In [5]:

```
def display(**kwargs):#creates in form of dictionary
    print(kwargs)
    print(type(kwargs))
     for k,v in kwargs.items():
        print(k,"=",v)
display(m1=100,m2=200,m3=300)
display(rollno=10,name="Arman",subject="python",marks=88)
```

```
{'m1': 100, 'm2': 200, 'm3': 300}
<class 'dict'>
m1 = 100
m2 = 200
m3 = 300
{'rollno': 10, 'name': 'Arman', 'subject': 'python', 'marks': 88}
<class 'dict'>
rollno = 10
name = Arman
subject = python
marks = 88
```

In [11]:

```
def display(**kwargs):#creates in form of dictionary
    print(kwargs)
    print(kwargs.values())
    print(type(kwargs))
    for k in kwargs.items():
        print(k)
display(m1=100,m2=200,m3=300)
display(rollno=10,name="Arman",subject="python",marks=88)
```

```
{'m1': 100, 'm2': 200, 'm3': 300}
dict_values([100, 200, 300])
<class 'dict'>
('m1', 100)
('m2', 200)
('m3', 300)
{'rollno': 10, 'name': 'Arman', 'subject': 'python', 'marks': 88}
dict_values([10, 'Arman', 'python', 88])
<class 'dict'>
('rollno', 10)
('name', 'Arman')
('subject', 'python')
('marks', 88)
```

Nested Function

In [18]:

```
def outer():
    print("Outer")
    def inner():
        print("Inner")
    inner()
outer()
```

```
Outer
Inner
```

In [20]:

```
def fun1():
    a=45
    def fun2():
        a=54
        print(a)
    fun2()
    print(a)
fun1()
```

```
54
45
```

In [25]:

```
for i in range(5):
    print(i)
else:
    print("Hello world")
```

```
0
1
2
3
4
Hello world
```

In [24]:

```
for i in range(5):
    if(i==3):
        break
    print(i)
else:
    print("Hello world")
```

```
0
1
2
```

In [30]:

```
i=1
while i<5:
    print(i)
    i+=1
else:
    print("Hi")
```

```
1
2
3
4
Hi
```

In [31]:

```
i=1
while i<5:
    if(i==2):
        break
    print(i)
    i+=1
else:
    print("Hi")
```

```
1
```

Q.WAP to check if the given number is Harshad or not

In [41]:

```
number=int(input("Enter a number:"))
temp=number

def harshad_num(n):
    sum=0
    while(n>0):
        sum+=(n%10)
        n//=10
    if(temp%sum==0):
        print("It is harshad number")
    else:
        print("Not a harshad number")
harshad_num(number)
```

```
Enter a number:81
It is harshad number
```

WAP that accept a single digit integer number and produces all possible six digit number for which the product of that digits is equal to the entered number

In [4]:

```
def sixdigit_num(number):
    if (number<1 or number>9):
        print("Enter between 1 and 9")
    else:
        for i in range (100000,1000000):
            product=1
            temp=i
            while temp>0:
                digit=temp%10
                product*=digit
                temp//=10
            if(product==number):
                print(i)
num=int(input("Enter a number:"))
sixdigit_num(num)
```

```
Enter a number:10
Enter between 1 and 9
```

WAP to get next date of a given date

if((year%4==0 and year%100!=0) or year%400==0):

In [1]:

```
def next_date(day,month,year):
    thirty1=(1,3,5,7,8,10,12)
    thirty=(4,6,9,11)
    if((day<0 or day>32)or(month<0 or month>12)):
        print("Enter valid date")
    elif((day==31)and(month==12)):
         print(f"Next date is:{1},{1},{year+1}")
    else:
        if((day==31)and(month in thirty1)):
            print(f"Next date is:{1},{month+1},{year}")
        if((day==30)and(month in thirty)):
            print(f"Next date is:{1},{month+1},{year}")
        if((day==28)and(month==2)):
             if((year%4==0 and year%100!=0) or year%400==0):
                    print(f"Next date is:{29},{month},{year}")
             else:
                print(f"Next date is:{1},{month+1},{year}")
        else:
             print(f"Next date is:{day+1},{month},{year}")

day=int(input("Enter day:"))
month=int(input("Enter month:"))
year=int(input("Enter year:"))
next_date(day,month,year)
```

```
Enter day:31
Enter month:12
Enter year:2024
Next date is:1,1,2025
```

Q.WAP to create a sequence where the first four members of sequence are equal to 1 and each successive term of the sequence is equal to the sum of the four previous ones find nth member of the sequence

In [4]:

```
n=int(input("Enter a number:"))
a=b=c=d=1
for i in range(1,n+1):
    if(i<=4):
        ans=1
    else:
        ans=a+b+c+d
        a=b
        b=c
        c=d
        d=ans
print(ans)
```

```
Enter a number:5
4
```

In [2]:

```
def nth_term(number):
    if number < 0:
        print("Please enter +ve number to find its term")
    if number <= 4:
        return 1
    sequence = [1,1,1,1]
    for i in range(4,number):
        next_term = sequence[i-1] + sequence[i-2] + sequence[i-3] + sequence[i-4]
        sequence.append(next_term)

    return sequence[number-1]

number = int(input("Enter Nth Term : "))
result = nth_term(number)
print("The Term is : ",result)
```

```
Enter Nth Term : 5
The Term is :  4
```