In [2]:
```
Q1. You are writing code for a company. The requirement of the company is that you
function that will check whether the password entered by the user is correct or no
take the password as input and return the string "Valid Password" if the entered pa
below-given password guidelines else it should return "Invalid Password".
Note: 1. The Password should contain at least two uppercase letters and at least tw
2. The Password should contain at least a number and three special characters.
3. The length of the password should be 10 characters long.




def check_password(password: str):
    special_chars = "[@_!#$%^&*()<>?/|}{~:]"

    cap = num = char = length = 0
    for i in password:
        if i.isupper() or i.islower():
            cap += 1
        elif i.isdigit():
            num += 1
        elif i in special_chars:
            char += 1
        else:
            continue

    if cap >= 4 and num >= 1 and char >= 3 and len(password) >= 10:
        return 'Valid Password'
    else:
        return 'Invalid Password'

pass1 = 'ABc123./@dx$#abx%^a2'
pass2 = 'adiu231$029AJY#-dj'
pass3 = 'ABcd12./#klajf23&'

print(pass1, ' = ', check_password(pass1))
print(pass2, ' = ', check_password(pass2))
print(pass3, ' = ', check_password(pass3))
```

```
ABc123./@dx$#abx%^a2  =  Valid Password
adiu231$029AJY#-dj  =  Invalid Password
ABcd12./#klajf23&  =  Valid Password
```

In [3]:
```
Q2. Solve the below-given questions using at least one of the following:
1. Lambda functioJ
2. Filter functioJ
3. Zap functioJ
4. List ComprehensioI
B Check if the string starts with a particular letterY
B Check if the string is numericY
B Sort a list of tuples having fruit names and their quantity. [("mango",99),("ora
B Find the squares of numbers from 1 to 10Y
B Find the cube root of numbers from 1 to 10Y
B Check if a given number is evenY
B Filter odd numbers from the given list.
[1,2,3,4,5,6,7,8,9,10-
B Sort a list of integers into positive and negative integers lists.
[1,2,3,4,5,6,-1,-2,-3,-4,-5,0]



sentence = 'I want to beacome a Data Scientist.'
```

```
def encrypt(str_: str) -> str:
    str_ = str_.lower()
    all_alpha = list('abcdefghijklmnopqrstuvwxyz')

    result = ''
    for i in str_:
        if i.isalpha():
            idx = all_alpha.index(i)
            result += all_alpha[-idx - 1]
        elif i.isspace():
            result += '$'
        else:
            result += i
    return result

print(f'{sentence = }')
print(f'{encrypt(sentence) = }')
```

```
sentence = 'I want to beacome a Data Scientist.'
encrypt(sentence) = 'r$dzmg$gl$yvzxlnv$z$wzgz$hxrvmgrhg.'
```

In [ ]: