

```
input="findig first non-repeating character from sentence"
```

```
def first_char(input):  
    for el in input:  
        if input.count(el) ==1:  
            return el  
    return none
```

```
result=(first_char(input))  
print(result)
```

```
input = ["navish", "duggu", "hibare"]  
output = []  
for el in input:  
    output.append((el, len(el)))  
print(output)
```

```
#####  
#####
```

```
input="how ae you? are u ok good"  
word= input.split()  
frequency={}
```

```
for el in word:  
    if el in frequency:  
        frequency[el]+=1  
    else:  
        frequency[el]=1  
print(frequency)
```

```
from collections import Counter  
input="how ae you? are u ok good"  
freq=Counter(input.split())  
print(freq)
```

```
c always capitals
```

```
#####  
#####
```

```
#group words that are anagrams of each other.
```

```
input_str = "eat,tan,tea,mat,bat"
```

```
words = input_str.split(',')  
anagram_groups = {}
```

```
for word in words:  
    key = ''.join(sorted(word))  
    if key in anagram_groups:  
        anagram_groups[key].append(word)  
    else:  
        anagram_groups[key] = [word]
```

```
output = list(anagram_groups.values())  
print(output)
```

```
data = [1, 2, 0, 3, 0, 4]
```

```
zeros = []  
non_zeros = []
```

```
for num in data:  
    if num == 0:  
        zeros.append(num)  
    else:  
        non_zeros.append(num)
```

```
result = zeros + non_zeros  
print(result)
```

```

data = ['0', '2', 'a', 'b', '1', 'g', '4', 'b', 'm', '3']

numbers = []
non_numbers = []

# Separate numbers (including '0') and non-numbers
for item in data:
    if item.isdigit(): # This includes '0'
        numbers.append(int(item))
    else:
        non_numbers.append(item)

# Sort numeric values
numbers.sort()

# Combine and print
result = numbers + non_numbers
print(result)

```

```

numbers = (1, 8, 3, 6, 2, 7)

for i in range(len(numbers)):
    for j in range(i + 1, len(numbers)):
        if numbers[i] + numbers[j] == 9:
            print((numbers[i], numbers[j]))

matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
rotated = [list(row) for row in zip(*matrix[::-1])]
print(rotated)

# Step 1: Transpose the matrix
n = len(matrix)
for i in range(n):
    for j in range(i + 1, n):
        matrix[i][j], matrix[j][i] = matrix[j][i], matrix[i][j]
# Step 2: Reverse each row
for row in matrix:
    row.reverse()
# Print rotated matrix
for row in matrix:
    print(row)

```

```
records = [("Alice", 85), ("Bob", 90), ("Charlie", 88)]
```

```
top_name = ""  
top_score = 0
```

```
for name, score in records:  
    if score > top_score:  
        top_score = score  
        top_name = name
```

```
print(top_name)
```

```
lst = [1, 2, 2, 3, 1, 4]
```

```
seen = set()  
result = []
```

```
for num in lst:  
    if num not in seen:  
        result.append(num)  
        seen.add(num)
```

```
print(result)
```

```
sentence = "python is great and python is easy"
```

```
words = sentence.split()  
freq = {}
```

```
for word in words:  
    if word in freq:  
        freq[word] += 1  
    else:  
        freq[word] = 1
```

```
print(freq)
```

```
list1 = [1, 2, 3, 4]
list2 = [3, 4, 5, 6]
common = []

for item in list1:
    if item in list2 and item not in common:
        common.append(item)

print(common)
```

```
# Print Armstrong numbers between 1 and 1000

print("Armstrong numbers between 1 and 1000 are:")

for num in range(1, 1001):
    order = len(str(num)) # Number of digits
    sum_of_powers = 0



    # Use for loop over each digit (converted to string)
    for digit in str(num):
        sum_of_powers += int(digit) ** order

    if num == sum_of_powers:
        print(num, end=' ')
```

```
# Check if the entered number is an Armstrong number
num = int(input("Enter a number: "))

order = len(str(num))
sum_of_powers = 0

for digit in str(num): # iterate over each digit
    sum_of_powers += int(digit) ** order

if num == sum_of_powers:
    print(f'{num} is an Armstrong number )
else:
    print(f'{num} is NOT an Armstrong number )
```

```
# Number of terms
n = 10

# First two terms
a, b = 0, 1

print("Fibonacci sequence:")
for _ in range(n):
    print(a, end=' ')
    a, b = b, a + b
```

```
# Print prime numbers up to n
n = 50

print("Prime numbers up to", n, "are:")

for num in range(2, n + 1):
    is_prime = True
    for i in range(2, int(num**0.5) + 1): # Check divisibility up to square root
        if num % i == 0:
            is_prime = False
            break
    if is_prime:
        print(num, end=' ')
```

```
output= [ 1,2,3,4,5,7,8]
```

```
result = []
for el in input:
    result.append(int(el))
print(result)
```

```
input=[2,3,4,5,6,7,8,9,12]

output = square of even index

result = []
```

```
n=len(input)
for el in range(n):
    if el % 2==0:
        result.append(input[el]**2)
    else:
        result.append(input[el])
print(result)
```

```
def flatten_list(nested):
    flat = []
    for item in nested:
        if isinstance(item, list):
            flat.extend(flatten_list(item))
        else:
            flat.append(item)
    return flat
```

```
# Example
data = [1, [2, [3, 4], 5], 6]
output = flatten_list(data)
print(output) # [1, 2, 3, 4, 5, 6]
```

```
input = ["navish", "hibare","duggu"]

# output = [("navish":5, "hibare",6,"duggu":5)]

output=[]
for elements in input:
    count=len(elements)
    output.append((elements,count))
print(output)
```

```
a=[(2,4),(4,6)]
b=[(3.5),(7.9)]
output = []

for (x1,y1), (x2,y2) in zip(a,b):
    output.append((x1+x2,y1+y2))
print(output)
```

```
a={'B' : 1, 'C' : 2, 'D' : 1 }
output={ 1 : [ 'B', 'D' ], 2 : [ 'C' ] }
```

```
output = {}
for key,value in a.items():
    if value not in output:
        output[value] =[]
    output[value].append(key)
print(output)
```

```
a={'A':1, 'B':{'X':10}}
b={'B':{'Y':20}, 'C':3 }

#S={'A':1, 'B':{'X':10,'Y':20},'C':3 }
```

```
S = a.copy()
```

```
for key, value in b.items():
    if key in S and isinstance(S[key], dict) and isinstance(value, dict):
        S[key].update(value)
    else:
        S[key] = value

print(S)
```