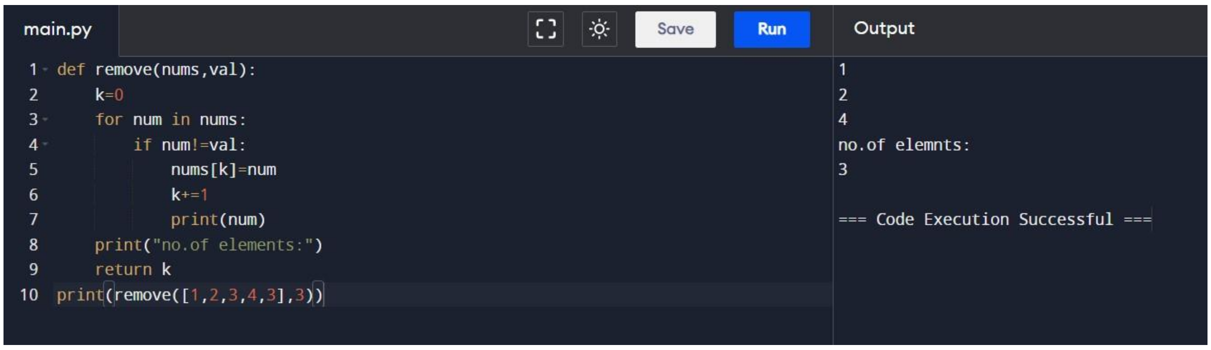
1.Remove element

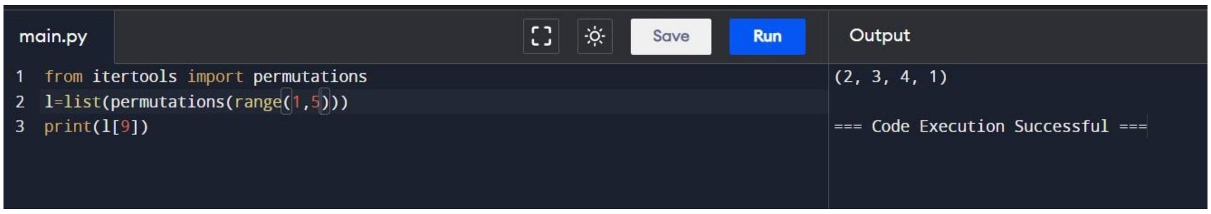


```python
def remove(nums,val):
    k=0
    for num in nums:
        if num!=val:
            nums[k]=num
            k+=1
            print(num)
    print("no.of elements:")
    return k
print(remove([1,2,3,4,3],3))
```

```
1
2
4
no.of elemnts:
3

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

Time complexity:O(n)

2.Permutations sequence

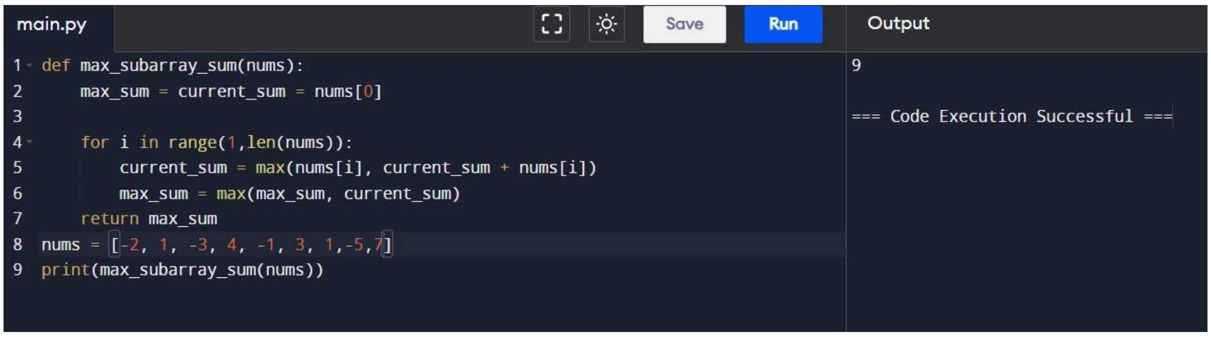


```python
from itertools import permutations
l=list(permutations(range(1,5)))
print(l[9])
```

```
(2, 3, 4, 1)

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

Time complexity:O(nlogn)

3.Maximum sub array

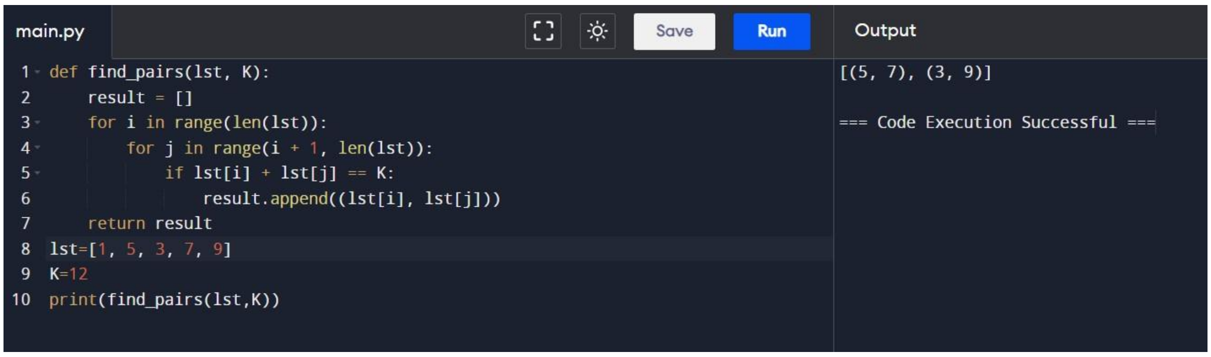


```python
def max_subarray_sum(nums):
    max_sum = current_sum = nums[0]

    for i in range(1,len(nums)):
        current_sum = max(nums[i], current_sum + nums[i])
        max_sum = max(max_sum, current_sum)
    return max_sum
nums = [-2, 1, -3, 4, -1, 3, 1,-5,7]
print(max_subarray_sum(nums))
```

```
9

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

Time complexity: O(n^2)

4.combination sum



```python
def find_pairs(lst, K):
    result = []
    for i in range(len(lst)):
        for j in range(i + 1, len(lst)):
            if lst[i] + lst[j] == K:
                result.append((lst[i], lst[j]))
    return result
lst=[1, 5, 3, 7, 9]
K=12
print(find_pairs(lst,K))
```

```
[(5, 7), (3, 9)]

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

Time complexity:O(nlogn)