1.Remove elements

```
def removeelements(nums,val):
    k=0
    for i in range(len(nums)):
        if nums[i]!=val:
            nums[k]=nums[i]
            k+=1
    return k
nums=[1,2,3,4,2]
val=2
k=removeelements(nums,val)
print(nums[:k])
```

2.Max subarray

```
def max_subarray(nums):
    max_sum=current_sum=nums[0]
    for num in nums[1:]:
        current_sum=max(num,current_sum+num)
        max_sum=max(max_sum,current_sum)
    return max_sum
nums=[-2,1,-3,4,-1,2,1,-5,4]
print(max_subarray(nums))
```

3.Permutations

```
import itertools
p = itertools.permutations([1, 1, 2])
unique = list(dict.fromkeys(list(p)))
output = [list(perm) for perm in unique]
print(output)
```

4.Permutations sequence

```
import math
def getPermutation(n, k):
    nums = [str(i) for i in range(1, n+1)]
    k -= 1
    res = ""
    while n > 0:
        n -= 1
        index = k // math.factorial(n)
        k %= math.factorial(n)
        res += nums.pop(index)
    return res
n = 3
k = 3
output = getPermutation(n, k)
print(output)
```

5.Count

```
def countAndSay(n):
    if n == 1:
        return "1"
    prev = countAndSay(n - 1)
    result = ""
    count = 1
    for i in range(len(prev)):
        if i + 1 < len(prev) and prev[i] == prev[i + 1]:
            count += 1
        else:
            result += str(count) + prev[i]
            count = 1
    return result
```

```
n = 1
print(countAndSay(n))
```

6.Suduko

```
def Sudoku(board):
    rows = [set() for i in range(9)]
    columns = [set() for i in range(9)]
    sub_boxes = [set() for i in range(9)]
    for i in range(9):
        for j in range(9):
            num = board[i][j]
            if num != '.':
                sub_box_index = (i // 3) * 3 + (j // 3)
                if (num in rows[i] or
                    num in columns[j] or
                    num in sub_boxes[sub_box_index]):
                    return False
                rows[i].add(num)
                columns[j].add(num)
                sub_boxes[sub_box_index].add(num)
    return True
board1 = [["5","3",".",".","7",".",".",".",".","."]
        ,["6",".",".","1","9","5",".",".",".","."]
        ,[".","9","8",".",".",".",".",".","6","."]
        ,["8",".",".",".",".","6",".",".",".","3"]
        ,["4",".",".","8",".","3",".",".",".","1"]
        ,["7",".",".",".",".","2",".",".",".","6"]
        ,[".","6",".",".",".",".",".","2","8","."]
        ,[".",".",".","4","1","9",".",".",".","5"]
        ,[".",".",".",".",".","8",".",".","7","9"]]
print(Sudoku(board1))
```

7.Combination sum

```
def combinationSum(candidates, target):
    dp = [[] for _ in range(target + 1)]
    dp[0] = [[]]
    for c in candidates:
        for i in range(c, target + 1):
            dp[i] += [comb + [c] for comb in dp[i - c]]
    return dp[target]
candidates = [2, 3, 6, 7]
target = 7
print(combinationSum(candidates, target))
```

8.Length of last word

```
s="hello world"
s1=s.split()
n=len(s1)
print(len(s1[n-1]))
```