

1. Write a program to print all Armstrong numbers in a given range.

**Program:**

```
x = 100 # Start Range
y = 500 # Stop Range
for num in range(x,y+1):
    length = len(str(num))
    sum=0
    temp=num
    while temp>0:
        sum = sum + ((temp % 10) ** length)
        temp //= 10
    if num == sum:
        print(num)
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
153
370
371
407
```

2. Program to Check Armstrong Number.

**Program:**

```
num = int(input("Enter a Number: "))

length = len(str(num))
sum = 0
temp = num

while(temp != 0):
    sum = sum + ((temp % 10) ** length)
    temp = temp // 10

if sum == num:
    print("Armstrong Number")
else:
    print("Not Armstrong Number")
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Enter a number: 371
Armstrong number
```

### 3. Write a program to Print Fibonacci sequence.

#### Program Using Loop:

```
num = int(input("Enter number of digits you want in series (minimum 2): "))

first = 0
second = 1

print("Fibonacci series is:")
print(first, ",", second, end=", ")

for i in range(2, num):
    next = first + second
    print(next, end=", ")

    first = second
    second = next
```

#### Program Using Recursion:

```
def recur_fibo(n):
    if n <= 1:
        return n
    else:
        return(recur_fibo(n-1) + recur_fibo(n-2))

num = int(input("Enter number of digits you want in series (minimum 2): "))
if num <= 0:
    print("Plese enter a positive integer")
else:
    print("Fibonacci series is:")
    for i in range(num):
        print(recur_fibo(i),end=',')
```

#### Output:

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Enter number of digits you want in series (minimum 2): 8
Fibonacci series is:
0 , 1, 1, 2, 3, 5, 8, 13,
```

### 4. Write a program to Check Prime Number.

#### Program:

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

for i in range(2, num//2):
    if num % i == 0:
        print("Not prime number")
        break
else:
    print(f"{num} Prime number")
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Enter a number: 67
67 Prime number
```

## 5. Write a program to Find Factorial of Number.

**Program Using Loop:**

```
num = int(input("Enter a number: "))
fac = 1
for i in range(1, num + 1):
    fac = fac * i
print("Factorial of ", num, " is ", fac)
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Enter a number: 6
Factorial of 6 is 720
```

**Program Using Recursion:**

```
def fact(n):
    if n == 0:
        return 1
    elif n == 1:
        return n
    else:
        return n*fact(n-1)

num = int(input("Enter a number: "))
if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
else:
    print("Factorial of",num,"is",fact(num))
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Enter a number: 6
Factorial of 6 is 720
```

## 6. Write a program to Check Leap Year.

**Program:**

```
year = int(input("Enter a year: "))
if(year%4==0 and (year%100!=0 or year%400==0)):
    print(year,"Leap year")
else:
    print(year,"Not leap year")
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Enter a year: 2020
2020 Leap year
```

## 7. Write a program to Check Palindrome Number.

### Program:

```
num = int(input("Enter a number: "))
temp , rev = num , 0
while temp != 0:
    rev = (rev * 10) + (temp % 10)
    temp //= 10
if num == rev:
    print(num,"Number is palindrome")
else:
    print(num,"Number is not palindrome")
```

### Output:

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Enter a number: 565
565 Number is palindrome
```

## 8. Write a program to multiply two numbers by repeated addition e.g. $5*6 = 5+5+5+5+5+5$

### Program Using Loop:

```
num1 = 5
num2 = 6
total = 0
for i in range(num2):
    total+=num1
print("Result",total)
```

### Program Using Recursion:

```
def rep_add(a,b):
    if b==0:
        return 0
    return a + rep_add(a,b-1)

print("Result",rep_add(5,6))
```

### Output:

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Result 30
```

9. Write a program to input two numbers as input and compute the greatest common divisor (GCD).

**Program Using Loop:**

```
n1 = 60
n2 = 48
loop=True
while loop:
    if n1 == n2:
        print(f"GCD: {n1}")
        loop=False
    elif n1>n2:
        n1 -= n2
    else:
        n2 -= n1
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
GCD: 12
```

**Program Using Recursion:**

```
def GCD(a,b):
    if(b==0):
        return a
    else:
        return GCD(b,a%b)
print (GCD(60,48))
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
12
```

10. Write a program to exchanging values of two variables.

**Program:**

```
x, y = 25, 36
temp = x
x = y
y = temp
print(f"After Swap x = {x}, y = {y}")
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
After Swap x = 36, y = 25
```

11. Write a program for converting a decimal number to binary equivalent.

**Program Using Recursion:**

```
def DecimalToBinary(num):  
    if num == 0:  
        return  
    else:  
        DecimalToBinary(num//2)  
        print(num % 2, end = '')
```

```
DecimalToBinary(8)
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop  
$ python kk.py  
1000
```

12. Write a program for converting a binary number to decimal equivalent.

**Program:**

```
def binaryToDecimal(binary):  
  
    binary1 = binary  
    decimal, i, = 0, 0  
    while binary != 0:  
        dec = binary % 10  
        decimal = decimal + dec * pow(2, i)  
        binary = binary//10  
        i += 1  
    print(binary1, '=', decimal)
```

```
binaryToDecimal(10101)
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop  
$ python kk.py  
10101 = 21
```

### 13. Write a program to Bubble Sort.

#### Program Using while Loop:

```
lis=[6,9,7,5,8,4]
print(f"Unsorted List: {lis}")
i=len(lis)-1
while i > 0 :
    j=0
    while j < i:
        if lis[j] > lis[j+1]:
            lis[j],lis[j+1] = lis[j+1],lis[j]
            print(lis)
        else:
            print(lis)
        j+=1
    print()
    i-=1
```

#### Program Using For Loop:

```
for i in range(len(lis)-1):
    for j in range(len(lis)-1-i):
        if lis[j] > lis[j+1]:
            lis[j],lis[j+1] = lis[j+1],lis[j]
            print(lis)
        else:
            print(lis)
    print()
print(f"Sorted List: {lis}")
```

#### Output:

UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop

\$ python kk.py

Unsorted List: [6, 9, 7, 5, 8, 4]

[6, 9, 7, 5, 8, 4]

[6, 7, 9, 5, 8, 4]

[6, 7, 5, 9, 8, 4]

[6, 7, 5, 8, 9, 4]

[6, 7, 5, 8, 4, 9]

[6, 7, 5, 8, 4, 9]

[6, 5, 7, 8, 4, 9]

[6, 5, 7, 8, 4, 9]

[6, 5, 7, 4, 8, 9]

[5, 6, 7, 4, 8, 9]

[5, 6, 7, 4, 8, 9]

[5, 6, 4, 7, 8, 9]

[5, 6, 4, 7, 8, 9]

[5, 4, 6, 7, 8, 9]

[4, 5, 6, 7, 8, 9]

Sorted List: [4, 5, 6, 7, 8, 9]

14. Write a program to print elements of upper triangular matrix.

**Program:**

```
mat=[[1,2,3],
     [4,5,6],
     [7,8,9]]
row=col=3
print('Elements of Upper triangular matrix')
if row==col:
    for i in range(row):
        for j in range(col):
            if j>=i:
                print(mat[i][j],end=" ")
else:
    print("not a square matrix")
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Elements of Upper triangular matrix
1 2 3 5 6 9
```

15. Write a program to print Pattern in Heart shape.

**Program:**

```
for row in range(6):
    for col in range(7):
        if (row==0 and col%3!=0) or (row==1 and col%3==0) or (row-col==2) or
(row+col==8):
            print("\U0001F49B ",end="")
        else:
            print(end=" ")
    print()
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
  ♥ ♥ ♥ ♥
♥   ♥   ♥
♥ ♥ ♥ ♥ ♥
  ♥ ♥ ♥
  ♥ ♥
  ♥
```

16. Write a program to print Triangle shape.

**Program:**

```
size = 4
for i in range(size):
    for j in range (i+1):
        if j==0 or j==i:
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
lowsize = size-1
for i in range(lowsize-1):
    for j in range(lowsize-i):
        if j==0 or j==lowsize-i-1:
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
print("*")
```

**Output:**

UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop

\$ python kk.py

```
*
* *
*  *
*  *
* *
*
```

17. Write a program to compute the wages of a daily laborer as per the following rules:

Hours Worked Rate Applicable upto first 8 hrs Rs100/- per hr

a) For next 4 hrs Rs30/- per hr extra

b) For next 4 hrs Rs40/- per hr extra

c) For next 4 hrs Rs50/- per hr extra

d) For rest Rs60/- per hr extra

**Program:**

```
employee_name = 'Suresh Kumar'
worked_hour = 15
if worked_hour <=8:
    wages = worked_hour*100
elif worked_hour >8 and worked_hour <=12:
    wages = (worked_hour-8)*30+worked_hour*100
elif worked_hour >12 and worked_hour <=16:
    wages = (worked_hour-8)*40+worked_hour*100
elif worked_hour >16 and worked_hour <=20:
    wages = (worked_hour-8)*50+worked_hour*100
else:
    wages = (worked_hour-8)*60+worked_hour*100
print("Total Wages: ",wages)
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Total Wages: 1780
```

18. Write a function that takes a string as parameter and returns a string with every successive repetitive character replaced by? e.g. school may become scho?l.

**Program:**

```
def fun1(st):
    ch=[]
    for i in st:
        ch.append(i)
    for i in range(len(ch)-1):
        if ch[i] == ch[i+1]:
            ch[i+1] = '?'
    return ''.join(ch)
```

```
st = 'schools schools'
print(fun1(st))
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
scho?ls scho?ls
```

19. Write a program that takes in a sentence as input and displays the number of words, number of capital letters, number of small letters, number of digit and number of special symbols.

**Program:**

```
snt='We1come To UPC1$$ Y0uTube Ch@nnel'  
if len(snt) == 0:  
    print("String is Empty")  
else:  
    word=1  
    capital=small=alnum=spe_ch=0  
    for i in snt:  
        if i.isspace():  
            word+=1  
        elif i.isupper():  
            capital+=1  
        elif i.islower():  
            small+=1  
        elif i.isalnum():  
            alnum+=1  
        else:  
            spe_ch+=1  
    else:  
        print(snt)  
        print(f"Number of Words: {word}")  
        print(f"Number of Upper: {capital}")  
        print(f"Number of Lower: {small}")  
        print(f"Number of Digit: {alnum}")  
        print(f"Number of Special Symbols: {spe_ch}")
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop  
$ python kk.py  
We1come To UPC1$$ Y0uTube Ch@nnel  
Number of Words: 5  
Number of Upper: 8  
Number of Lower: 15  
Number of Digit: 3  
Number of Special Symbols: 3
```

20. Write a Python function that takes two lists and returns true if they have at least one common item.

**Program:**

```
def common_data(list1, list2):  
    result = False  
    for x in list1:  
        for y in list2:  
            if x == y:  
                result = True  
                return result  
    return result
```

```
a = [1, 2, 3, 4, 5]  
b = [5, 6, 7, 8, 9]  
print(common_data(a, b))
```

```
a = [1, 2, 3, 4, 5]  
b = [6, 7, 8, 9]  
print(common_data(a, b))
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop  
$ python kk.py  
True  
False
```

21. Write a program which takes list of numbers as input and finds, the largest number in the list, the smallest number in the list and Product of all the items in the list.

**Program:**

```
li=[ int(l) for l in input("List: ").split(",")]
print("The list is ",li)
max=min=li[0]
pro=1
for x in li:
    if x>max:
        max=x
    if x<min:
        min=x
    pro*=x
print(f"Largest Number: {max}")
print(f"Smallest Number: {min}")
print(f"Product of All Item: {pro}")
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
List: 3,5,2,4,8,7
The list is [3, 5, 2, 4, 8, 7]
Largest Number: 8
Smallest Number: 2
Product of All Item: 6720
```

22. Write a program that takes sentence as input from the user and computes the frequency of each letter. Use a variable of dictionary type to maintain and show the frequency of each letter.

**Program:**

```
test_str = "welcome to upciss"
all_freq = {}
for i in test_str:
    if i in all_freq:
        all_freq[i] += 1
    else:
        all_freq[i] = 1
print ("Count of all characters in GeeksforGeeks is :\n " + str(all_freq))
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ python kk.py
Count of all characters in GeeksforGeeks is :
{'w': 1, 'e': 2, 'l': 1, 'c': 2, 'o': 2, 'm': 1, ' ': 2, 't': 1, 'u': 1, 'p': 1, 'i': 1, 's': 2}
```

23. Write a function that takes two filenames f1 and f2 as input. The function should read the contents of f1 line by line and write them onto f2.

**Program:**

```
def filefunc(f1,f2):
    openf1=open(f1,'r')
    openf2=open(f2,'w+')
    print("File 1 Content:\n",openf1.read())
    openf1.seek(0)
    openf2.write(openf1.read())
    openf2.seek(0)
    print("File 2 Content:\n",openf2.read())
    openf1.close()
    openf2.close()
```

```
f1=input("Enter a Frist File Name: ")
f2=input("Enter a Second File Name: ")
filefunc(f1,f2)
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
$ py demo.py
Enter a Frist File Name: f1.txt
Enter a Second File Name: f2.txt
File 1 Content:
Hello Students
Welcome To UPCISS
Thank for Like and Subscribe
File 2 Content:
Hello Students
Welcome To UPCISS
Thank for Like and Subscribe
```

24. Write a function that reads the contents of the file f3.txt and counts the number of alphabets, blank spaces, lowercase letters, and number of words starting with a vowel.

**Program:**

```
def filefunc(f3):
    file3=open(f3,'r')
    snt=file3.read()
    file3.close()
    if len(snt) == 0:
        print("String is Empty")
    else:
        alpha=space=lower=vowel=0
        for i in snt:
            if i.isalpha():
                alpha+=1
            elif i.isspace():
                space+=1
            if i.islower():
                lower+=1
        vow='aeiou'
        li=snt.split(' ')
        for word in li:
            for x in vow:
                if word.startswith(x):
                    vowel+=1
        print(f"File Content:\n{snt}\n")
        print(f"Number of Alphabets: {alpha}")
        print(f"Number of Blank Space: {space}")
        print(f"Number of Lowercase: {lower}")
        print(f"Starting with a vowel: {vowel}")
```

```
filefunc("f3.txt")
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop
```

```
$ py demo.py
```

```
File Content:
```

```
Hello Students!
```

```
Welcome TO UPCISS
```

```
Thank for Like and Subscribe
```

```
Number of Alphabets: 51
```

```
Number of Blank Space: 9
```

```
Number of Lowercase: 38
```

```
Starting with a vowel: 1
```

25. Write a program to replace 'a' with 'b', 'b' with 'c'.....'z' with 'a' and similarly for 'A' with 'B', 'B' with 'C'..... 'Z' with 'A' in a file. The other characters should remain unchanged.

**Program:**

```
s1='UPci$$ c0mpuTer'  
s2=[]  
for i in s1:  
    if i.isalpha():  
        s2.append(chr(ord(i)+1))  
    else:  
        s2.append(i)  
s2=''.join(s2)  
print(s2)
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop  
$ py demo.py  
VQdj$$ d0nqvUfs
```

26. Write a NumPy program to find the most frequent value in an array.

**Program:**

```
import numpy as np  
x = np.array([5,9,6,9,6,2,6])  
print("Original array:")  
print(x)  
print("Most frequent value in the above array:")  
print(np.bincount(x).argmax())
```

**Output:**

```
UPCISS@DESKTOP-32007C6 MINGW64 ~/desktop  
$ py demo.py  
Original array:  
[5 9 6 9 6 2 6]  
Most frequent value in the above array:  
6
```

27. Take two NumPy arrays having two dimensions. Concatenate the arrays on axis 1.

**Program:**

```
import numpy as np  
arr1 = np.arange(1,10).reshape(3,3)  
arr2 = np.arange(10,19).reshape(3,3)  
print(arr1)  
print()  
print(arr2)  
print()  
print(np.concatenate((arr1,arr2),axis=1))
```