]);
        }
    }
   int sum=0;
for(int i=0;i<n;i++)
{
    sum=sum+arr1[i][i];

}
printf("%d",sum);
}
```

```
/tmp/5YKVabZLxY.o
3
1 2 3 4 5 6 7 8 9
15

=== Code Execution Successful ===
```

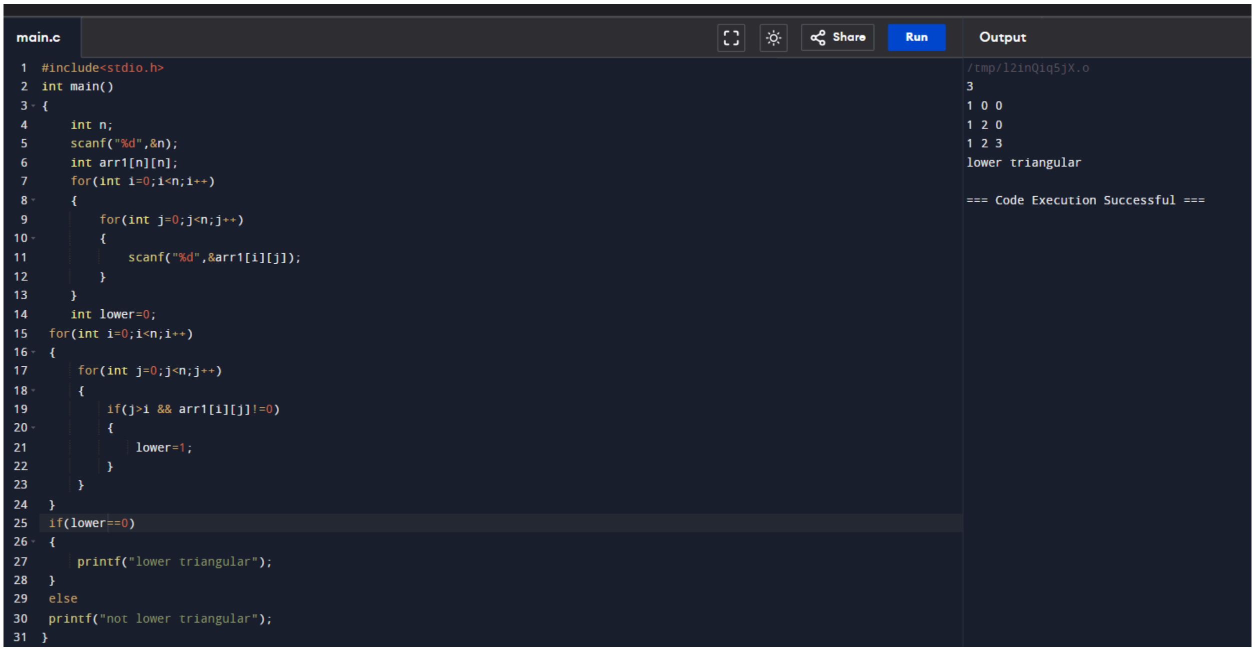
58.UPPER TRIANGULAR MATRIX



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr1[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr1[i][j]);
        }
    }
    int upper=0;
 for(int i=0;i<n;i++)
 {
     for(int j=0;j<n;j++)
     {
         if(j<i && arr1[i][j]!=0)
         {
             upper=1;
         }
     }
 }
 if(upper==0)
 {
     printf("upper triangular");
 }
 else
 printf("not upper triangular");
}
```

```
/tmp/5mvtnf4v1v.o
3
1 2 3
0 2 3
0 0 3
upper triangular

=== Code Execution Successful ===
```

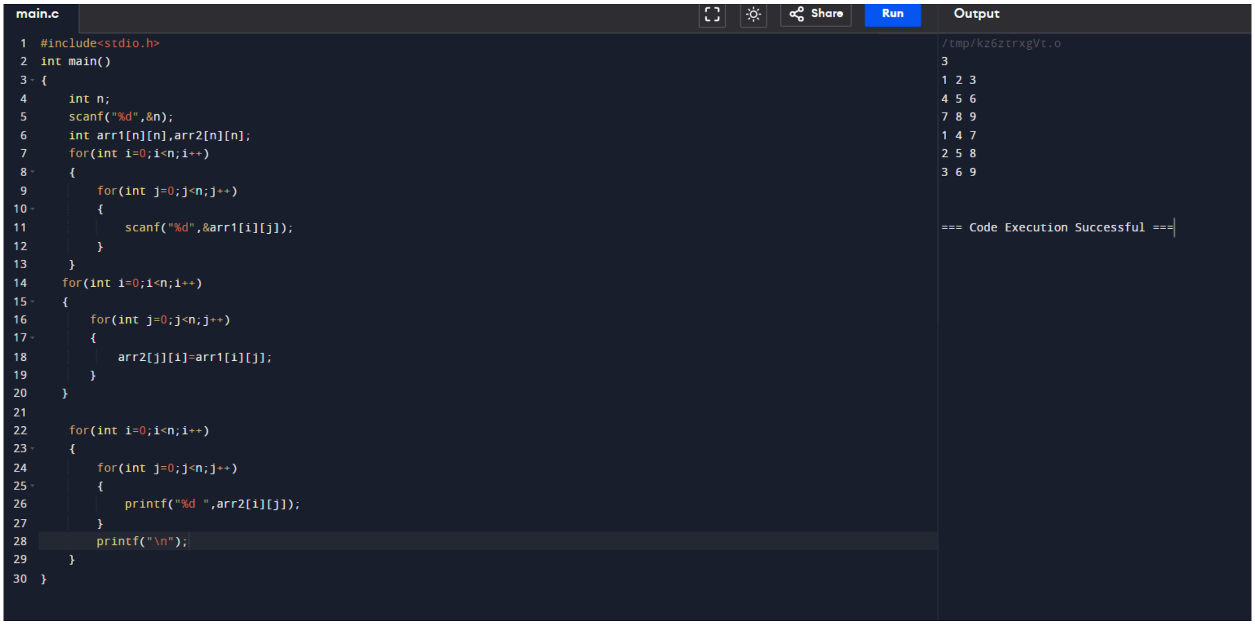
59.LOWER TRIANGULAR MATRIX



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr1[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr1[i][j]);
        }
    }
    int lower=0;
 for(int i=0;i<n;i++)
 {
     for(int j=0;j<n;j++)
     {
         if(j>i && arr1[i][j]!=0)
         {
             lower=1;
         }
     }
 }
 if(lower==0)
 {
     printf("lower triangular");
 }
 else
 printf("not lower triangular");
}
```

```
/tmp/l2inQiq5jX.o
3
1 0 0
1 2 0
1 2 3
lower triangular

=== Code Execution Successful ===
```

60.TRANSPOSE OF MATRIX



```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr1[n][n],arr2[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr1[i][j]);
        }
    }
   for(int i=0;i<n;i++)
   {
       for(int j=0;j<n;j++)
       {
           arr2[j][i]=arr1[i][j];
       }
   }

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("%d ",arr2[i][j]);
        }
        printf("\n");
    }
}
```

```
/tmp/kz6ztrxgVt.o
3
1 2 3
4 5 6
7 8 9
1 4 7
2 5 8
3 6 9


=== Code Execution Successful ===
```