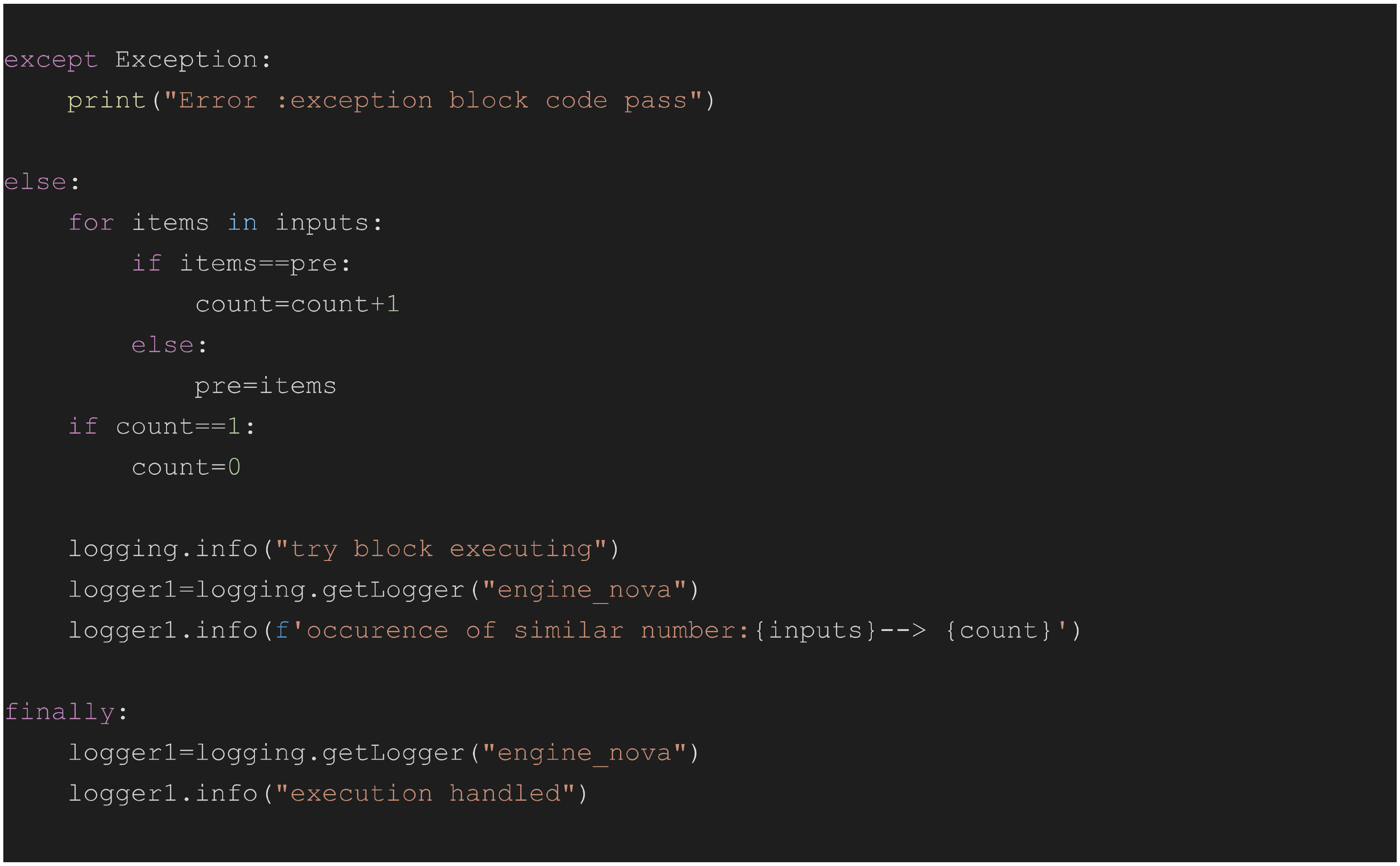
PYTHON BASIC

Assignment_2

1. Create a function that takes three integer arguments (a, b, c) and returns the amount of
   integers which are of equal value.
   Examples
   equal(3, 4, 3) ➞ 2
   equal(1, 1, 1) ➞ 3
   equal(3, 4, 1) ➞ 0
   Notes
   Your function must return 0, 2 or 3.

Sol:

```
import logging
pre=1+1j
count=1
inputs=[]

try:
    logging.basicConfig(filename=
'/content/logfile12.log',level=logging.DEBUG,format='%(name)s - %(asctime)s -
%(message)s - %(levelname)s')

#Create Handler :
    console_log=logging.StreamHandler()
    console_log.setLevel(logging.DEBUG)

# log formatting function:
    format=logging.Formatter('%(name)s , %(asctime)s , %(message)s ,%(levelname)s')
    console_log.setFormatter(format)
    logging.getLogger('').addHandler(console_log)

    ##  loop for input variable
    for i in range(3):
        a=int(input("number"))
        inputs.append(a)

    inputs.sort()
```



```
except Exception:
    print("Error :exception block code pass")

else:
    for items in inputs:
        if items==pre:
            count=count+1
        else:
            pre=items
    if count==1:
        count=0

    logging.info("try block executing")
    logger1=logging.getLogger("engine_nova")
    logger1.info(f'occurence of similar number:{inputs}--> {count}')

finally:
    logger1=logging.getLogger("engine_nova")
    logger1.info("execution handled")
```

Output:

```
number3
number4
number3
root , 2022-06-11 17:01:34,461 , try block executing ,INFO
engine_nova , 2022-06-11 17:01:34,469 , occurrence of similar number:[3, 3, 4]--> 2 ,INFO
engine_nova , 2022-06-11 17:01:34,478 , execution handled ,INFO
```

Q2: Write a function that converts a dictionary into a list of keys-values tuples.
Note:Return the elements in the list in alphabetical order.

```
d={"D":2,"B":3,"C":1}

di=list(d.items())

di.sort()

di
```

OUTPUT:

```
>>    [('B', 3), ('C', 1), ('D', 2)]
```

```
x={'likes':2,'dislikes': 3,'followers': 10}

xi=list(x.items())

xi.sort()

xi
```

OUTPUT:

```
      >>  [('dislikes', 3), ('followers', 10), ('likes', 2)]
```

3.Write a function that creates a dictionary with each (key, value) pair being the (lower case,upper case) versions of a letter, respectively.

Sol:

```
def func(*args):

    mylist= []

    for i in args:

        mylist.append(i)

    dictionary=dict((i.lower(), i.upper()) for i in mylist)

    return dictionary

func('a','B',"c","a","F")
```

OUTPUT:

```
{'a': 'A', 'b': 'B', 'c': 'C', 'f': 'F'}
```

Q4. Write a function that replaces all vowels in a string with a specified vowel.

```
try:
    ip=input("Enter the string: ").lower()
    vl=["a","e","i","o","u"]
    vob=input("Type vowel to replace strign vowel: ").lower()
    iplist=[]
    if vob in vl:
        if vob.isalpha():
            for z in range(len(ip)):
                if ip[z] in vl:
                    rep=ip[z].replace(ip[z],vob)
                    iplist.append(rep)
                else:
                    iplist.append(ip[z])
    else:
        print('input is incorrect! enter vowels only')

except Exception:
    print('executed except: Error')

finally:
    outpt=""
    for i in iplist:
        outpt=outpt+i
    print(outpt)
```

OUTPUT:

```
Enter the string: apple and banana
Type vowel to replace strign vowel: i
ippli ind binini
```

5.Create a function that takes a string as input and capitalizes a letter if its ASCII code is even and returns its lowercase version if its ASCII code is odd.

Sol:

```
try:
    cap=[]
    inp=input("enter string: ")
    for i in inp:
        if ord(i)%2==0:
            cap.append(i.upper())
        else:
            cap.append(i.lower())

except Exception:
    print('executed except: Error')

finally:
    outpt=""
    for i in cap:
        outpt=outpt+i
    print(outpt)
```

```
>>
      enter string: Oh what a beautiful morning.
>>     oH wHaT a BeauTiFuL moRNiNg.
```