Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No1

الجمهورية العربية السورية

اللاذقية -جامعـة تشريـــن

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة1 برمجة شبكات

Name:_____, Number:_____, Submitted To GitHub:_____

First Network Programming Homework

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 **}**

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

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 ()' , startswith() methods.

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}

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.

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.

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.

Notes "! important"

- Homework is accepted as **well explained** Pdf & "Nicely Formatted Code" "You can do all job in one notebook then print as pdf or "copy and paste" on word document "use" then convert into pdf with extra info "

-You have to show:

Question number >>Question itself>> your answer code with explanations > your Result "you can use this doc as template"

-You Have to Show code execution as Screenshots from your laptop or phone''.

-Apply your full name and number, Homework number to pdf.

-Similar Solutions will rejected and not accepted.

- The Homework is accepted until the date of "27/5/2024", if after >> mark=mark- (current_date -27/5/2024)*0.3

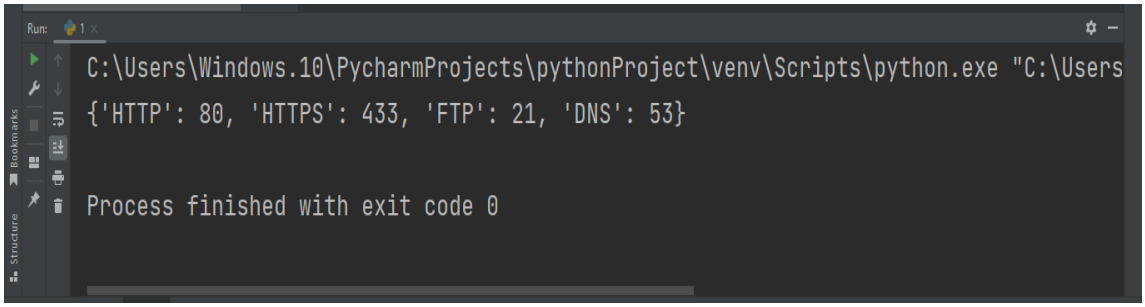
- upload your code to your GitHub Account, "PDF + Code"

My Name is: Batol Bassam Dayyub Number: 1799

Question1: python Basic?

A-

```python
d= {}
L1 = ['HTTP','HTTPS','FTP','DNS']
L2 = [80,433,21,53]
for i,j in zip(L1,L2):
    d[i]=j
print(d)
```



B-

```python
def factorial (n):
    if n== 0:
        return 1
    else:
        return n * factorial(n-1)
    num = int(input("Enter a number"))
    if num < 0:
        print("factorial is'nt defined for negative numbers")
    elif num==0:
        print("the factorial is 1")
    else:
        result = factorial(num)
        print (f"the factorial of {num} is {result}")
```

C-

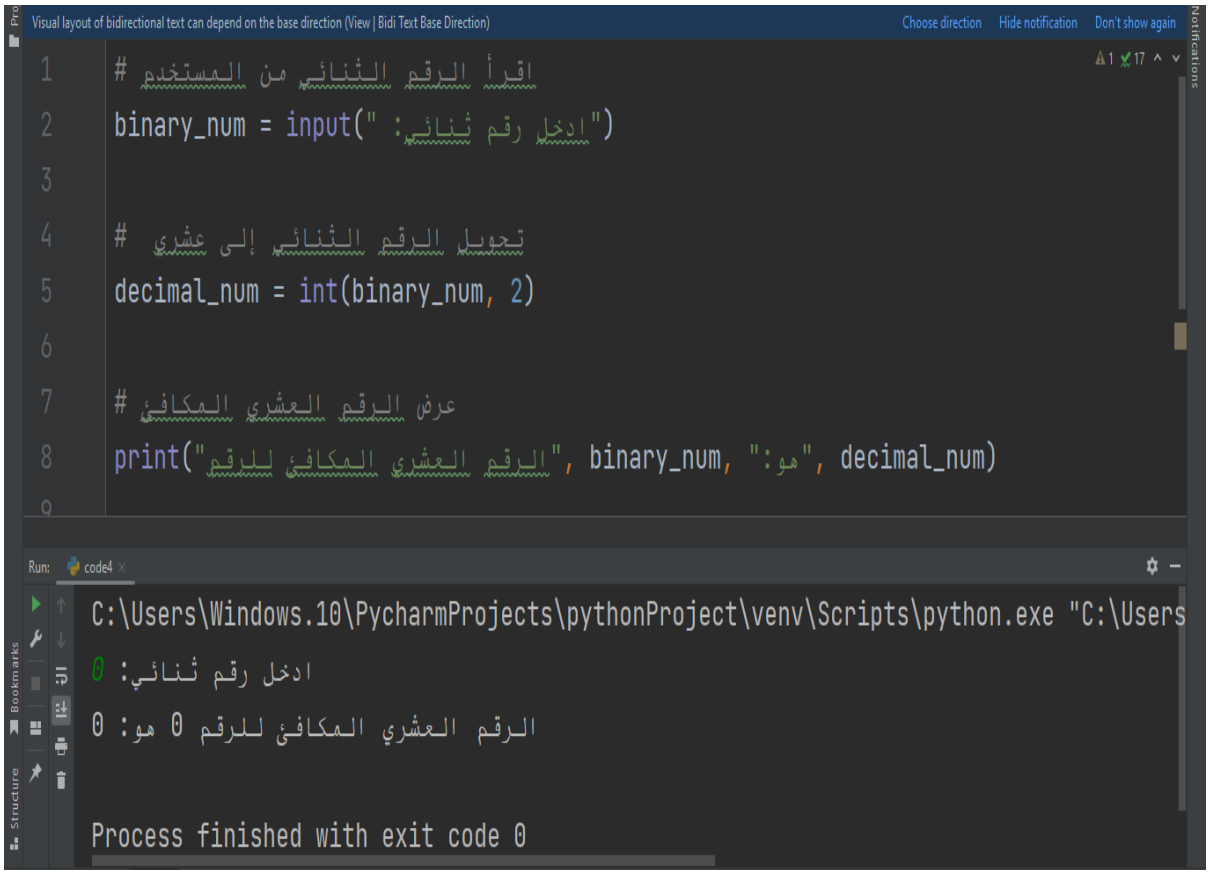
```python
L= ['Network', 'Bio','Programming','Physics','Music']
i = 0
for i in range(len(L)):
    if L[i].startswith("B"):
        print(L[i])
```

D-

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

```
C:\Users\Windows.10\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users
{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
```

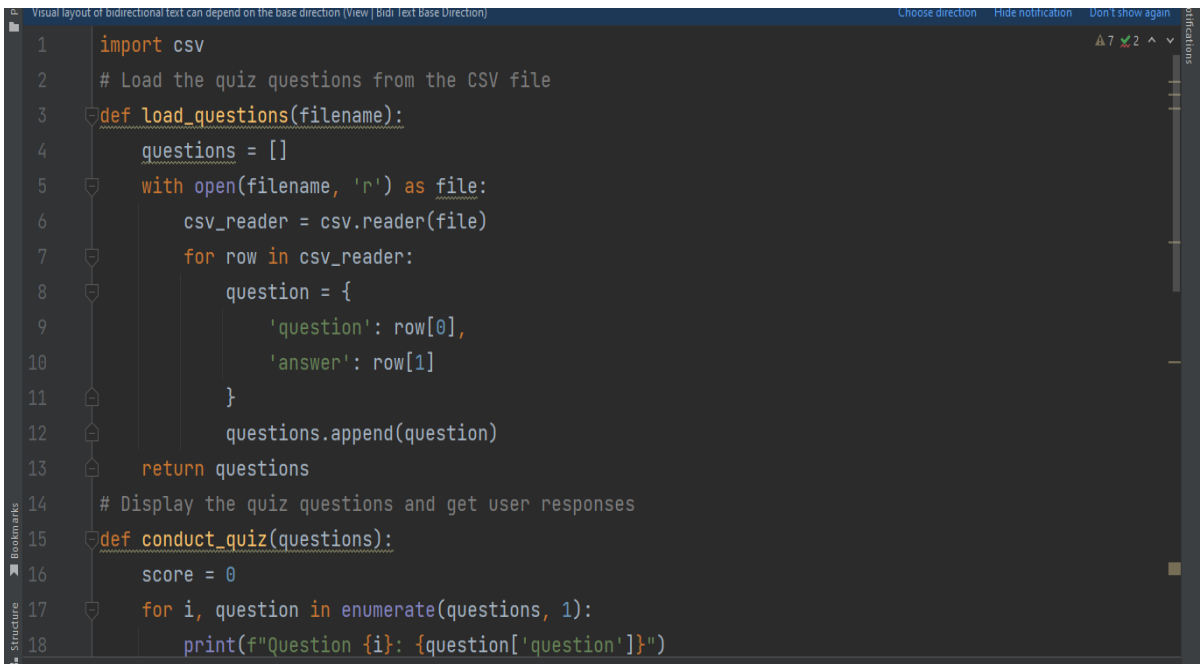
Question2: Convert from Binary to Decimal



```
# اقرأ الرقم الثنائي من المستخدم
binary_num = input(" :ادخل رقم ثنائي")

#  تحويل الرقم الثنائي إلى عشري
decimal_num = int(binary_num, 2)

# عرض الرقم العشري المكافئ
print("الرقم العشري المكافئ للرقم", binary_num, ":هو", decimal_num)
```

```
C:\Users\Windows.10\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users
0 :ادخل رقم ثنائي
الرقم العشري المكافئ للرقم 0 هو: 0

Process finished with exit code 0
```

Question3: Working with files "Quiz Program"

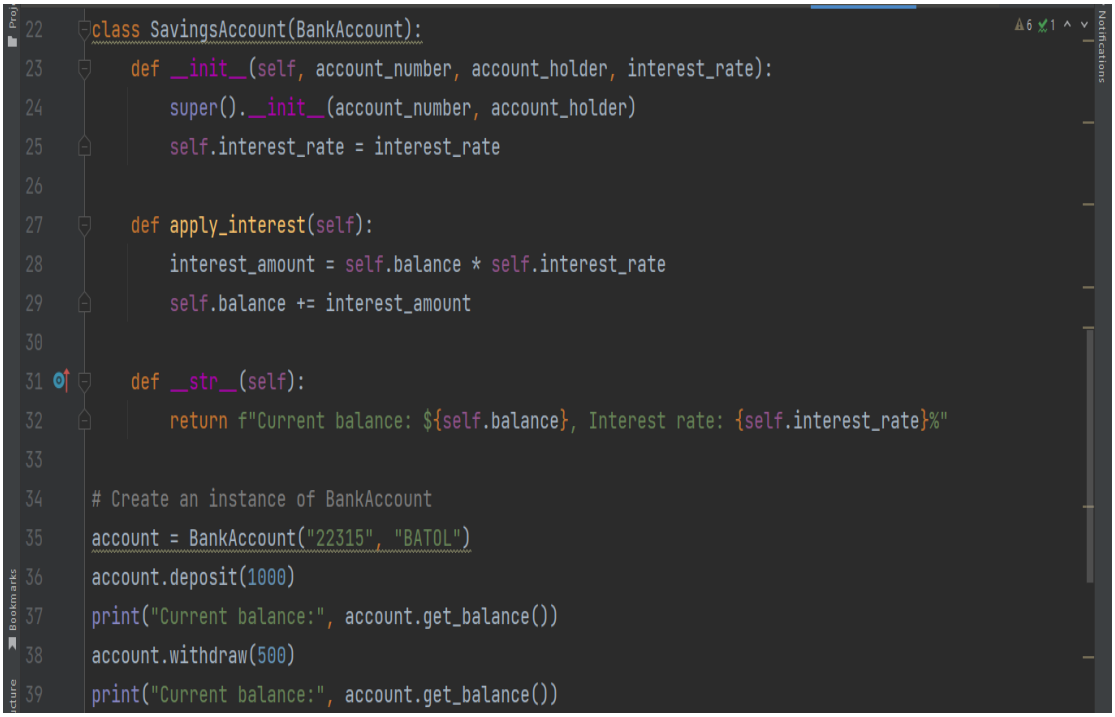


```
import csv
# Load the quiz questions from the CSV file
def load_questions(filename):
    questions = []
    with open(filename, 'r') as file:
        csv_reader = csv.reader(file)
        for row in csv_reader:
            question = {
                'question': row[0],
                'answer': row[1]
            }
            questions.append(question)
    return questions
# Display the quiz questions and get user responses
def conduct_quiz(questions):
    score = 0
    for i, question in enumerate(questions, 1):
        print(f"Question {i}: {question['question']}")
```

```
        print(f"Question {i}: {question['question']}")
        user_answer = input("Your answer: ")
        if user_answer.lower() == question['answer'].lower():
            score += 1
    return score
# Get user name
user_name = input("Enter your name: ")
# Load questions from the CSV file
questions = load_questions('quiz_questions.csv')
# Conduct the quiz
user_score = conduct_quiz(questions)
# Print the user's score
print(f"Dear {user_name}, your score is: {user_score}")
# Store user name and score in a CSV file
with open('user_results.csv', 'a', newline='') as file:
    csv_writer = csv.writer(file)
    csv_writer.writerow([user_name, user_score])
```

Question4 : Object Oriented Programming – Bank class

```
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):
        if amount > 0:
            self.balance += amount
            print(f"Deposited ${amount}. New balance is ${self.balance}")
        else:
            print("Invalid deposit amount.")
    def withdraw(self, amount):
        if 0 < amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew ${amount}. New balance is ${self.balance}")
        else:
            print("Invalid withdrawal amount or insufficient balance.")
    def get_balance(self):
        return self.balance
```



```python
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 __str__(self):
        return f"Current balance: ${self.balance}, Interest rate: {self.interest_rate}%"

# Create an instance of BankAccount
account = BankAccount("22315", "BATOL")
account.deposit(1000)
print("Current balance:", account.get_balance())
account.withdraw(500)
print("Current balance:", account.get_balance())
```

```python
# Create an instance of BankAccount
account = BankAccount("22315", "BATOL")
account.deposit(1000)
print("Current balance:", account.get_balance())
account.withdraw(500)
print("Current balance:", account.get_balance())

# Create an instance of SavingsAccount
savings_account = SavingsAccount("7758", "NOUR", 0.05)
savings_account.deposit(2000)
print("Current balance before interest:", savings_account.get_balance())
savings_account.apply_interest()
print(savings_account)
```