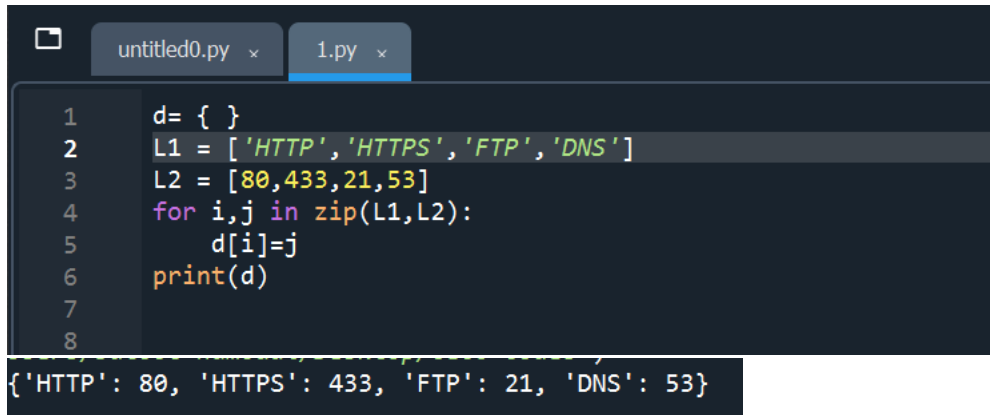


**Name:** Nour Khasrouf

**Number:**2649

**Question 1:**

**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 }



```
1 d= { }
2 L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
3 L2 = [80,443,21,53]
4 for i,j in zip(L1,L2):
5     d[i]=j
6 print(d)
7
8
```

{'HTTP': 80, 'HTTPS': 433, 'FTP': 21, 'DNS': 53}

.....

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

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

```
enter your number10  
3628800
```

.....

**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  
for i in range(len(L)):  
    if L[i].startswith("B"):  
        print(L[i])
```

Bio

.....

**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= {a:a+1 for a 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}
```

.....

### 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.

```
b = list(input("Input a binary number: "))
x = 0

for i in range(len(b)):
    d = b.pop()
    if d == '1':
        x = x + pow(2, i)
print("The decimal ", x)
```

```
Input a binary number: 1001
The decimal value of the number is 9
```

.....

### 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
ques = { }
result = 0

num=1

n = open("questions.txt",'r')
ques = json.load(n)
n.close()

name = input("Enter your name: ")

for i in ques:

    print("Question",num , ": ", i)
    answer = input("The answer is ")

    if answer == ques[i]:
        result = result + 1

    num = num + 1

final={name:result}
a = open("result.txt",'w')
final = json.dump(final,a)
a.close()
```



result - Notepad

File Edit Format View Help

```
{ "Nour Kh": 9 }
```

.....

#### 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

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

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

    def get_balance(self):
        return self.balance

class SavingsAccount(BankAccount):
    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):
```

```
def print(self):  
    print(f'current balance: ${self.balance}, interest rate: {self.interest_rate}')  
  
bank_acc=BankAccount('278952361','Nour kh')  
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('224466888', 'Ali', 0.05)  
savings_acc.deposit(2000)  
savings_acc.apply_interest()  
savings_acc.print()
```

.....