Name:Samarth Motka

202311023

Programming Lab Assignment-1

1.Write a Python program from scratch to generate all permutations of a list in Python(without itertools).

```
x = [1,2,3]
ans = []
for i in x:
    for j in x:
        for k in x:
            if(i==j)or (j==k) or(k==i):
                pass
            else:
                ans.append((i,j,k))
print(ans)
```

```
[(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)]
```

2.Write a function that gives number of days of given year.

```
def ifLeap(year):
    if(year % 400 == 0 ) and (year % 100 == 0):
        return 366
    elif (year % 4 == 0) and (year % 100 != 0):
        return 366
    else :
        return 365

print(ifLeap(1990))
print(ifLeap(2044))
```

```
365
366
```

3.Print the name(s) of any car(s) having the second expensive car in. If there are multiple cars, order their names alphabetically and print each one on a new line.

```
cars = [['BMW',5000000],
['TOYOTA', 1000000],
```

```
['NISSAN',1500000],
['JAGUAR',8000000],
['MERCEDES',8000000],
['FERRARI',15000000]]



sorted_cars = sorted(cars, key=lambda x: x[1], reverse=True)
second_price = sorted_cars[1][1]
second_cars = [car[0] for car in sorted_cars if car[1] ==
second_price]

second_cars.sort()
for car_name in second_cars:
    print(car_name)
```

```
JAGUAR
MERCEDES
```

4.Count the frequency of each character in a string and store it in a dictionary.

```
a = 'adcbbdaacd'
count ={}
unique= set(a)
for i in unique:
    x = a.count(i)
    count[i]=x

x=list(count.keys())
x.sort()
sorted_count = {i:count[i] for i in x}
print(sorted_count)
```

```
{'a': 3, 'b': 2, 'c': 2, 'd': 3}
```

5.Output the space separated tuples of the cartesian product(without built-in function).

```
def cartesian(A,B):

        return [(i,j) for i in A for j in B ]

cartesian([1,2],[3,4])
```

```
[(1, 3), (1, 4), (2, 3), (2, 4)]
```

6.Write a Program which remove all the element with having count 2

```
a = [1,2,3,4,4,5,5,5,6,7,6,8,8,8,8]
count ={}
unique= set(a)
for i in unique:
    x = a.count(i)
    count[i]=x

print(count)
for x,y in count.items():
    if y == 2:
        [a.remove(x) for i in range(2) ]
a
```

```
{1: 1, 2: 1, 3: 1, 4: 2, 5: 3, 6: 2, 7: 1, 8: 4}
```

```
[1, 2, 3, 5, 5, 5, 7, 8, 8, 8, 8]
```

7.Write a program to find N largest and N smallest elements from a list. An example is given below.

```
arr = [13, 29, 26, 15, 4, 70, 89, 57, 34, 66, 10, 49]
arr.sort()
arr
n=int(input("Enter value of N "))

largest=[]
smallest=[]

for i in range(n):
    smallest.append(arr[i])

for i in range(len(arr)-1,len(arr)-n-1,-1):
    largest.append(arr[i])


print('Smallest : ',smallest)
print('Largest : ',largest)
```

```
Enter value of N 3
Smallest :  [4, 10, 13]
Largest :  [89, 70, 66]
```

8.Write a program to reverse a stack without using reverse() function. An example is given below.

```
stack =[3,7,2]
rev_stack=[]
def reverse(stack):
    i = len(stack)-1
    while(i>-1):

        rev_stack.append(stack[i])
        i -=1
    return rev_stack

print(reverse(stack))
```

```
[2, 7, 3]
```

9.Write a function to print N “Perfect” numbers. A perfect number is a positiveinteger that is equal to the sum of its proper positive divisors. I.e. 6 is a perfect number because 1+2+3 = 6

```

def isPerfect( n ):

    sum = 1
    i = 2
    while i * i <= n:
        if n % i == 0:
            sum = sum + i + n/i
        i += 1

    return (True if sum == n and n!=1 else False)

n = 4

for n in range (10000):
    if isPerfect (n):
        print(n)
```

```
6
28
496
8128
```

10.Write a function to check whether the given password is valid or not. Password is a combination of alphanumeric characters along with special characters. A password is correct if and only if all the following conditions are satisfied:

- Length of the password should be a minimum of 10.
- The alphabet must be between [a-z].
- At least one numeric character [0-9].
- At least one alphabet should be of Upper Case [A-Z].
- At least 1 special character from [_ or @ or $].
- No numeric character and special character should be adjacent.

```
import re


def check(password):
    if len(password) < 10:
        print("Invalid: Password length should be at least 10
characters.")
    elif not re.search('[a-z]', password):
        print("Invalid: Password must contain at least one lowercase
letter [a-z].")
    elif not re.search('[A-Z]', password):
        print("Invalid: Password must contain at least one uppercase
letter [A-Z].")
    elif not re.search('[0-9]', password):
        print("Invalid: Password must contain at least one numeric
character [0-9].")
    elif not re.search('[_@$]', password):
        print("Invalid: Password must contain at least one of the
special characters [_@$].")
    else:
        validity = False
        for i in range(len(password) - 1):
            if (password[i].isdigit() and (password[i+1] in '_@$')) or
(password[i] in '_@$' and password[i+1].isdigit()):
                validity = True
                break

    if validity:
        print("Invalid: Password should not have adjacent numeric or
special characters.")
    else:
        print("Valid password!")
```

```
password = 'D@aiict123'
password1 = 'Daiict_123'

check(password)
check(password1)
```

```
Valid password!
Invalid: Password should not have adjacent numeric or special
characters.
```