Assignment-8(RID:001,Madhur Jodhwani)

1.Design python application which creates two thread named as even and odd. Even

thread will display first 10 even numbers and odd thread will display first 10 odd numbers.

Code:

```
import threading

def Even(value):
    print("Inside Even function")
    for i in range(1,2*value):
        if i%2==0:
            print(i)

def Odd(value):
    print("Inside Odd function")
    for i in range(1,2*value):
        if i%2!=0:
            print(i)

def main():
    print("Inside Main")
    print("Enter Number")
    no=int(input())
    t1=threading.Thread(target=Even,args=(no,))
    t2=threading.Thread(target=Odd,args=(no,))
    t1.start()
    t2.start()


if __name__=="__main__":
    main()
```

```
Inside Main
Enter Number
10
Inside Even function
2
4
6
8
10
12
14
16
18
Inside Odd function
1
3
5
7
9
11
13
15
17
19
PS C:\Users\INTEL\Desktop\Python files>
```

2.Design python application which creates two threads as evenfactor and oddfactor.

Both the thread accept one parameter as integer. Evenfactor thread will display addition of even factors of given number and oddfactor will display addition of odd

factors of given number. After execution of both the thread gets completed main thread should display message as “exit from main”.

Code:

```
import threading

def EvenFactors(value):
    sum_even=0
    print("Inside Even function")
    for i in range(1,(value//2)+1):
        if i%2==0:
```

```
            sum_even+=i
    print(sum_even)

def OddFactors(value):
    sum_odd=0
    print("Inside Odd function")
    for i in range(1,(value//2)+1):
        if i%2!=0:
            sum_odd+=i
    print(sum_odd)

def main():
    print("Inside Main")
    print("Enter Number")
    no=int(input())
    t1=threading.Thread(target=EvenFactors,args=(no,))
    t2=threading.Thread(target=OddFactors,args=(no,))
    t1.start()
    t2.start()
    t1.join()
    t2.join()
    print("exit from main")


if __name__=="__main__":
    main()
```

Ans:

```
PS C:\Users\INTEL\Desktop\Python files> & 'C:\Users\INTEL\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\INTEL\.vscode\extensions\ms-python.python-2021.2.6369
28669\pythonFiles\lib\python\debugpy\launcher' '61623' '--' 'c:\Users\INTEL\Desktop\Python files\Assignment-8\EvenOddFactors.py'
Inside Main
Enter Number
6
Inside Even function
2
Inside Odd function
4
exit from main
PS C:\Users\INTEL\Desktop\Python files>
```

3.Design python application which creates two threads as evenlist and oddlist. Both the

threads accept list of integers as parameter. Evenlist thread add all even elements from input list and display the addition. Oddlist thread add all odd elements from input

list and display the addition.

Code:

```
import threading

def EvenFactors(value):
    sum_even=0
    print("Inside Even function")
    for i in range(0,len(value)):
        if value[i]%2==0:
            sum_even+=value[i]
    print(sum_even)

def OddFactors(value):
    sum_odd=0
    print("Inside Odd function")
    for i in range(0,len(value)):
        if value[i]%2!=0:
            sum_odd+=value[i]
    print(sum_odd)

def main():
    print("Inside Main")
    print("Enter Number of elements")
    no=int(input())
    arr=[]
    for i in range (0,no):
        arr.append(int(input()))
    t1=threading.Thread(target=EvenFactors,args=(arr,))
    t2=threading.Thread(target=OddFactors,args=(arr,))
    t1.start()
    t2.start()
```

```
if __name__=="__main__":
    main()
```

Ans:

```
PS C:\Users\INTEL\Desktop\Python files>  & 'C:\Users\INTEL\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\INTEL\.vscode\extensions\ms-python.python-2021.2.6369
28669\pythonFiles\lib\python\debugpy\launcher' '61727' '--' 'c:\Users\INTEL\Desktop\Python files\Assignment-8\EvenOddList.py'
Inside Main
Enter Number of elements
6
12
23
23
12
45
89
Inside Even function
24
Inside Odd function
180
PS C:\Users\INTEL\Desktop\Python files>
```

4.Design python application which creates three threads as small, capital, digits. All the

threads accepts string as parameter. Small thread display number of small characters,

capital thread display number of capital characters and digits thread display number of

digits. Display id and name of each thread.

Code:

```
import os
import threading

def Small(value):
    print("PID of Small function thread is: ",os.getpid())
    count_small=0
    for i in range(0,len(value)):
        if ((value[i]>='a')and(value[i]<='z')):
            count_small+=1
    print(count_small)

def Capital(value):
```

```
    print("PID of Capital function thread is: ",os.getpid())
    count_capital=0
    for i in range(0,len(value)):
        if((value[i]>='A')and(value[i]<='Z')):
            count_capital+=1
    print(count_capital)


def Digits(value):
    print("PID of Digit function thread is: ",os.getpid())
    count_digits=0
    for i in range(0,len(value)):
        if((value[i]>='0') and (value[i]<='9')):
            count_digits+=1
    print(count_digits)

def main():
    print("Inside Main")
    print("PID of main function: ",os.getpid())
    print("PPID of main function: ",os.getppid())
    print("enter string")
    str=input()
    t1=threading.Thread(target=Small,args=(str,))
    t2=threading.Thread(target=Capital,args=(str,))
    t3=threading.Thread(target=Digits,args=(str,))
    t1.start()
    t2.start()
    t3.start()


if __name__=="__main__":
    main()
```

Ans:

```
PS C:\Users\INTEL\Desktop\Python files>  & 'C:\Users\INTEL\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\INTEL\.vscode\extensions\ms-python.python-2021.2.636928669\pythonFiles\lib\python\debugpy\launcher' '61921' '--' 'c:\Users\INTEL\Desktop\Python files\Assignment-8\3Thread.py'
Inside Main
PID of main function:  10400
PPID of main function:  9472
enter string
qweQWE123
PID of Small function thread is:  10400
3
PID of Digit function thread is:  10400
3
PID of Capital function thread is:  10400
3
PS C:\Users\INTEL\Desktop\Python files>
```

5.Design python application which contains two threads named as thread1 and thread2.

Thread1 display 1 to 50 on screen and thread2 display 50 to 1 in reverse order on screen. After execution of thread1 gets completed then schedule thread2.

Code:

```python
import threading

def printOrder(value,lock):
    lock.acquire()
    print("Inside PrintOrderly Thread")
    for i in range(1,value+1):
        print("Thread PrintOrderly: ",i)
    lock.release()

def printReverse(value,lock):
    lock.acquire()
    print("Inside Print Reverse Thread")
    while(value!=0):
        print("Thread Printreverse: ",value)
        value-=1
    lock.release()

def main():
    print("Enter Number")
    no=int(input())
    lock=threading.Lock()
    t1=threading.Thread(target=printOrder,args=(no,lock,))
    t2=threading.Thread(target=printReverse,args=(no,lock,))
```

```
    t1.start()
    t2.start()
    print("End of Main")

if __name__=="__main__":
    main()
```

Ans:

```
PS C:\Users\INTEL\Desktop\Python files>  & 'C:\Users\INTEL\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\INTEL\.vscode\extensions\ms-python.python-2021.2.636928669\pythonFiles\lib\python\debugpy\launcher' '62017' '--' 'c:\Users\INTEL\Desktop\Python files\Assignment-8\Print50.py'
Enter Number
10
Inside PrintOrderly Thread
End of Main
Thread PrintOrderly:  1
Thread PrintOrderly:  2
Thread PrintOrderly:  3
Thread PrintOrderly:  4
Thread PrintOrderly:  5
Thread PrintOrderly:  6
Thread PrintOrderly:  7
Thread PrintOrderly:  8
Thread PrintOrderly:  9
Thread PrintOrderly:  10
Inside Print Reverse Thread
Thread Printreverse:  10
Thread Printreverse:  9
Thread Printreverse:  8
Thread Printreverse:  7
Thread Printreverse:  6
Thread Printreverse:  5
Thread Printreverse:  4
Thread Printreverse:  3
Thread Printreverse:  2
Thread Printreverse:  1
PS C:\Users\INTEL\Desktop\Python files>
```