

```
In [5]: # WAP to count the occurrence of each character in your name.
st = input("Enter the word to check: ")
freq = []
for i in st:
    for c in freq:
        if c[0]==i:
            c[1]+=1
            break
    else:
        freq.append([i,1])

print("Letter frequencies:",freq)
```

Letter frequencies: [['r', 1], ['u', 1], ['p', 1], ['a', 1], ['k', 1]]

```
In [7]: #Q.2. WAP to implement Selection Sort.

def selection_sort(array):
    length = len(array)
    for i in range(length-1):
        minIndex = i
        for j in range(i+1, length):
            if array[j]<array[minIndex]:
                minIndex = j
        array[i], array[minIndex] = array[minIndex], array[i]
    return array
array = [72,4,10,38,2]
print("The sorted array is: ", selection_sort(array))
```

The sorted array is: [2, 4, 10, 38, 72]

```
In [9]: #Q.3. WAP to check whether a number is Krishnamurthy or not.

def factorial(n):
    fact=1
    for i in range(1,(n+1)):
        fact=fact*i
    return fact

if __name__ == "__main__":
    n=int(input("Enter the number to check: "))
    s=0
    temp=n
    while(n!=0):
        r=n%10
        s+=factorial(r)
        n=n//10
    if(s==temp):
        print("%d is a Krishnamurthy Number."%temp)
    else:
        print("%d is not a Krishnamurthy Number."%temp)
```

145 is a Krishnamurthy Number.

```
In [11]: #Q.4. WAP to check whether a given string is palindrome or not.

str1 = input("Enter a string: ")
```

```

str2 = str1.replace(" ", "").lower()

if str2 == str2[::-1]:
    print(f"{str1} is a Palindrome String.")
else:
    print(f"{str1} is not a Palindrome String.")

```

madam is a Palindrome String.

In [1]: *#Q.6. Write a python program to check whether a number is Armstrong or not.*

```

n=int(input("Enter the number to check: "))
temp=n
cnt=0
s=0
while(n!=0):
    cnt=cnt+1
    n=n//10
n=temp

while(n!=0):
    r=n%10
    s=s+r**cnt
    n=n//10

if(temp==s):
    print("%d is a Armstrong Number.">%temp)
else:
    print("%d is not a Armstrong Number.">%temp)

```

153 is a Armstrong Number.

In [17]: *#Q.7. WAP to find GCD of two numbers using recursion.*

```

def gcd(a,b):
    if b==0:
        return a
    return gcd(b,a%b)

x,y = [int(c) for c in input("Enter two numbers:").split()]

res=gcd(x,y)
print(f"The GCD of {x} and {y} is {res}.")

```

The GCD of 48 and 18 is 6.