1)Find a pair with the given sum in an array,Given an unsorted integer array, find a pair with the given sum in it.

Source Code:

```
def find(nums, target):
    seen = set()
    print(seen)
    for num in nums:
        complement = target - num
        if complement in seen:
            print("Pair found:", (complement, num))
        seen.add(num)
nums = []
target = int(input("Enter the target"))
n=int(input("Enter the number of elements:"))
for i in range(n):
    element=int(input("Enter the elements"))
    nums.append(element)
find(nums, target)
```

Output:

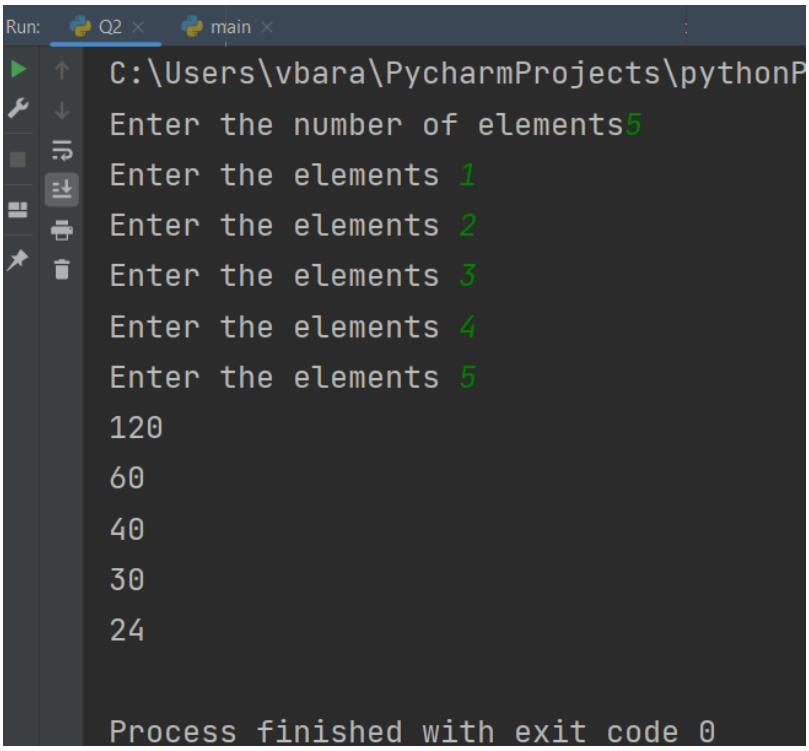
```
C:\Users\vbara\PycharmProjects\pythonProject2\venv
Enter the target10
Enter the number of elements:6
Enter the elements8
Enter the elements7
Enter the elements2
Enter the elements5
Enter the elements3
Enter the elements1
set()
Pair found: (8, 2)
Pair found: (7, 3)

Process finished with exit code 0
```

2) Given an integer array, replace each element with the product of every other element without using the division operator.

Source Code:

```
l=[]
n=int(input("Enter the number of elements"))
for i in range(n):
    elements=int(input("Enter the elements "))
    l.append(elements)
mul=1
for k in range(n):
    temp=l[k]
    for m in l:
        if m==temp:
            continue
        else:
            mul=mul*m
    print(mul)
    mul=1
```

Output:

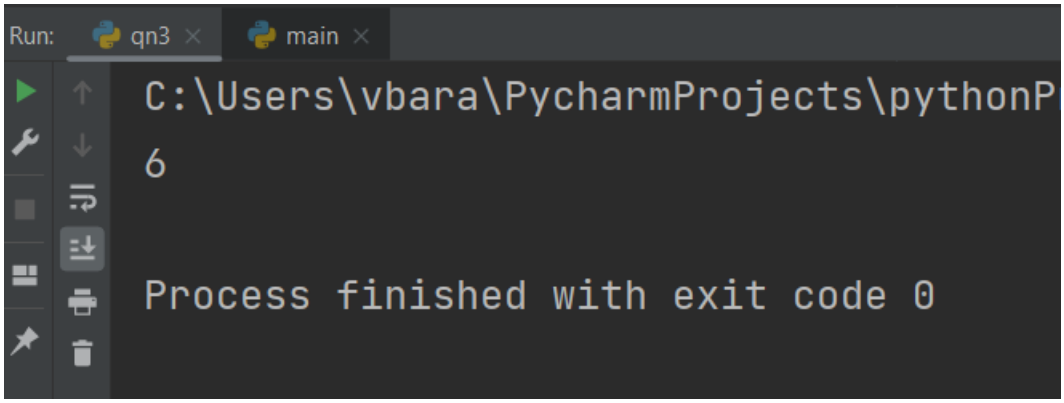

Run:
Q2
main
C:\Users\vbara\PycharmProjects\pythonP
Enter the number of elements5
Enter the elements 1
Enter the elements 2
Enter the elements 3
Enter the elements 4
Enter the elements 5
120
60
40
30
24
Process finished with exit code 0

3) Maximum Sum Circular Subarray. Given a circular integer array, find a subarray with the largest sum in it.

Source Code:

```
def fun(nums):
    total_sum = 0
    max_sum = 0
    current_max = 0
    min_sum = 0
    current_min = 0

    for num in nums:
        total_sum += num

        current_max = max(num, current_max + num)
        max_sum = max(max_sum, current_max)

        current_min = min(num, current_min + num)
        min_sum = min(min_sum, current_min)

    if total_sum == min_sum:
        return max_sum

    return max(max_sum, total_sum - min_sum)

input_array = [2, 1, -5, 4, -3, 1, -3, 4, -1]
result = fun(input_array)
print(result)
```

Output:


Run:
qn3
main
C:\Users\vbara\PycharmProjects\pythonP
6
Process finished with exit code 0

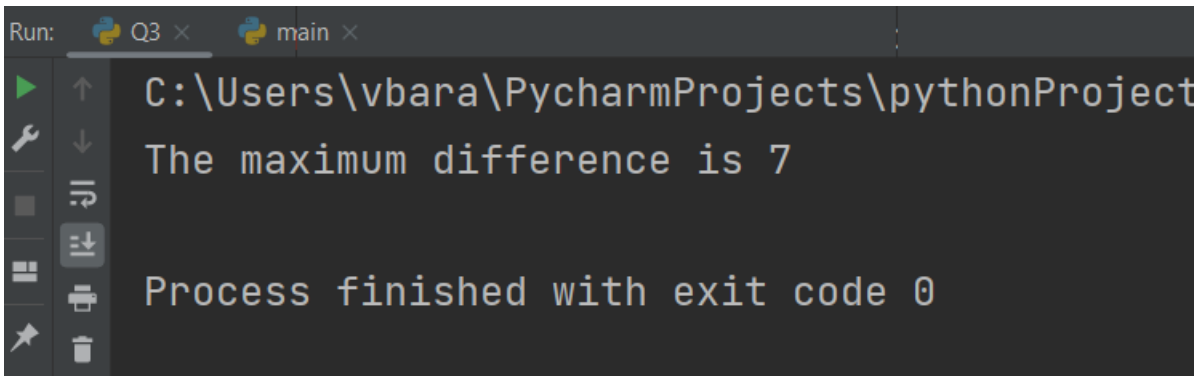
4) Find the maximum difference between two array elements that satisfies the given constraints

Given an integer array, find the maximum difference between two elements in it such that the smaller element appears before the larger element.

Source Code:

```
def max_difference(arr):

    min_element = arr[0]
    max_difference = arr[1] - arr[0]

    for num in arr:
        if num - min_element > max_difference:
            max_difference = num - min_element


        if num < min_element:
            min_element = num

    return max_difference


arr = [2, 7, 9, 5, 1, 3, 5 ]
result = max_difference(arr)
print("The maximum difference is",result)
```

Output:

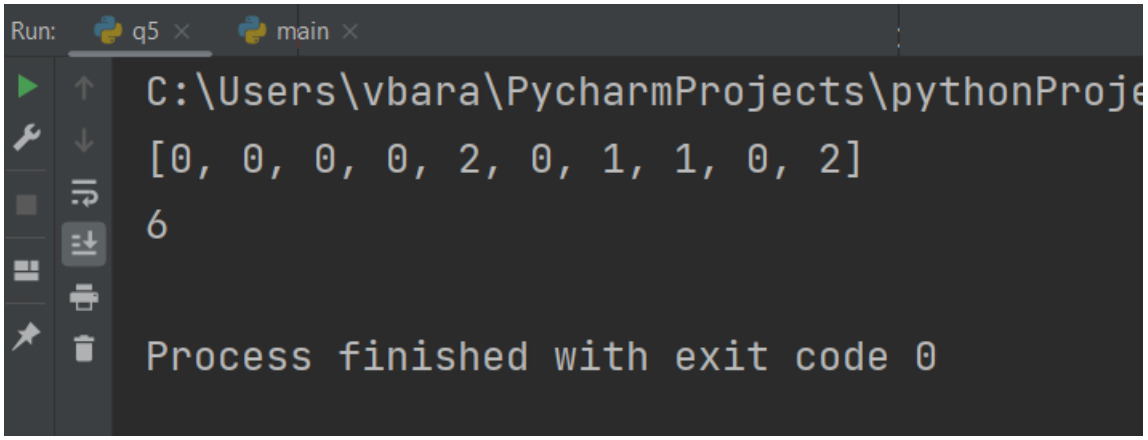

Run:
Q3
main
C:\Users\vbara\PycharmProjects\pythonProject
The maximum difference is 7
Process finished with exit code 0

5) Given an array of integers of size N, the task is to find the first non-repeating element in this array.

Source Code:

```
def repeating_element(arr):

    frequency_list = [0] * (max(arr) + 1)

    for num in arr:
        frequency_list[num] += 1
    print(frequency_list)
    for num in arr:
        if frequency_list[num] == 1:
            return num

    return None


arr = [9, 4, 9, 6, 7, 4]
result = repeating_element(arr)
print(result)
```

Output:



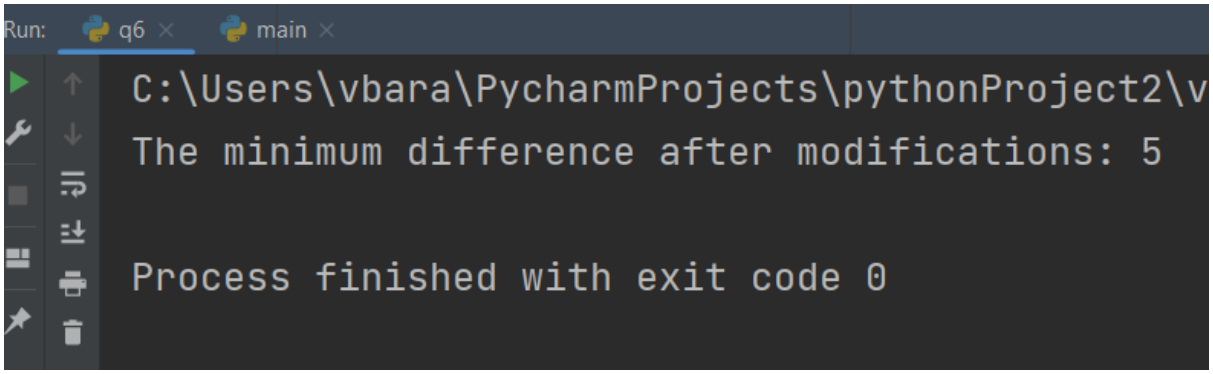
6) Minimize the maximum difference between the heights

Given the heights of N towers and a value of K, Either increase or decrease the height of every tower by K (only once) where K > 0. After modifications, the task is to minimize the difference between the heights of the longest and the shortest tower and output its difference.

Source Code:

```
def fun(heights, k):
    min_height = min(heights)
    max_height = max(heights)

    for i in range(len(heights)):
        if heights[i] - k >= min_height:
            heights[i] -= k
        elif heights[i] + k <= max_height:
            heights[i] += k


    new_min_height = min(heights)
    new_max_height = max(heights)


    return new_max_height - new_min_height

tower_heights = [1, 15, 10]
k_value = 6
result = fun(tower_heights, k_value)
print("The minimum difference after
modifications:", result)
```

Output:


Run:
q6
main
C:\Users\vbara\PycharmProjects\pythonProject2\v
The minimum difference after modifications: 5
Process finished with exit code 0