

NAME : AJITHKUMAR S

PHONE NO : 8838362439

**PROBLEM STATEMENT :
HATFD1025**

Find the Second Largest Element in an Array

Write a program to find the second-largest element in an array of integers without using any sorting

algorithms or built-in array functions.

Instructions: Traverse the array manually to find both the largest and second-largest elements

PROGRAM CODE :

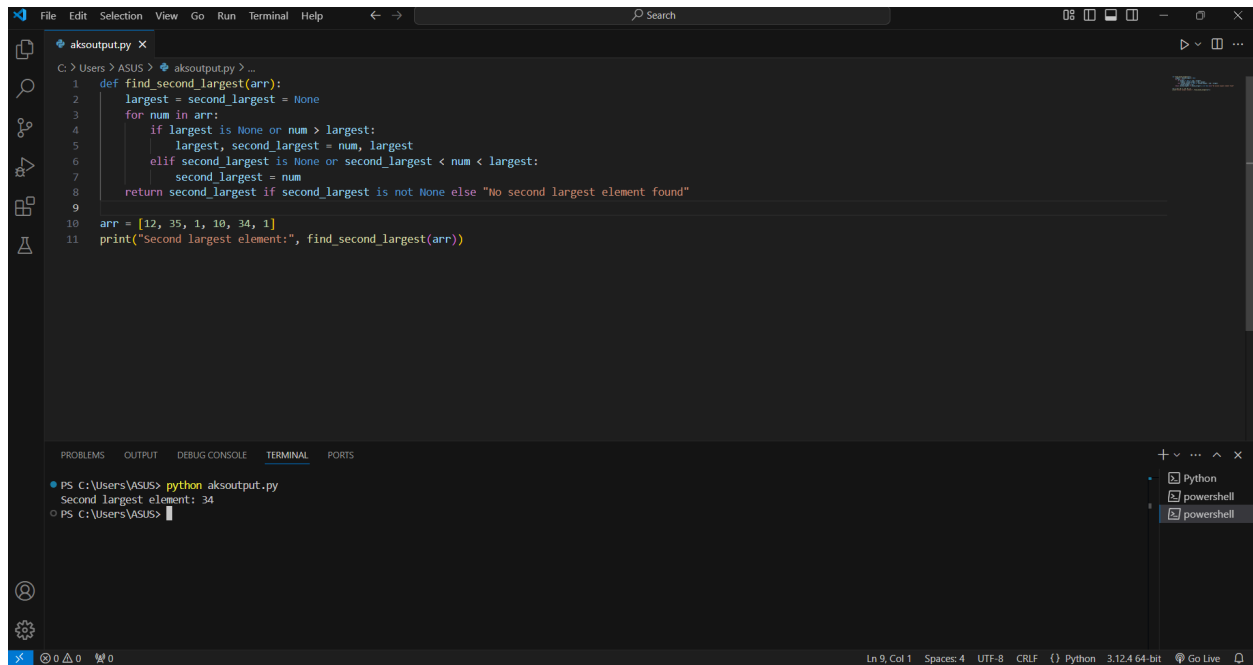
1. SAMPLE INPUT

```
def find_second_largest(arr):  
    largest = second_largest = None  
  
    for num in arr:  
        if largest is None or num > largest:  
            second_largest = largest  
            largest = num  
  
        elif second_largest is None or num > second_largest:  
            if num != largest:  
                second_largest = num  
  
    if second_largest is None:  
        return "No second largest element found"
```

```
    return second_largest
```

```
arr = [12, 35, 1, 10, 34, 1]
```

```
print("Second largest element:", find_second_largest(arr))
```



The screenshot shows a Visual Studio Code editor window with a file named 'aksoutput.py'. The code defines a function 'find_second_largest(arr)' that iterates through the array to find the second largest element. It initializes 'largest' and 'second_largest' to 'None'. For each number in the array, it checks if it's greater than the current 'largest'. If so, it updates 'largest' and 'second_largest'. If it's between 'second_largest' and 'largest', it updates 'second_largest'. After the loop, it returns 'second_largest' if it's not 'None', otherwise it returns a message. The array is defined as [12, 35, 1, 10, 34, 1], and the function is called, printing the second largest element, which is 34. The terminal at the bottom shows the command 'python aksoutput.py' and the output 'Second largest element: 34'.

```
1 def find_second_largest(arr):
2     largest = second_largest = None
3     for num in arr:
4         if largest is None or num > largest:
5             largest, second_largest = num, largest
6         elif second_largest is None or second_largest < num < largest:
7             second_largest = num
8     return second_largest if second_largest is not None else "No second largest element found"
9
10 arr = [12, 35, 1, 10, 34, 1]
11 print("Second largest element:", find_second_largest(arr))
```

PS C:\Users\ASUS> python aksoutput.py
Second largest element: 34
PS C:\Users\ASUS>

2.SAMPLE INPUT

```
def find_second_largest(arr):
```

```
    largest = second_largest = None
```

```
    for num in arr:
```

```
        if largest is None or num > largest:
```

```
            second_largest = largest
```

```
            largest = num
```

```
        elif second_largest is None or num > second_largest:
```

```
            if num != largest:
```

```
        second_largest = num
```

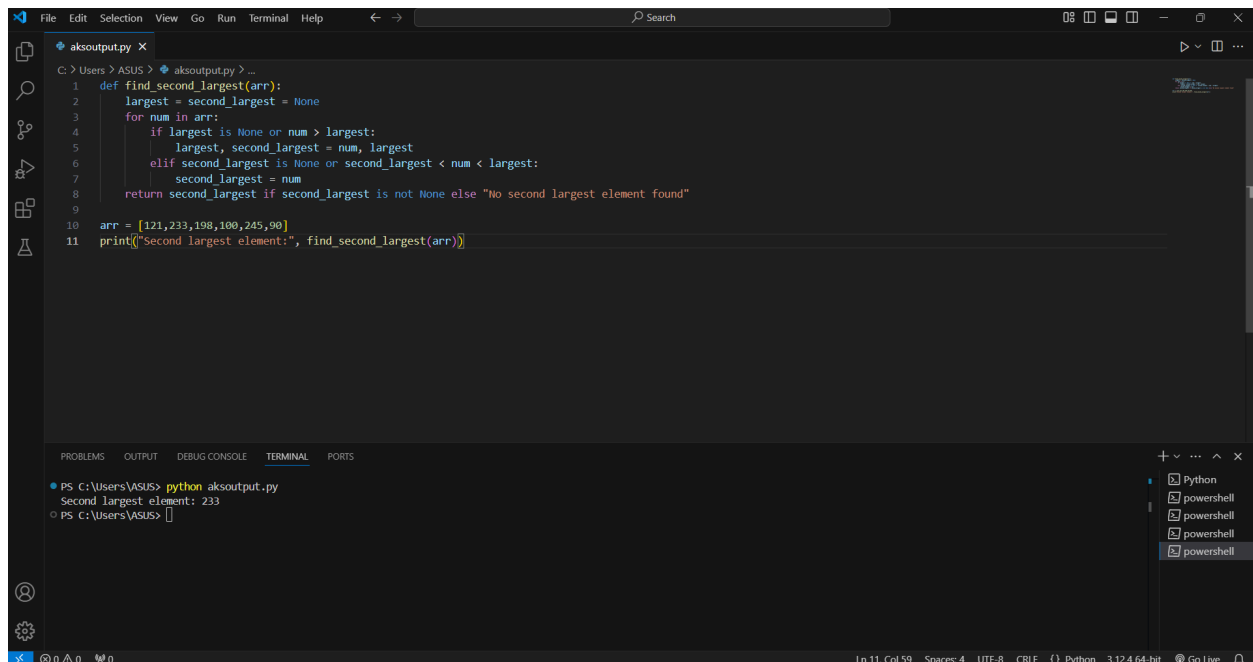
```
    if second_largest is None:
```

```
        return "No second largest element found"
```

```
    return second_largest
```

```
arr = [121,233,198,100,245,90]
```

```
print("Second largest element:", find_second_largest(arr))
```



The screenshot shows a Visual Studio Code window with a file named 'aksoutput.py'. The code in the editor is as follows:

```
1 def find_second_largest(arr):
2     largest = second_largest = None
3     for num in arr:
4         if largest is None or num > largest:
5             largest, second_largest = num, largest
6         elif second_largest is None or second_largest < num < largest:
7             second_largest = num
8     return second_largest if second_largest is not None else "No second largest element found"
9
10 arr = [121,233,198,100,245,90]
11 print("Second largest element:", find_second_largest(arr))
```

Below the editor, the 'TERMINAL' panel shows the command 'python aksoutput.py' being executed, resulting in the output 'Second largest element: 233'.

3.SAMPLE INPUT

```
def find_second_largest(arr):
```

```
    largest = second_largest = None
```

```
    for num in arr:
```

```
        if largest is None or num > largest:
```

```
            second_largest = largest
```

```
largest = num
```

```
elif second_largest is None or num > second_largest:
```

```
    if num != largest:
```

```
        second_largest = num
```

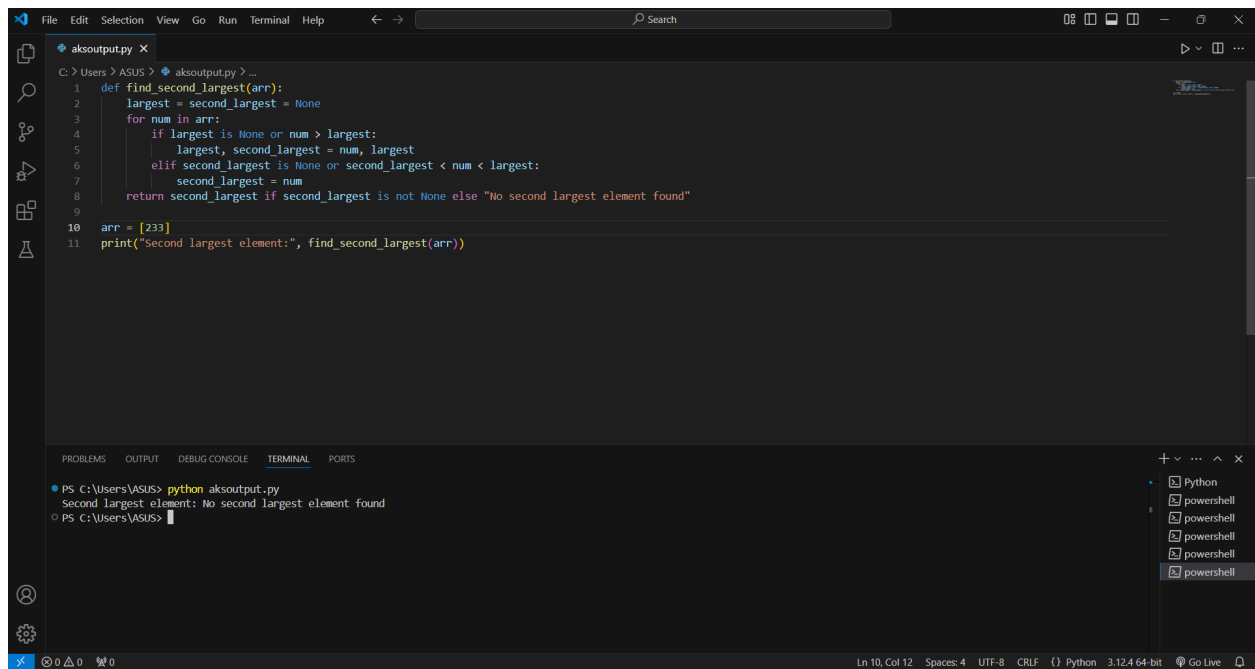
```
if second_largest is None:
```

```
    return "No second largest element found"
```

```
return second_largest
```

```
arr = [233]
```

```
print("Second largest element:", find_second_largest(arr))
```



The screenshot shows a Visual Studio Code editor window with a file named `aksoutput.py`. The code defines a function `find_second_largest(arr)` that iterates through the array to find the second largest element. It initializes `largest` and `second_largest` to `None`. For each number in the array, it checks if it's greater than the current `largest` or if it's between `second_largest` and `largest`. If found, it updates the values. After the loop, it returns `second_largest` if it's not `None`, otherwise it returns a string "No second largest element found". The array `arr` is set to `[233]`, and the function is called, with the result printed.

```
1 def find_second_largest(arr):
2     largest = second_largest = None
3     for num in arr:
4         if largest is None or num > largest:
5             largest, second_largest = num, largest
6         elif second_largest is None or second_largest < num < largest:
7             second_largest = num
8     return second_largest if second_largest is not None else "No second largest element found"
9
10 arr = [233]
11 print("Second largest element:", find_second_largest(arr))
```

The terminal output shows the command `python aksoutput.py` being executed, resulting in the message: `Second largest element: No second largest element found`.