# CS PRACTICAL List

**Ojas Mittal XII-K**

---

Q1

```
num = int(input('enter num: '))
digit = int(input('enter digit: '))

def count(n):
   return len(str(n))

def reverse(n):
   return int(str(n)[::-1])

def hasDigit(n,d):
   return str(d) in str(n)

print('hasdigit: ', hasDigit(num,digit))
print('count:', count(num), '\nreverse: ', reverse(num))
```

```
ojasmittal@pop-os ~/D/Code [1]> python3 q1.py
enter num: 2384658924450
enter digit: 1
hasdigit:  False
count: 13
reverse:  544298564832

ojasmittal@pop-os ~/D/Code> python3 q1.py
enter num: 947437439
enter digit: 9
hasdigit:  True
count: 9
reverse:  934734749
```

```
Q2
number = int(input('enter num: '))

def generateFactors(num):
    factors = []
    div = 1
    while div < num:
        if num % div == 0:
            factors.append(div)
        div += 1
    return factors

def isPrimeNo(num):
    if len(generateFactors(num)) < 2:
        return 'Prime'
    return 'Not Prime'

def isPerfectNo(num):
    if sum(generateFactors(num)) == num:
        return 'Perfect'
    return 'Not Perfect'

print(isPerfectNo(number), isPrimeNo(number))
```

```
ojasmittal@pop-os ~/D/Code> python3 q1.py
enter num: 56
Not Perfect Not Prime

ojasmittal@pop-os ~/D/Code> python3 q1.py
enter num: 6
Perfect Not Prime

ojasmittal@pop-os ~/D/Code> python3 q1.py
enter num: 17
Not Perfect Prime
```

```
Q3
def romanToInt(n):
   roman = {
       "I":1, "V":5, "X":10,
       "L":50, "C":100,
       "D":500, "M":1000
   }

   total = 0
   i = 0
   while i < len(n):
       if i + 1 < len(n) and roman[n[i]] < roman[n[i+1]]:
           total += roman[n[i+1]] - roman[n[i]]
           i += 2
       else:
           total += roman[n[i]]
           i += 1
   return total

num = input("Enter Roman numeral: ")
print(romanToInt(num))
```

```
ojasmittal@pop-os ~/D/Code (main)> python3 q3.py
Enter Roman numeral: MCMXIV
1914

ojasmittal@pop-os ~/D/Code (main)> python3 q3.py
Enter Roman numeral: MMMDCCCLXXXVIII
3888

ojasmittal@pop-os ~/D/Code (main) [1]> python3 q3.py
Enter Roman numeral: CD
400
```

```
Q4
num = int(input('enter decimal num: '))
con = input('B:binary H:Hex O:Octal\nenter conversion:')

def B(n):
   f = n
   binary = ''
   while f ⩾ 1:
       binary = binary + str(f%2)
       f = f//2
   return binary[::-1]

def H(n):
   f = n
   hex = ''
   j = ord('A')
   while f ⩾ 1:
       digit = f%16
       if digit > 9:
           digit = chr(j + digit - 10)
       hex = hex + str(digit)
       f = f//16
   return hex[::-1]

def O(n):
   f = n
   octal = ''
   while f ⩾ 1:
       octal = octal + str(f%8)
       f = f//8
   return octal[::-1]

def joinList(list):
   string = ''
   for i in list:
```

```
        string = string + i
    return string

dict = {
    'B': B(num),
    'H': H(num),
    'O': O(num)
}

print(joinList(dict[con]))
```

```
ojasmittal@pop-os ~/D/Code> python3 q4.py
enter decimal num: 999
B:binary H:Hex O:Octal
enter conversion:H
3E7

ojasmittal@pop-os ~/D/Code> python3 q4.py
enter decimal num: 999
B:binary H:Hex O:Octal
enter conversion:O
1747

ojasmittal@pop-os ~/D/Code> python3 q4.py
enter decimal num: 999
B:binary H:Hex O:Octal
enter conversion:B
1111100111
```

Q5

```
matrix = eval(input('enter matrix: '))
rows = int(input('enter r: '))
columns = int(input('enter c: '))

def reshape(mat,r,c):
    ro = len(mat)
```

```
    co = len(mat[0])
    if ro*co ≠ r*c:
        return 'Invalid Dimensions'

    new_mat = []
    flat = []
    for i in range(len(mat)):
        for j in mat[i]:
            flat.append(j)

    index = 0
    for i in range(r):
        new_mat.append([])
        for j in range(c):
            new_mat[i].append(flat[index])
            index += 1

    return new_mat

print(reshape(matrix,rows,columns))
```

```
ojasmittal@pop-os ~/D/Code> python3 q5.py
enter matrix: [[13, 14, 15], [16, 17, 18], [19, 20, 21],
[22, 23, 24]]
enter r: 2
enter c: 6
[[13, 14, 15, 16, 17, 18], [19, 20, 21, 22, 23, 24]]
ojasmittal@pop-os ~/D/Code> python3 q5.py
enter matrix: [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11,
12]]
enter r: 39
enter c: 2
Invalid Dimensions
```