Enrollment No. : 210031101611003
Name : Chaman Deshmukh
Course : BTech in CSE (Specialization in Cyber Security) - Sem 3
Subject : Python Programming Language (Practicals)

Practical 1

Q : Explore Different Data Types in Python
Sol :

```python
_int_ = 3
_str_ = "Chaman Deshmukh"
_list_ = ['Chaman','Deshmukh']
_tuple_ = ('Chaman',' Deshmukh')
_set_ = {"Chaman",'Deshmukh'}
_dic_ = { 'name':'Chaman Deshmukh', 'course':'B.Tech(III)',
'subject':'Python'}

print('Integer :',_int_,'Type :',type(_int_))
print('String :',_str_,'Type :',type(_str_))
print('List :',_list_,'Type :',type(_list_))
print('Tuple :',_tuple_,'Type :',type(_tuple_))
print('Set :',_set_,'Type :',type(_set_))
print('Dict :',_dic_,'Type :',type(_dic_))
```

Output :

Practical 2

Q : Python Program to Swap Two Variables
Sol :

```python
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 :

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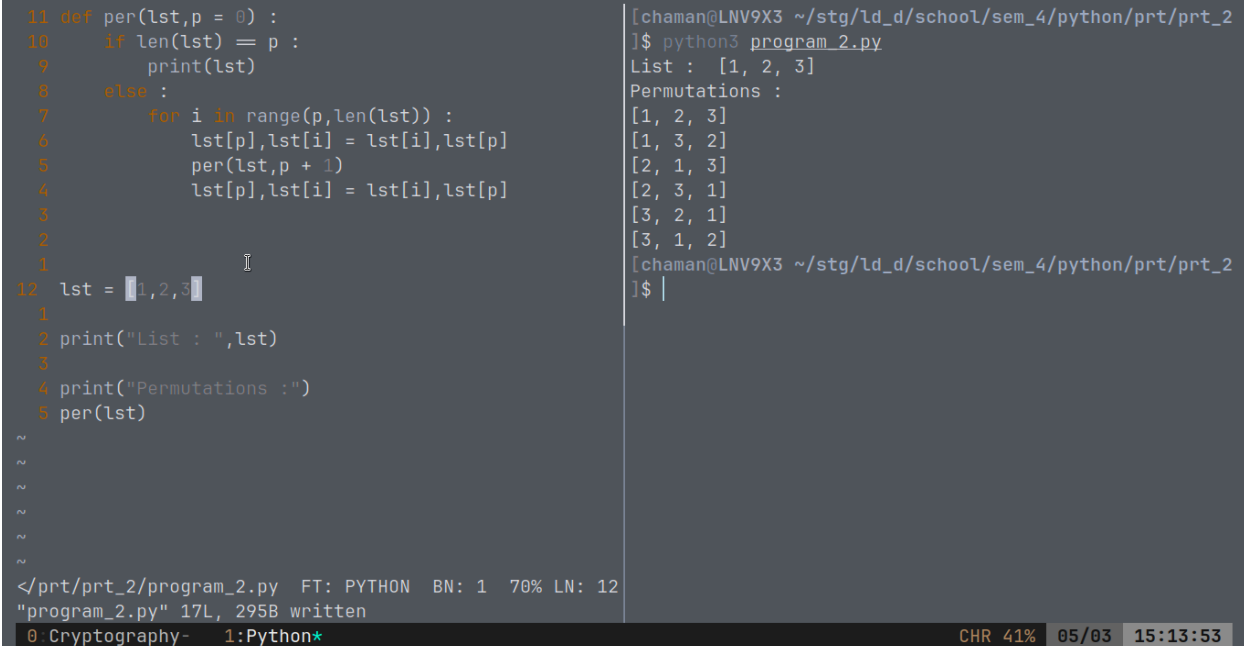
```python
lst = [1,2,3]

print("List : ",lst)

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

Output :

```
11 def per(lst,p = 0) :                        [chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
10      if len(lst) == p :                       ]$ python3 program_2.py
 9          print(lst)                           List :  [1, 2, 3]
 8      else :                                   Permutations :
 7          for i in range(p,len(lst)) :         [1, 2, 3]
 6              lst[p],lst[i] = lst[i],lst[p]     [1, 3, 2]
 5              per(lst,p + 1)                    [2, 1, 3]
 4              lst[p],lst[i] = lst[i],lst[p]     [2, 3, 1]
 3                                                [3, 2, 1]
 2                                                [3, 1, 2]
 1                                                [chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
12  lst = [1,2,3]                                ]$
 1
 2 print("List : ",lst)
 3
 4 print("Permutations :")
 5 per(lst)
~
~
~
~
~
~
</prt/prt_2/program_2.py  FT: PYTHON  BN: 1  70% LN: 12
"program_2.py" 17L, 295B written
0 Cryptography-   1:Python*                                          CHR 41%  05/03  15:13:53
```

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

```python
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_)
```

```python
print(_list2_)

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

```
11 def union(lst_1,lst_2) :
10     lst_3 = []
 9     lst_3 = set(lst_1+lst_2)
 8     lst_3 = list(lst_3)
 7     return lst_3
 6
 5 def diff(lst_1,lst_2) :
 4     lst_3 = []
 3     lst_3 = list(set(list(set(lst_1)-set(lst_2)) +
 2     return lst_3
 1
12 |
 1 _list_ = [1,2,3,4,5,8,8,0,1,6,7]
 2 _list2_ = [1,2,3,4,5,6,7,8,0,4,1,6,9]
 3
 4 print(_list_)
 5 print(_list2_)
 6
 7 print('Union : {}'.format(union(_list_,_list2_)))
 8 print('Differance : {}'.format(diff(_list_,_list2_))

~
~
~
</prt/prt_2/program_3.py  FT: PYTHON  BN: 1  60% LN: 12
10 fewer lines
 0:Cryptography-    1:Python*
```

```
[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]
Differance : [9]
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$
```

```
NCH 89%  05/03  18:28:14
```

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

```python
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)
```

```
1   n = int(input("Enter : "))
1
2 for num in range(2,n) :
3     p=True
4     for i in range(2,num) :
5         if (num%i==0) :
6             p=False
7     if p :
8         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 :

```python
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 :

```python
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)))
```

```
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$ python3 program_5.py
Enter : 10
Factorial of 10 : 3628800
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$ python3 program_5.py
Enter : 5
Factorial of 5 : 120
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$ python3 program_5.py
Enter : 8
Factorial of 8 : 40320
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_2
]$
```

<n/prt/prt_2/program_5.py FT: PYTHON   BN: 1 12% LN: 1
"program_5.py" 8L, 161B

0:Cryptography-   1:Python*   CHR 44% 05/03 15:17:05

Practical 3

Q : Write a program for Merge Sort.

Sol :

```python
import random
def __merge_sort__(_list_):
    if len(_list_) > 1 :

        left_list = _list_[:len(_list_)//2]
        right_list = _list_[len(_list_)//2:]

        __merge_sort__(left_list)
        __merge_sort__(right_list)

        i = j = k = 0

        while i < len(left_list) and j < len(right_list):
            if left_list[i] <  right_list[j] :
                _list_[k] = left_list[i]
                i+=1
            else :
                _list_[k] = right_list[j]
                j+=1
            k+=1
        while i < len(left_list) :
            _list_[k] = left_list[i]
            i+=1
            k+=1
        while j < len(right_list) :
            _list_[k] = right_list[j]
            j+=1
            k+=1

size = int(input("Enter Size : "))
_list_ = random.sample(range(100),size)
print("Before :",_list_)
__merge_sort__(_list_)
print("After : {}".format(_list_))
```

Output :

```
program_1.py  program_2.py  program_3.py
1    import random
1 def __merge_sort__(_list_):
2     if len(_list_) > 1 :
3
4         left_list = _list_[:len(_list_)//2]
5         right_list = _list_[len(_list_)//2:]
6
7         __merge_sort__(left_list)
8         __merge_sort__(right_list)
9
10        i = j = k = 0
11
12        while i < len(left_list) and j < len(right_list):
13            if left_list[i] <  right_list[j] :
14                _list_[k] = left_list[i]
15                i+=1
16            else :
17                _list_[k] = right_list[j]
18                j+=1
19            k+=1
20        while i < len(left_list) :
21            _list_[k] = left_list[i]
22            i+=1
23            k+=1
24        while j < len(right_list) :
25            _list_[k] = right_list[j]
26            j+=1
27            k+=1
28
29 size = int(input("Enter Size : "))
30 _list_ = random.sample(range(100),size)
31 print("Before :",_list_)
32 __merge_sort__(_list_)
33 print("After : {}".format(_list_))
```

```
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_3]$ python3 program_2.py
Enter Size : 10
Before : [80, 35, 14, 55, 17, 2, 9, 43, 45, 87]
After : [2, 9, 14, 17, 35, 43, 45, 55, 80, 87]
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_3]$
```

```
<_d/school/sem_4/python/prt/prt_3/program_2.py  FT: PYTHON  BN: 2  2% LN: 1:1
34 lines yanked into "+
0:Cryptography   1:Python*   2:vim-                                    CHR 96% 20/03  22:19:43
```

Q : Write a program for Bubble Sort.
Sol :

```python
import random

def __bubble_sort__(lst):
    for i in range(len(lst)):
        for j in range(len(lst)):
            if lst[i]<lst[j]:
                temp = lst[j]
                lst[j] = lst[i]
                lst[i] = temp
    return lst

_list_ = random.sample(range(100),10)
print("Before :",_list_)
print("After : ",__bubble_sort__(_list_))
```

Output :

Q : Write a program for Quick Sort.
Sol :

```python
import random

def __quick_sort__(lst, _left_, _right_):
    if _left_<_right_ :
        block = __partition__(lst, _left_, _right_)
        __quick_sort__(lst, _left_, block - 1 )
        __quick_sort__(lst, block + 1, _right_)

def __partition__(lst, left, right):
    i = left
    j = right - 1
    pivot = lst[right]

    while i < j :
        while i < right and lst[i] < pivot :
            i += 1
        while j > left and lst[j] >= pivot :
            j -= 1
        if i < j:
            lst[i], lst[j] = lst[j], lst[i]
    if lst[i] > pivot:
        lst[i], lst[right] = lst[right], lst[i]

    return i
```

```python
size = int(input("Enter Size : "))
_list_ = random.sample(range(100),size)
print("Before :",_list_)
__quick_sort__(_list_, 0, len(_list_)-1)
print("After : {}".format(_list_))
```

Output :


```python
3 import random
2
1 def __quick_sort__(lst, _left_,_right_):
4     if _left_<_right_ :
1         block = __partition__(lst, _left_, _right_)
2         __quick_sort__(lst, _left_, block - 1 )
3         __quick_sort__(lst, block + 1, _right_)
4
5 def __partition__(lst, left, right):
6     i = left
7     j = right - 1
8     pivot = lst[right]
9
10    while i < j :
11        while i < right and lst[i] < pivot :
12            i += 1
13        while j > left and lst[j] ≥ pivot :
14            j -= 1
15        if i < j:
16            lst[i], lst[j] = lst[j], lst[i]
17    if lst[i] > pivot:
18        lst[i], lst[right] = lst[right], lst[i]
19
20    return i
21
22
23 size = int(input("Enter Size : "))
24 _list_ = random.sample(range(100),size)
25 print("Before :",_list_)
26 __quick_sort__(_list_, 0, len(_list_)-1)
27 print("After : {}".format(_list_))
<d/school/sem_4/python/prt/prt_3/program_3.py  FT: PYTHON  BN: 3  12% LN: 4:5

0 Cryptography   1:Python*   2:vim-                                CHR 98% 20/03 22:25:13
```

```
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_3]$ python3 program_3.py
Enter Size : 10
Before : [96, 55, 6, 89, 54, 10, 20, 44, 28, 2]
After : [2, 6, 10, 20, 28, 44, 54, 55, 89, 96]
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_3]$ python3 program_3.py
Enter Size : 18
Before : [14, 60, 29, 79, 55, 40, 11, 35, 84, 20, 71, 8, 67, 92, 97, 17, 44, 34]
After : [8, 11, 14, 17, 20, 29, 34, 35, 40, 44, 55, 60, 67, 71, 79, 84, 92, 97]
[chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_3]$ 
```

Practical 4

Q : Write a Python program to list only directories, files and all directories, files in a specified path.
Sol :

```python
import os

_path_ = input("Enter Path : ")

if _path_ :
    for folder in os.scandir(_path_) :
        if folder.is_dir() :
            print(folder.path)
else :
    _path_ = os.getcwd()
    print("Path : ",_path_)
    for folder in os.scandir(_path_) :
        if folder.is_dir() :
            print(folder.path)
```

Output :

Q : Write a Python program to scan a specified directory and identify the subdirectories and files.
Sol :
import os

```python
_path_ = input("Enter Path : ")

if _path_ :
    for curr,folder,file in os.walk(_path_) :
        print("Current Dir : ",curr)
        print("Folders : ",folder)
        print("Files : ",file)
else :
    _path_ = os.getcwd()
    print("Path : ",_path_)
    for curr,folder,file in os.walk(_path_) :
        print("Current Dir : ",curr)
        print("Folders : ",folder)
        print("Files : ",file)
```

Output :

Q : Write a Python program to get the size, permissions, owner, device, created, last modified and last accessed date and time of a specified path.

Sol :

```python
import os
from datetime import datetime

_path_ = input("Enter Path : ")

if _path_ :
    print("Mode : ",os.stat(_path_).st_mode)
    print("Owner : ",os.stat(_path_).st_dev)
```

```python
        print("Created : ",datetime.fromtimestamp(os.stat(_path_).st_ctime))
        print("Last Modified : ",datetime.fromtimestamp(os.stat(_path_).st_mtime))
        print("Last Accessed : ",datetime.fromtimestamp(os.stat(_path_).st_atime))
else :
        _path_ = os.getcwd()
        print("Path : ",_path_)
        print("Mode : ",os.stat(_path_).st_mode)
        print("Owner : ",os.stat(_path_).st_dev)
        print("Created : ",datetime.fromtimestamp(os.stat(_path_).st_ctime))
        print("Last Modified : ",datetime.fromtimestamp(os.stat(_path_).st_mtime))
        print("Last Accessed : ",datetime.fromtimestamp(os.stat(_path_).st_atime))
```

Output :

Q : Write a Python program to access environment variables and the value of the environment variable.

Sol :

```python
import os

print("Environment Variables : ",os.environ)
print("\n\nValue of Env Variable for HOME : ",os.getenv('HOME'))
```

Output :

```
  program_1.py   program_2.py   program_3.py   program_4.py        X  [chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_4]$ python program
  2 import os                                                       _4.py
  1                                                                 Environment Variables :  environ({'BROWSER': 'w3m', 'DBUS_SESSION_BUS_AD
  3   print("Environment Variables : ",os.environ)                 DRESS': 'unix:path=/run/user/1000/bus', 'DEBUGINFOD_URLS': 'https://debu
  1   print("\n\nValue of Env Variable for HOME : ",os.getenv(      ginfod.archlinux.org ', 'DISPLAY': ':0', 'EDITOR': 'vim', 'FZF_DEFAULT_O
  2                                                                 PTS': '--layout=reverse', 'HOME': '/home/chaman', 'INVOCATION_ID': 'c423
  ~                                                                 d6e3abcf48c3afe8445a793119d9', 'KEYTIMEOUT': '1', 'LANG': 'en_IN.UTF-8',
  ~                                                                  'LC_COLLATE': 'C', 'LESSHISTFILE': '/home/chaman/.cache/less/history',
  ~                                                                 'LOGNAME': 'chaman', 'MAIL': '/var/spool/mail/chaman', 'MOTD_SHOWN': 'pa
  ~                                                                 m', 'OLDPWD': '/home/chaman/stg/ld_d/school/sem_4/python/prt/prt_4', 'PA
  ~                                                                 TH': '/home/chaman/.cargo/bin:/usr/local/sbin:/usr/local/bin:/usr/bin:/u
  ~                                                                 sr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/core_perl:/home/chaman/.l
  ~                                                                 ocal/bin', 'PWD': '/home/chaman/stg/ld_d/school/sem_4/python/prt/prt_4',
  ~                                                                  'SHELL': '/usr/bin/zsh', 'SHLVL': '2', 'SYSTEMD_EXEC_PID': '295', 'TERM
  ~                                                                 ': 'tmux-256color', 'TERMINAL': 'st', 'TERM_PROGRAM': 'tmux', 'TERM_PROG
  ~                                                                 RAM_VERSION': '3.3a', 'TMUX': '/tmp/tmux-1000/default,618952,3', 'TMUX_P
  ~                                                                 ANE': '%13', 'USER': 'chaman', 'WATCH_INTERVAL': '0.1', 'WINDOWID': '293
  ~                                                                 60134', 'WINDOWPATH': '1', 'XAUTHORITY': '/tmp/Xauthority', 'XDG_CACHE_D
  ~                                                                 IR': '/home/chaman/.cache', 'XDG_CONFIG_DIR': '/home/chaman/.config', 'X
  ~                                                                 DG_DOCUMENTS_DIR': '/home/chaman/docs', 'XDG_DOWNLOAD_DIR': '/home/chama
  ~                                                                 n/dwla', 'XDG_HOME_DIR': '/home/chaman', 'XDG_RUNTIME_DIR': '/run/user/1
  ~                                                                 000', 'XDG_SEAT': 'seat0', 'XDG_SESSION_CLASS': 'user', 'XDG_SESSION_ID'
  ~                                                                 : '1', 'XDG_SESSION_TYPE': 'tty', 'XDG_VTNR': '1', 'ZDOTDIR': '/home/cha
  ~                                                                 man/.config/zsh', '_': '/usr/bin/python'})
  ~
  ~
  ~                                                                 Value of Env Variable for HOME :  /home/chaman
  ~                                                                 [chaman@LNV9X3 ~/stg/ld_d/school/sem_4/python/prt/prt_4]$
<hon/prt/prt_4/program_4.py  FT: PYTHON  BN: 4  60% LN: 3:14
"program_4.py" 5L, 122B [w]
 0:Cryptography-    1:Python*   2:LSS   3:DAA   4:Temp                                              BAT 73%  30/03  05:05:08
```