Question 1: Python Basics?

A-If you have two lists, L1=['HTTP','HTTPS','FTP','DNS'] L2=[80,443,21,53], convert it to generate this dictionary d={'HTTP':80,'HTTPS':443,'FTP':21,'DNS':53}

```
d=\{\}
L1 = ['HTTP' L'HTTPS' L'FTP' L'DNS']
L2=[80,443,21,53]
for i j in zip(L1 L2):
    d[i]=j
print(d)
```

```
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
Process finished with exit code 0
```

B- Write a Python program that calculates the factorial of a given number entered by user.

```
x = int(input('enter the number '))
mul = 1
for i in range (x,1,-1):
    mul = mul * i

print(mul)
```

```
enter the number 5
120
```

```
C-L=['Network', 'Bio', 'Programming', 'Physics', 'Music']
```

In this exercise, you will implement a Python program that reads the items of the previous list and identifies the items that starts with 'B' letter, then print it on screen.

Tips: using loop, 'Len ()', starts with() method

```
L=['Network','Bio', 'Programming', 'Physics', 'Music']
i=0

while i< len(L):
    if L[i].startswith('B'):
        print(L[i])

i=i+1
```

```
Bio
Process finished with exit code 0
```

D- Using Dictionary comprehension, Generate this dictionary

 $d = \{0:1,1:2,2:3,3:4,4:5,5:6,6:7,7:8,8:9,9:10,10:11\}$

```
d={i:i+1 for i in range(0,11)}
print(d)
```

```
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}

Process finished with exit code 0
```

Question 2: Convert from Binary to Decimal

Write a Python program that converts a Binary number into its equivalent Decimal number.

The program should start reading the binary number from the user. Then the decimal equivalent number must be calculated. Finally, the program must display the equivalent decimal number on the screen.

Tips: solve input errors.

```
n = int(input('n = '))
i = res = 0
while n! = 0:
    res = res + (n%10) * (2**i)
    n = n / / 10
i + = 1
print("n = " res)
```

```
n= 1100
n= 12

Process finished with exit code 0
```

Question 3: Working with Files" Quiz Program"

Type python quiz program that takes a text or json or csv file as input for (20 (Questions, Answers)). It asks the questions and finally computes and prints user results and store user name and result in separate file csv or json file.

```
import json
que = { }
#define a variable for the score
res = 0
num=1
b = open("questions.txt",'r')
que = json.load(b)
b.close()
name = input("Enter your name: ")
#display the questions
for i in que.keys():
    print("Question", num , ": ", i)
    a = input("The answer is ")
    if a == que[i]:
      res = res + 1
    num = num + 1
result={name:res}
m = open("result.txt",'w')
result = json.dump(result,m)
m.close()
```

```
🗾 questions - Notepad
File Edit Format View Help
" 'abc'.islower() return true " : "t",
" 'abc'.isupper() return true " : "f"
" 'abc'.isdigit() return true " : "f"
" 'abc'.isalnum() return true " : "t"
" 'ab12'.isalpha() return true " : "f"
" 'ab12'.isdigit() return true " : "f"
" 'ab12'.isalnum() return true " : "t",
" 'abc'.startswith('a') return true " : "t",
" 'abc'.startswith('ab') return true " : "t",
" 'abc'.startswith('b') return true " : "f",
" 'abc'.startswith('c') return true " : "f",
" 'abc'.endswith('c') return true " : "t",
" 'abc'.endswith('bc') return true " : "t",
" 'abc'.endswith('a') return true " : "f",
" '_' mean it is private " : "t",
" there are three types of errors in python " : "t",
" 'ABC'.islower() return true " : "f",
" 'ABC'.isupper() return true " : "t",
" '123'.isdigit() return true " : "t",
" '12n'.isalnum() return true " : "t",
```

result - Notepad

File Edit Format View Help {"Reem Alali": 11}

Question 4: Object-Oriented Programming - Bank Class

Define a class BankAccount with the following attributes and methods: Attributes: account_number (string), account_holder (string), balance (float, initialized to 0.0) Methods:deposit(amount), withdraw(amount), get_balance() - Create an instance of BankAccount, - Perform a deposit of \$1000, - Perform a withdrawal of \$500. - Print the current balance after each operation. - Define a subclass SavingsAccount that inherits from BankAccount and adds interest_rate Attribute and apply_interest() method that Applies interest to the balance based on the interest rate. And Override print() method to print the current balance and rate. - Create an instance of SavingsAccount, and call apply_interest() and print() functions

```
class BankAccount:

def __init__(self_account_number__account_holder_):
    self.account_number = account_number
    self.account_holder = account_holder

self.balance = 0.0

2 usages

def deposit(self_amount):
    self.balance += amount

1 usage

def withdraw(self_amount):
    if self.balance >= amount

2 usages

def get_balance(self):
    return self.balance
```

```
def __init__(self_account_number, account_holder, interest_rate):
        super().__init__(account_number, account_holder)
        self.interest_rate = interest_rate
   def apply_interest(self):
        interest_amount = self.balance *self.interest_rate
        self.balance += interest_amount
   def print(self):
        print(f"Current balance: ${self.balance}, Interest rate: {self.interest_rate} ")
bank_acc=BankAccount('963852741','Reem Alali' )
bank_acc.deposit(1000)
print(f'Balance after deposit: $ {bank_acc.get_balance()}')
bank_acc.withdraw(500)
print(f'Balance after withdraw: $ {bank_acc.get_balance()}')
savings_acc=SavingsAccount('123456789', 'Nabel ', 0.05)
savings_acc.deposit(2000)
savings_acc.apply_interest()
savings_acc.print()
```

```
Balance after deposit: $ 1000.0

Balance after withdraw: $ 500.0

Current balance: $2100.0, Interest rate: 0.05

Process finished with exit code 0
```