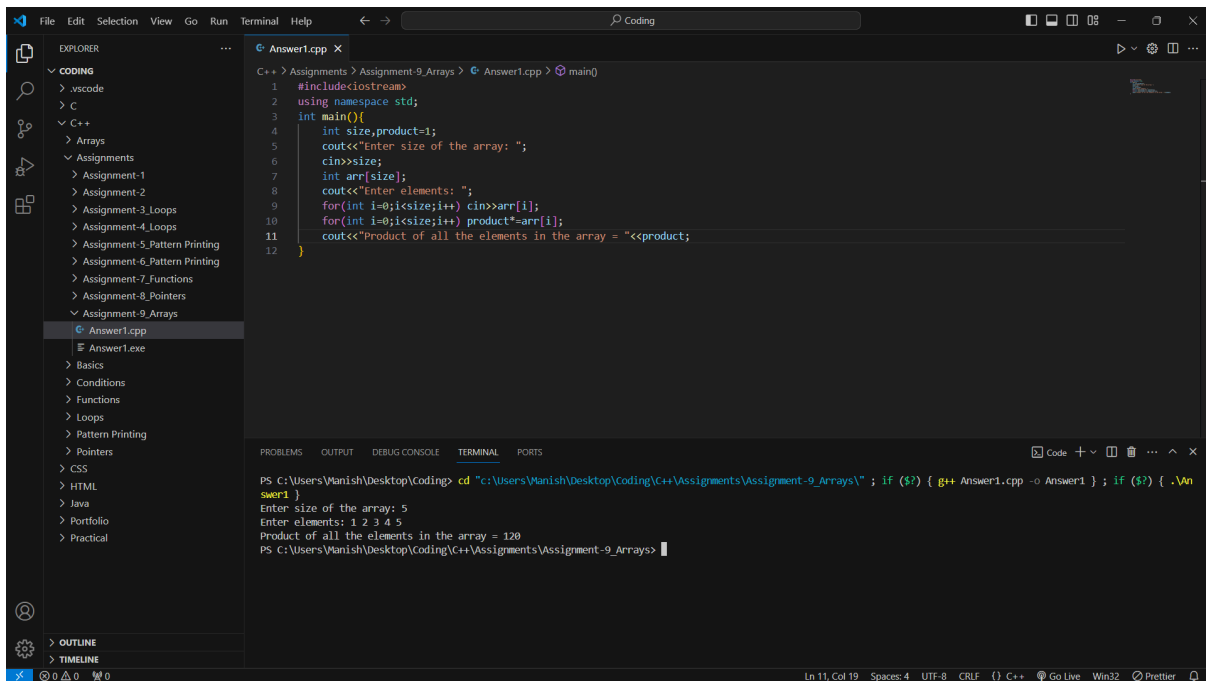


# Assignment (Arrays-1)

**Q1.** Calculate the product of all the elements in the given array.

**Answer:**

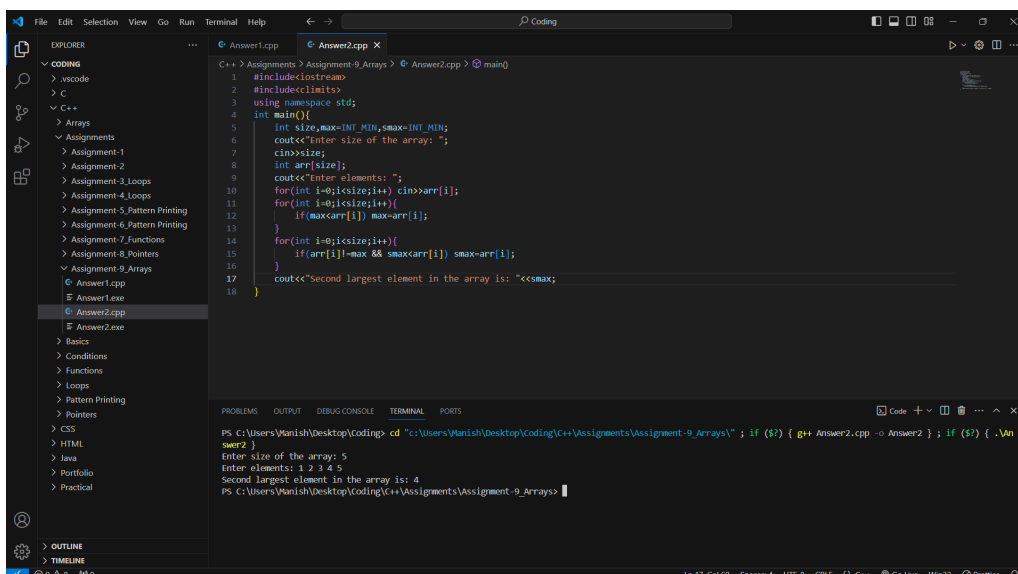


```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     int size,product=1;
5     cout<<"Enter size of the array: ";
6     cin>>size;
7     int arr[size];
8     cout<<"Enter elements: ";
9     for(int i=0;i<size;i++) cin>>arr[i];
10    for(int i=0;i<size;i++) product*=arr[i];
11    cout<<"Product of all the elements in the array = "<<product;
12 }
```

PS C:\Users\Vanish\Desktop\Coding> cd "C:\Users\Vanish\Desktop\Coding\C++\Assignments\Assignment-9\_Arrays" ; if (\$?) { g++ Answer1.cpp -o Answer1 } ; if (\$?) { .\Answer1.exe }  
Enter size of the array: 5  
Enter elements: 1 2 3 4 5  
Product of all the elements in the array = 120  
PS C:\Users\Vanish\Desktop\Coding\C++\Assignments\Assignment-9\_Arrays>

**Q2.** Find the second largest element in the given Array in one pass.

**Answer:**

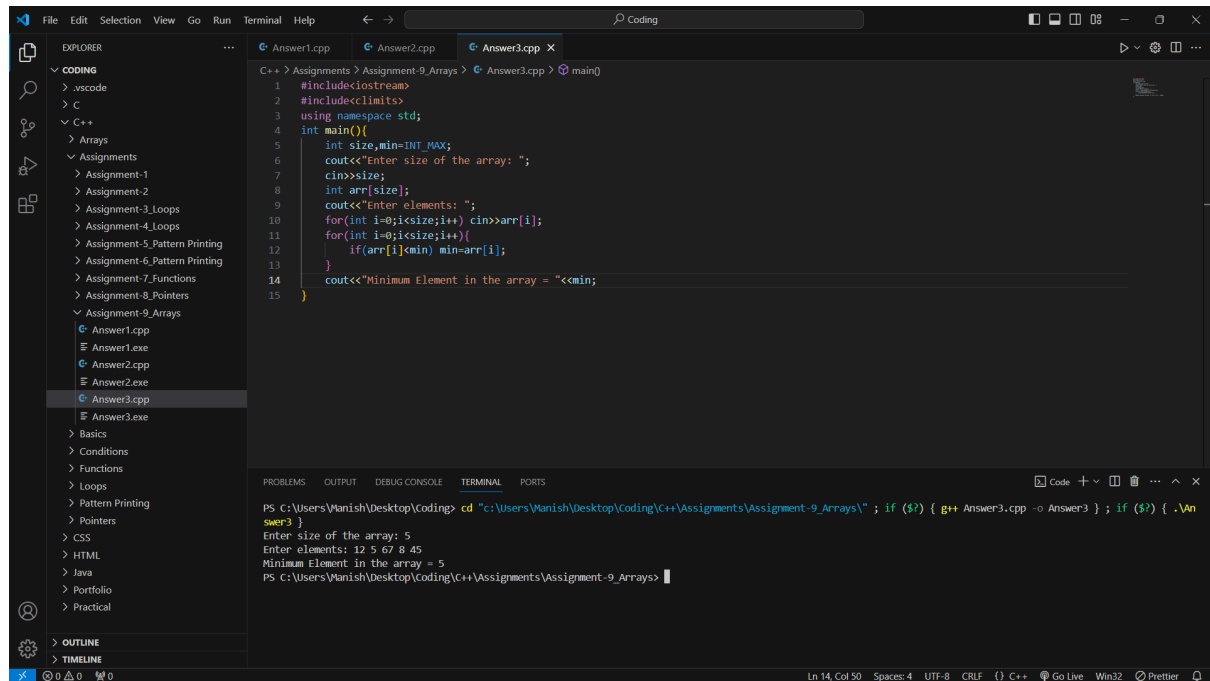


```
1 #include<iostream>
2 #include<limits>
3 using namespace std;
4 int main(){
5     int size,max=INT_MIN,smax=INT_MIN;
6     cout<<"Enter size of the array: ";
7     cin>>size;
8     int arr[size];
9     cout<<"Enter elements: ";
10    for(int i=0;i<size;i++) cin>>arr[i];
11    for(int i=0;i<size;i++){
12        if(max<arr[i]) max=arr[i];
13    }
14    for(int i=0;i<size;i++){
15        if(arr[i]!=max && smax<arr[i]) smax=arr[i];
16    }
17    cout<<"Second largest element in the array is: "<<smax;
18 }
```

PS C:\Users\Vanish\Desktop\Coding> cd "C:\Users\Vanish\Desktop\Coding\C++\Assignments\Assignment-9\_Arrays" ; if (\$?) { g++ Answer2.cpp -o Answer2 } ; if (\$?) { .\Answer2.exe }  
Enter size of the array: 5  
Enter elements: 1 2 3 4 5  
Second largest element in the array is: 4  
PS C:\Users\Vanish\Desktop\Coding\C++\Assignments\Assignment-9\_Arrays>

**Q3.** Find the minimum value out of all elements in the array.

**Answer:**



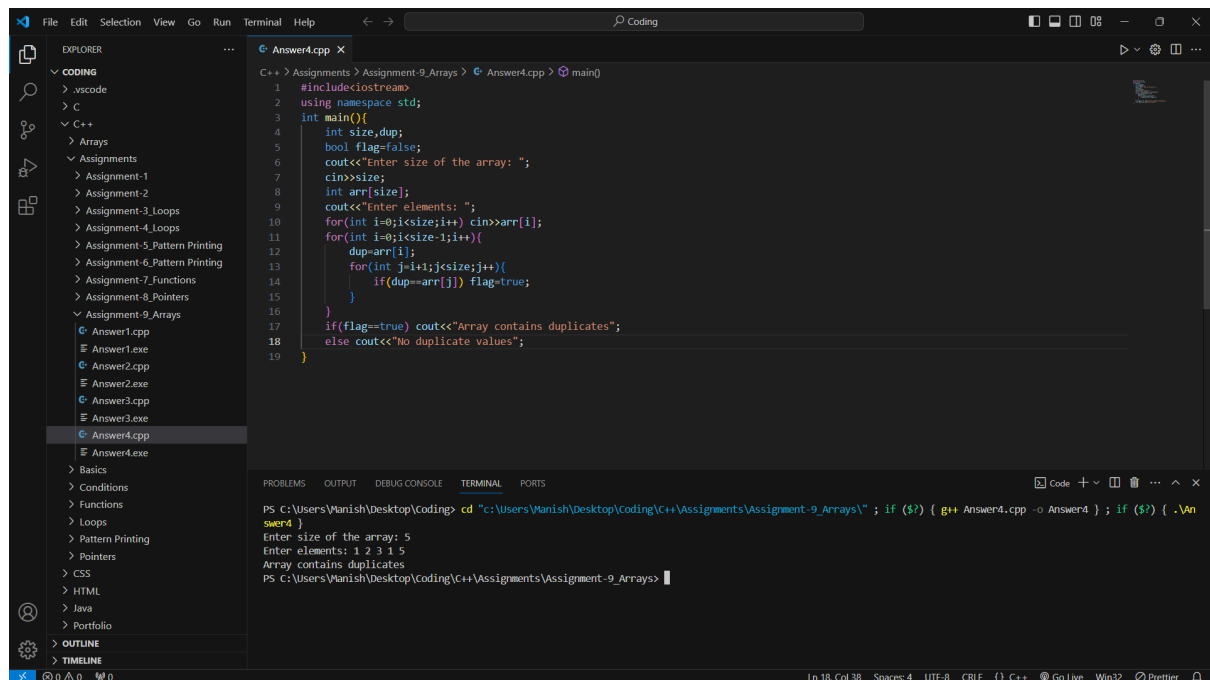
The screenshot shows a Visual Studio Code editor with a C++ file named `Answer3.cpp`. The code implements a function to find the minimum element in an array. The terminal output shows the program execution with the following steps:

```
PS C:\Users\Manish\Desktop\coding> cd "C:\Users\Manish\Desktop\coding\C++\Assignments\Assignment-9_Arrays\" ; if ($?) { g++ Answer3.cpp -o Answer3 } ; if ($?) { .\Answer3 }
Enter size of the array: 5
Enter elements: 12 5 67 8 45
Minimum Element in the array = 5
PS C:\Users\Manish\Desktop\coding\C++\Assignments\Assignment-9_Arrays>
```

```
1 #include<iostream>
2 #include<climits>
3 using namespace std;
4 int main(){
5     int size,min=INT_MAX;
6     cout<<"Enter size of the array: ";
7     cin>>size;
8     int arr[size];
9     cout<<"Enter elements: ";
10    for(int i=0;i<size;i++) cin>>arr[i];
11    for(int i=0;i<size;i++){
12        if(arr[i]<min) min=arr[i];
13    }
14    cout<<"Minimum Element in the array = "<<min;
15 }
```

**Q4.** Given an array, predict if the array contains duplicates or not.

**Answer:**



The screenshot shows a Visual Studio Code editor with a C++ file named `Answer4.cpp`. The code implements a function to check if an array contains duplicates. The terminal output shows the program execution with the following steps:

```
PS C:\Users\Manish\Desktop\coding> cd "C:\Users\Manish\Desktop\coding\C++\Assignments\Assignment-9_Arrays\" ; if ($?) { g++ Answer4.cpp -o Answer4 } ; if ($?) { .\Answer4 }
Enter size of the array: 5
Enter elements: 1 2 3 1 5
Array contains duplicates
PS C:\Users\Manish\Desktop\coding\C++\Assignments\Assignment-9_Arrays>
```

```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     int size,dup;
5     bool flag=false;
6     cout<<"Enter size of the array: ";
7     cin>>size;
8     int arr[size];
9     cout<<"Enter elements: ";
10    for(int i=0;i<size;i++) cin>>arr[i];
11    for(int i=0;i<size-1;i++){
12        dup=arr[i];
13        for(int j=i+1;j<size;j++){
14            if(dup==arr[j]) flag=true;
15        }
16    }
17    if(flag==true) cout<<"Array contains duplicates";
18    else cout<<"No duplicate values";
19 }
```

**Q5.** WAP to find the smallest missing positive element in the sorted Array that contains only positive elements.

**Q6:** Predict the output.

```
int main()
{
    int sub[50], i ;
    for ( i = 0 ; i ≤ 48 ; i++ ) ;
    {
        sub[i] = i ;
        cout<<sub[i]<<endl ;
    }
    return 0;
}
```

**Answer:**

49