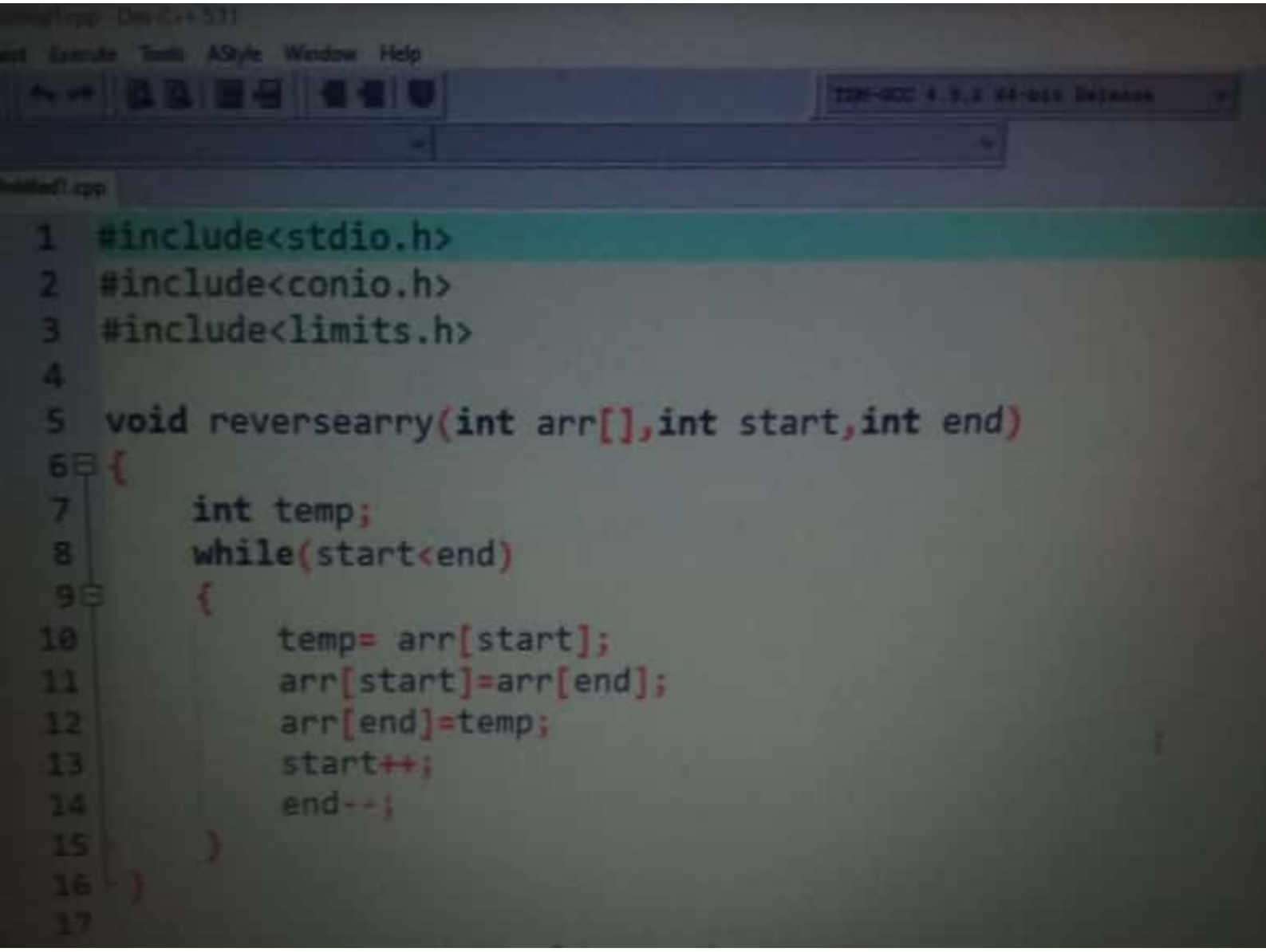
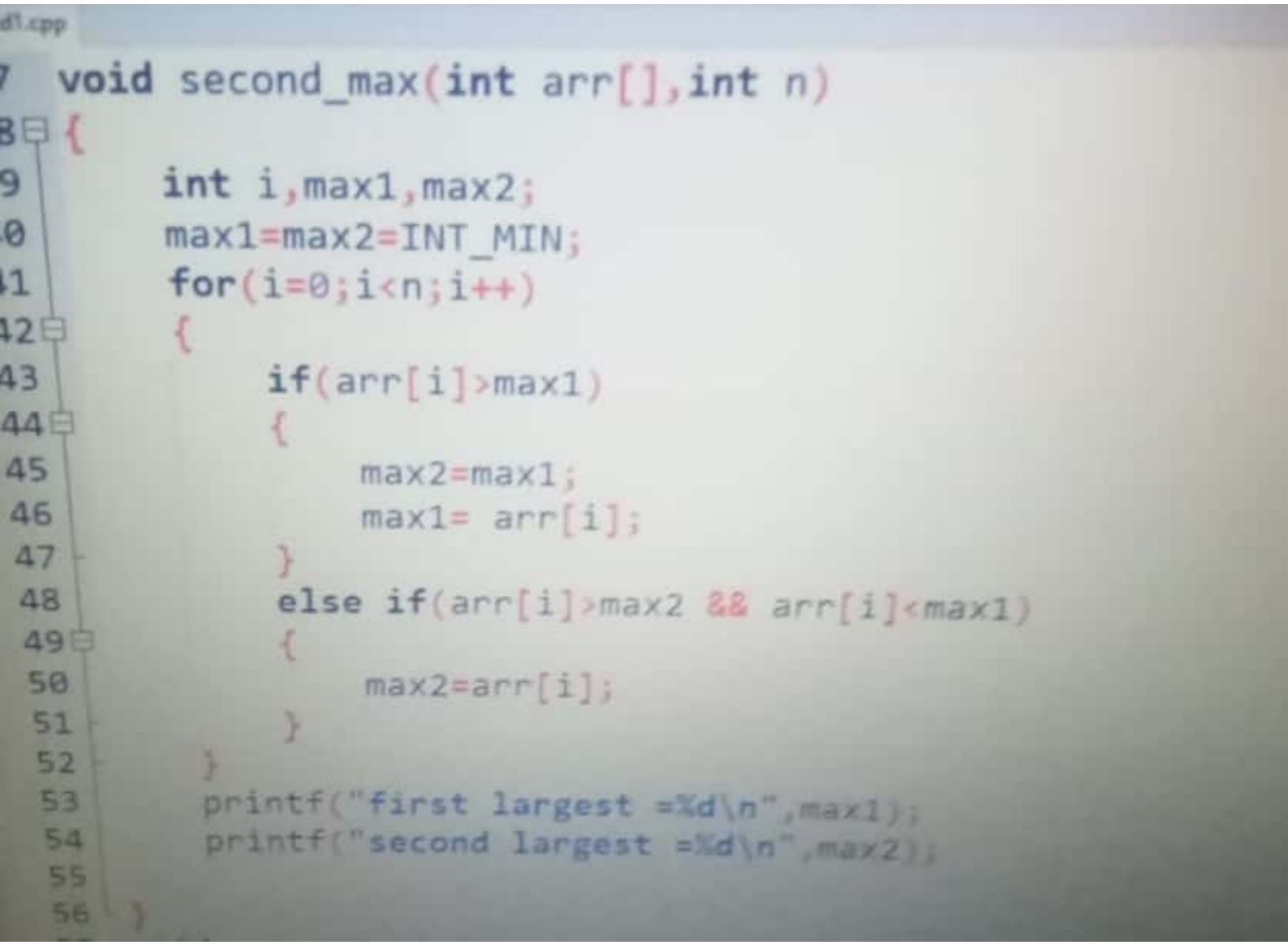
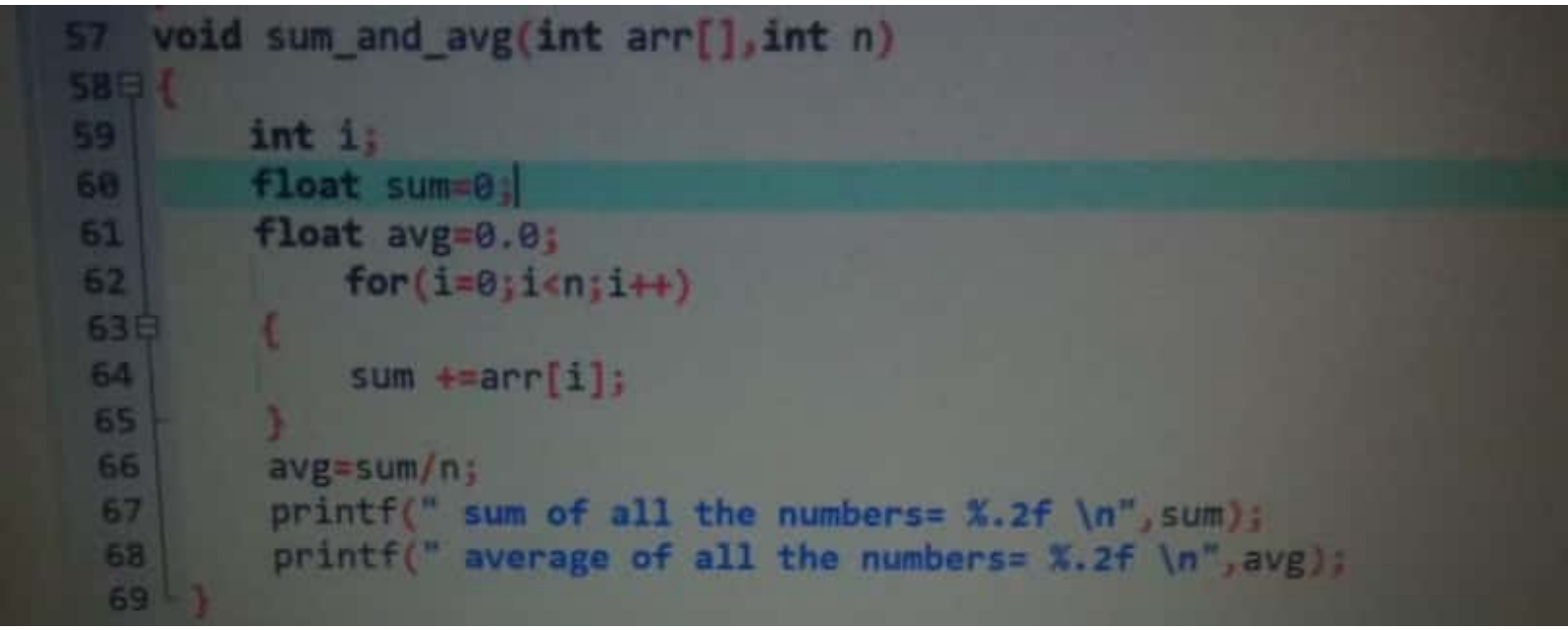


```
#include<stdio.h>
#include<conio.h>
#include<limits.h>

void reversearry(int arr[],int start,int end)
{
    int temp;
    while(start<end)
    {
        temp= arr[start];
        arr[start]=arr[end];
        arr[end]=temp;
        start++;
        end--;
    }
}
```

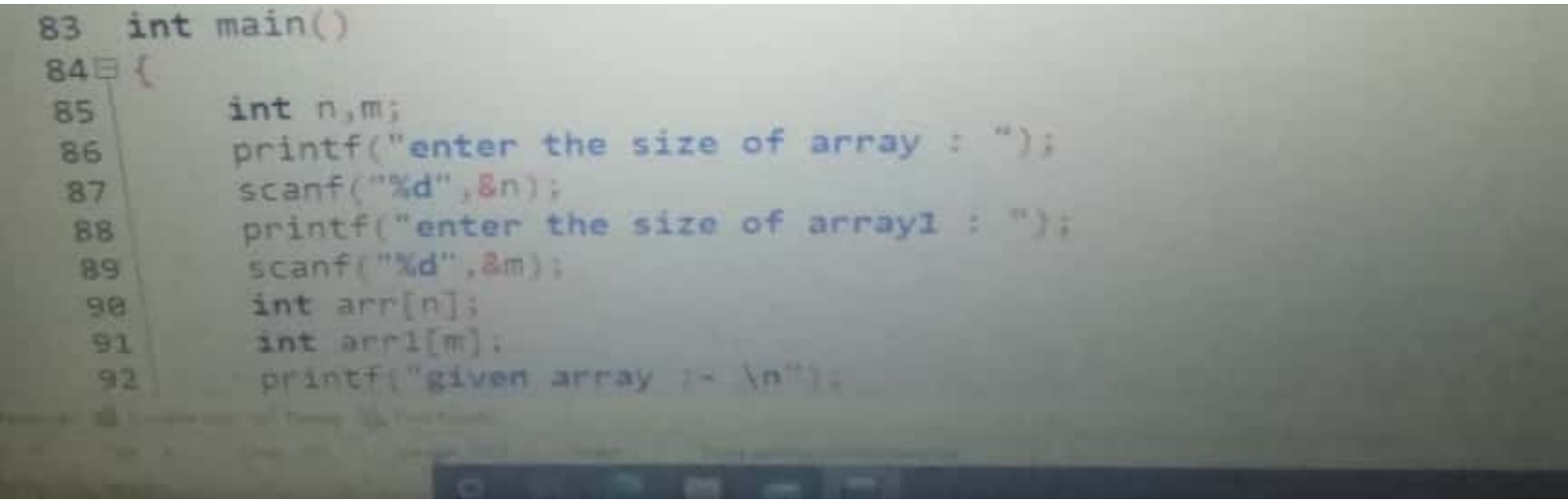
```
int remove_dublicate_elements(int arr[],int n)
{
    if(n==0||n==1)
    return n;
    int temp[n];
    int j=0;
    int i;

    for(i=0;i<n-1;i++)
    if(arr[i]!=arr[i+1])
    temp[j++]=arr[i];
    temp[j++]=arr[n-1];

    for(i=0;i<j;i++)
    arr[i]=temp[i];

    return j;

}
```





```
57  void sum_and_avg(int arr[],int n)
58  {
59      int i;
60      float sum=0;
61      float avg=0.0;
62          for(i=0;i<n;i++)
63      {
64          sum +=arr[i];
65      }
66      avg=sum/n;
67      printf(" sum of all the numbers= %.2f \n",sum);
68      printf(" average of all the numbers= %.2f \n",avg);
69  }
```

```
int disjoint_arrays(int arr[],int arr1[],int n,int m)
{
  int i,j;
  for(i=0;i<n;i++)
  {
      for(j=0;j<m;j++)
      {
        if(arr[i]==arr1[j])
          return -1;
      }
  }
  return 1;
}
```



```
83  int main()
84  {
85      int n,m;
86      printf("enter the size of array : ");
87      scanf("%d",&n);
88      printf("enter the size of array1 : ");
89      scanf("%d",&m);
90      int arr[n];
91      int arr1[m];
92      printf("given array :- \n");
```

```
int arr1[m];
printf("given array :- \n");
for(int i=0;i<n;i++)
{
    scanf("%d",&arr[i]);
}
printf("given array 2:- \n");
for(int i=0;i<m;i++)
{
    scanf("%d",&arr1[i]);
}
reversearry(arr,0,n-1);
printf("reverse array is :\n");
for(int i=0;i<n;i++)
{
    printf("%d \t",arr[i]);
}
printf("\n");
```

```
n=remove_dublicate_elements(arr, n);
printf("\n after removing dublicate element array is : \n");
for(int i=0;i<n;i++)
printf("%d \n",arr[i]);

 second_max(arr,n);

 sum_and_avg(arr,n);

 int res= disjoint_arrays(arr,arr1,n,m);
 if(res==-1)
 printf("\n arrays are not disjoint\n");
 else
 printf("\n arrays are disjoint \n");

 return 0;
}
```