```
1]def isVowel(ch):
    if(ch=='a'or ch=='A'):
        print("the given ch is a vowel",ch)
    else:
        print("is not a vowel",ch)
ch=input("enter a alphabet :")
isVowel(ch)

2] def histogram(items):
    for n in items:
        output = ""
        times = n
        while times > 0:
            output += "*"
            times -= 1
        print(output)

histogram([4, 9, 7])

3] def fact(n):
    if n==1:
        return 1
    else:
        return(n*fact(n-1))
num=int(input("enter a no:"))
print("the factorial is:",fact(num))

4] file=open('read.txt','r')
text=file.readlines()
print(text[-1])
file.close()

5] file=open('read.txt','a+')
text=file.read()
print(text)
file.close()

6] def isPangram(sentence, alphabet):
    flag = True
    for char in alphabet:
        if char not in sentence.lower():
            flag = False
            break  # You should exit the loop once you find a missing character

    if flag:
        print("The sentence is a pangram")
    else:
        print("Not a pangram")

sentence = "the quick brown fox jumps over a lazy dog"
alphabet = "abcdefghijklmnopqrstuvwxyz"
isPangram(sentence, alphabet)
```

```
7] dict1={'a':1,'b':2}
total=sum(dict1.values())
print("dict1",dict1)
print("sum of values",total)

8] def is_palindrome(num):
    temp = num
    rev = 0

    while temp != 0:
        rem = temp % 10
        rev = rev * 10 + rem
        temp //= 10

    return num == rev

num = int(input("Enter a number:"))

if is_palindrome(num):
    print(num, "is a palindrome")
else:
    print(num, "is not a palindrome")

9] def common(list1, list2):
    for x in list1:
        for y in list2:
            if x == y:
                return True  # Use 'True' with a capital 'T'
    return False  # Use 'False' with a capital 'F'

list1 = [1, 2, 3, 4, 5]
list2 = [1, 3, 4, 6, 7]

print("list1", list1)
print("list2", list2)
print("Common in list1 and list2 is:", common(list1, list2))

10] def str_len(input_string):
    return len(input_string)
input_string="Donebro"
length=str_len(input_string)
print(length)

11] class parent1:
    def f1(self):
        print("parent11 class")
        class parent2:
            def f2(self):
                print("parent2")
                class child(parent1,parent2):
                    def f3(self):
                        print("child class")
                        t=child()
                        t.f1()
                        t.f2()
                        t.f3()
```

```
12] n = int(input("Enter the number of terms you want to print in the Fibonacci series:"))
n1, n2 = 0, 1
count = 0

if n <= 0:
    print("Enter a positive integer")
elif n == 1:
    print("Fibonacci series:")
    print(n1)
else:
    print("Fibonacci series:")
    while count < n:
        print(n1)
        nth = n1 + n2
        n1 = n2
        n2 = nth
        count += 1
```

```
13] try:
    n=int(input("enter a no:"))
    re=100/n
except(ValueError,ZeroDivisionError):
    print("wrong input")
else:
    print("Result:",re)
```

```
14] def clone(list):
    clone=list.copy()
    return clone
mylist=[3,3,4,5,6,7,8,]
print(mylist)
print("clone of my list:")
clone_list=clone(mylist)
print("mylist",clone_list)
```

```
15] dict1={'a':1,'b':2}
dict2={'c':3,'d':4}
print("dict-1",dict1)
print("dict-2",dict2)
concatenated_dict={**dict1,**dict2}
print("concatenated-dict",concatenated_dict)
```

```
16] class student:
    def info(self,name,city,number):
         print("name:",name,"city:",city,"number:",number)
obj=student()
obj.info("maaz","mumbai","7715814200")
```

```
17] dict1 = {'a': 1, 'b': 2, 'c': 4, 'd': 3}
asc = dict(sorted(dict1.items()))
dsc = dict(sorted(dict1.items(), reverse=True))
```

```
print("dict1", dict1)
print("asc", asc)
print("dsc", dsc)
```