Practical 2

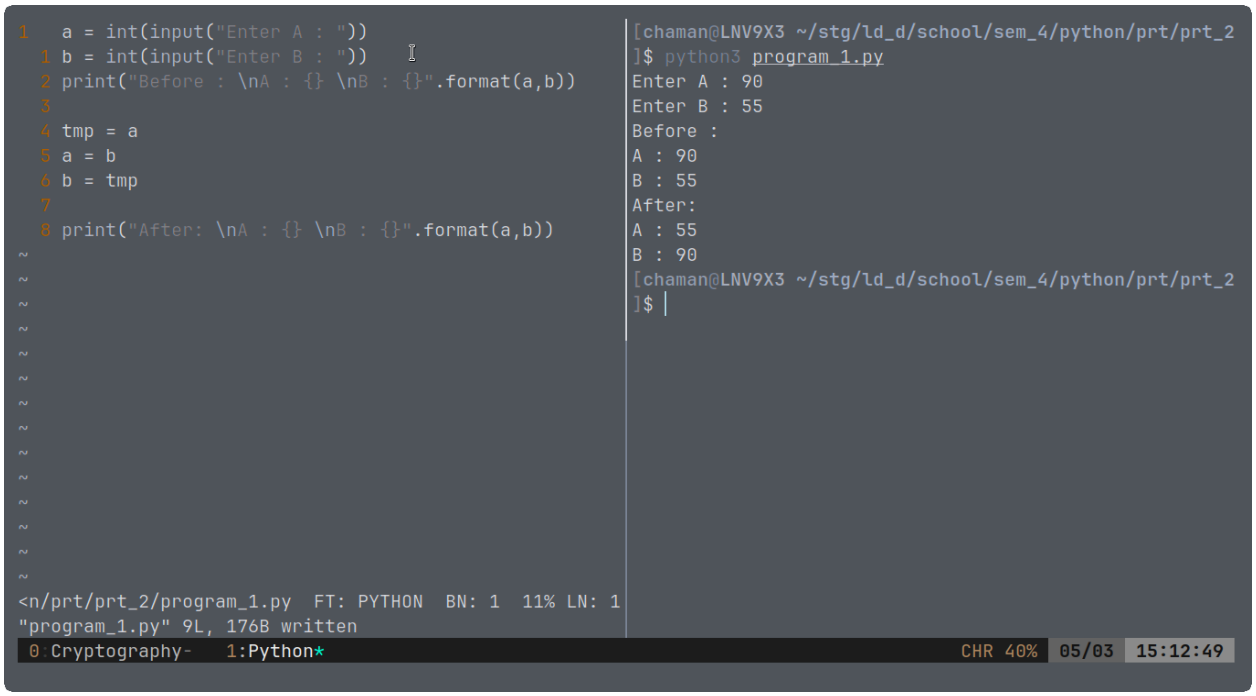
Q : Python Program to Swap Two Variables
Sol :

```
a = int(input("Enter A : "))
b = int(input("Enter B : "))
print("Before : \nA : {} \nB : {}".format(a,b))

tmp = a
a = b
b = tmp

print("After: \nA : {} \nB : {}".format(a,b))
```

Output :



Q : Write a Python function that finds all the permutations of the members of a list.
Sol :

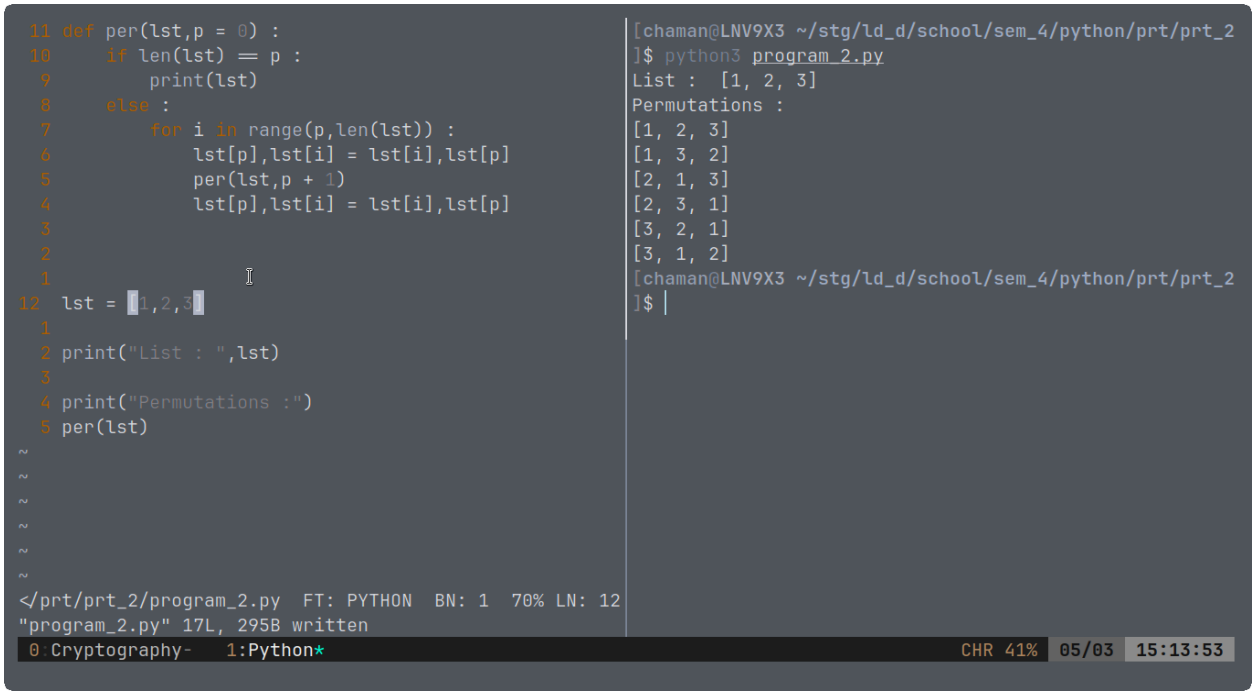
```
def per(lst,p = 0) :
    if len(lst) == p :
        print(lst)
    else :
        for i in range(p,len(lst)) :
            lst[p],lst[i] = lst[i],lst[p]
            per(lst,p + 1)
            lst[p],lst[i] = lst[i],lst[p]

lst = [1,2,3]
```

```
print("List : ",lst)

print("Permutations :")
per(lst)
```

Output :



Q : Write a Python function to find the union and intersection of two lists.
Sol :

```
def union(lst_1,lst_2) :
    lst_3 = []
    lst_3 = set(lst_1+lst_2)
    lst_3 = list(lst_3)
    return lst_3

def diff(lst_1,lst_2) :
    lst_3 = []
    lst_3 = list(set(list(set(lst_1)-set(lst_2)) + list(set(lst_2)-set(lst_1))))
    return lst_3

_list_ = [1,2,3,4,5,8,8,0,1,6,7]
_list2_ = [1,2,3,4,5,6,7,8,0,4,1,6,9]

print(_list_)
print(_list2_)

print('Union : {}'.format(union(_list_,_list2_)))
print('Differance : {}'.format(diff(_list_,_list2_)))
```

```
def union(lst_1,lst_2) :
    lst_3 = []
    lst_3 = set(lst_1+lst_2)
    lst_3 = list(lst_3)
    return lst_3

def diff(lst_1,lst_2) :
    lst_3 = []
    lst_3 = list(set(list(set(lst_1)-set(lst_2)) +
    return lst_3

_list_ = [1,2,3,4,5,8,8,0,1,6,7]
_list2_ = [1,2,3,4,5,6,7,8,0,4,1,6,9]

print(_list_)
print(_list2_)

print('Union : {}'.format(union(_list_,_list2_)))
print('Difference : {}'.format(diff(_list_,_list2_)
~
~
~
</prt/prt_2/program_3.py  FT: PYTHON  BN: 1  60% LN: 12
10 fewer lines
 0 Cryptography-   1:Python*                                NCH 89%  05/03  18:28:14
```

```
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$ python3 program_3.py
[1, 2, 3, 4, 5, 8, 8, 0, 1, 6, 7]
[1, 2, 3, 4, 5, 6, 7, 8, 0, 4, 1, 6, 9]
Union : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Difference : [9]
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$
```

Q : Python program to generate the prime numbers from 1 to N.

Sol :

```
n = int(input("Enter : "))

for num in range(2,n) :
    p=True
    for i in range(2,num) :
        if (num%i==0) :
            p=False
    if p :
        print(num)
```

```
n = int(input("Enter : "))

for num in range(2,n) :
    p=True
    for i in range(2,num) :
        if (num%i==0) :
            p=False
    if p :
        print(num)
~
~
~
~
~
~
~
~
~
~
~
~
~
<n/prt/prt_2/program_4.py  FT: PYTHON  BN: 1  11% LN: 1
"program_4.py" 9L, 165B
 0 Cryptography-   1:Python*                                CHR 43%  05/03  15:15:49
```

```
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$ python3 program_4.py
Enter : 10
2
3
5
7
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$ python3 program_4.py
Enter : 15
2
3
5
7
11
13
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$
```

Q : Python program to find the factorial of a number using recursion.
Sol :

```
def fact(n) :
    if n == 1 :
        return 1
    else :
        return n*fact(n-1)
a = int(input("Enter : "))

print("Factorial of {} : {}".format(a,fact(a)))
```

Output :