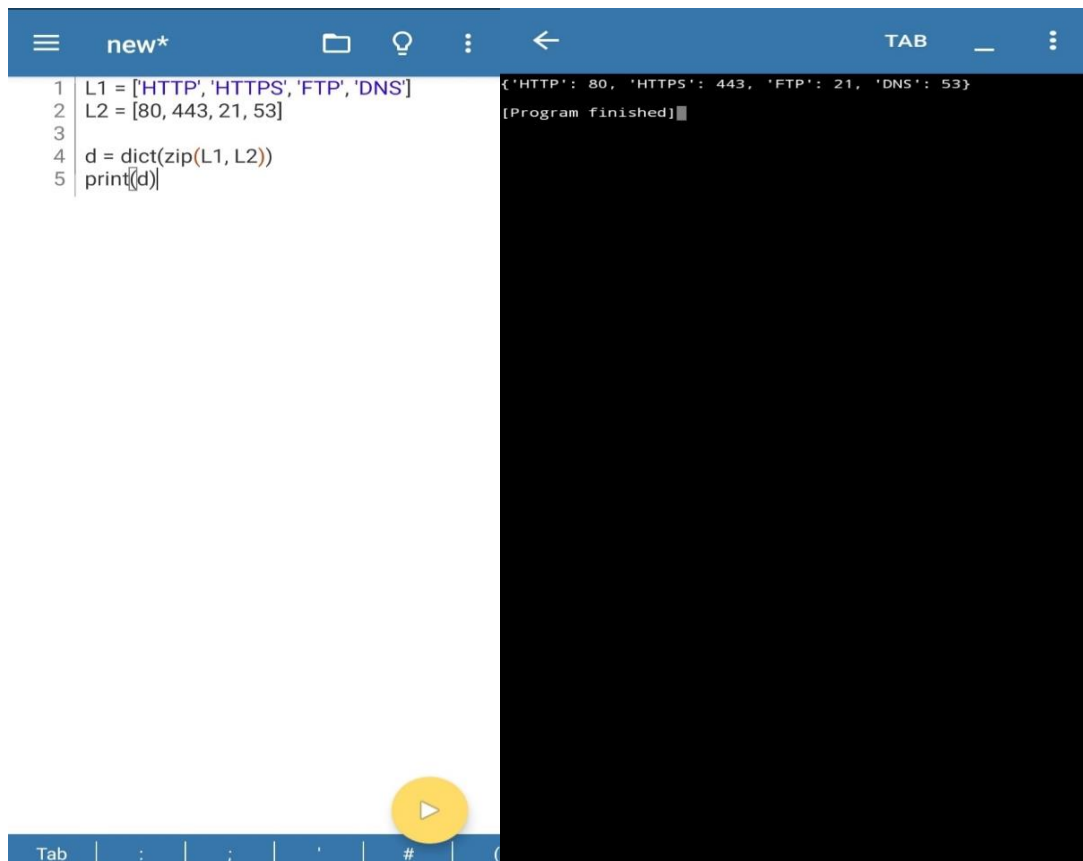


Question 1:

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} .

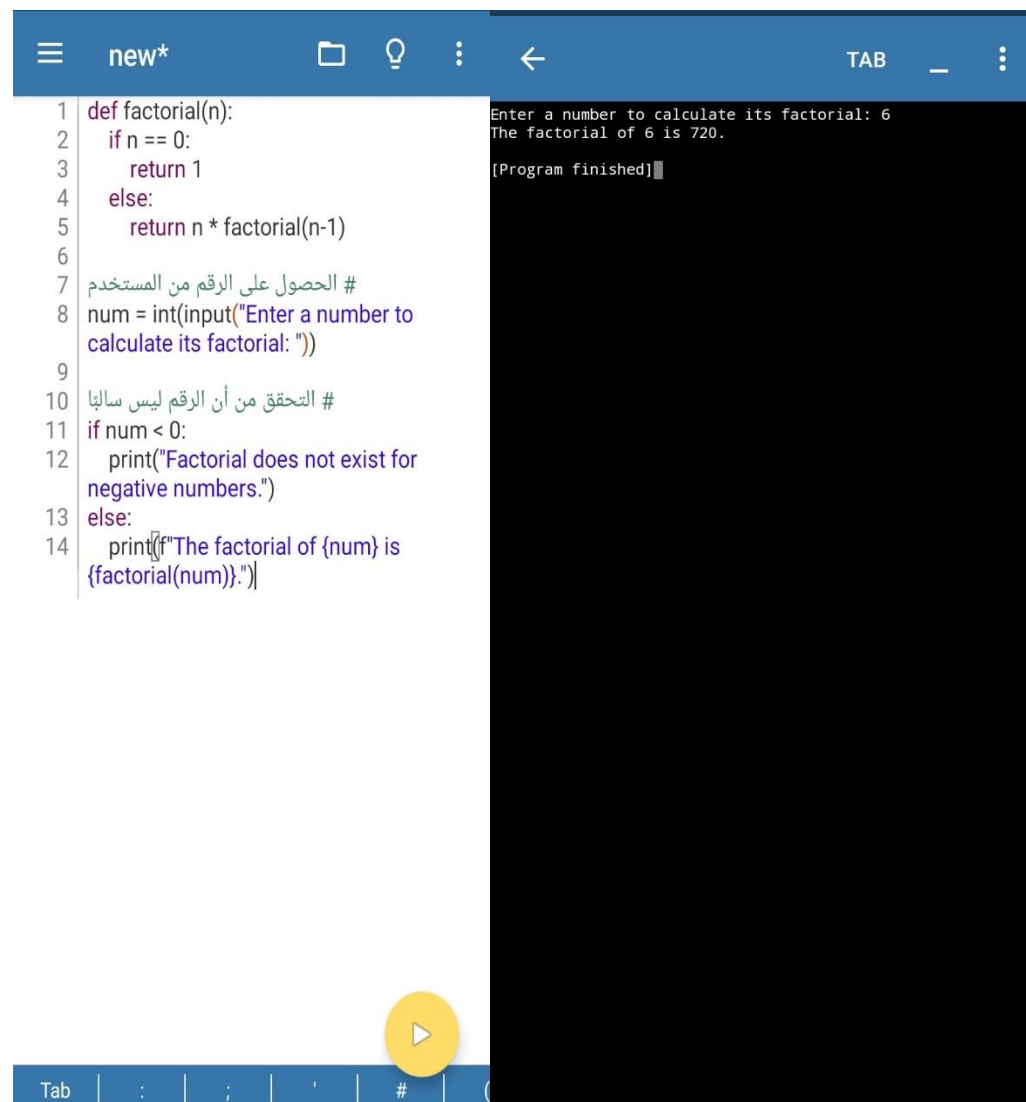


```
1 L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
2 L2 = [80, 443, 21, 53]
3
4 d = dict(zip(L1, L2))
5 print(d)
```

```
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
[Program finished]
```

أن التعليمة () Zip تدمج القائمتين معاً ضمن القاموس الذي تم انشاءه.

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

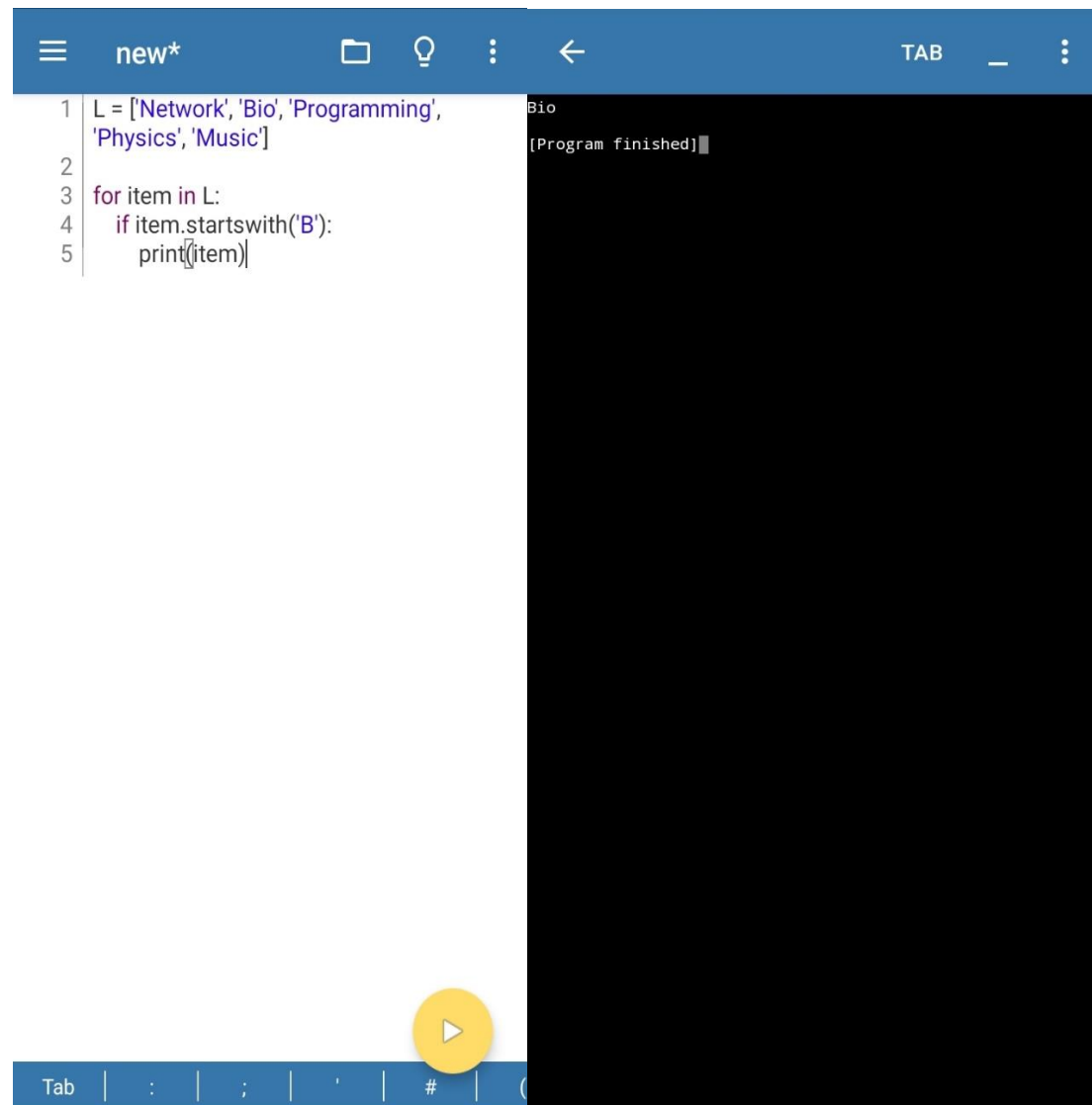


```
1 def factorial(n):
2     if n == 0:
3         return 1
4     else:
5         return n * factorial(n-1)
6
7 # الحصول على الرقم من المستخدم
8 num = int(input("Enter a number to
9 calculate its factorial: "))
10
11 # التحقق من أن الرقم ليس سالباً
12 if num < 0:
13     print("Factorial does not exist for
14 negative numbers.")
15 else:
16     print(f"The factorial of {num} is
17 {factorial(num)}.")
```

Enter a number to calculate its factorial: 6
The factorial of 6 is 720.
[Program finished]

أولاً قممت بتعريف تابع اسمه `factorial(n)` هذا التابع يأخذ بارامتر واحد وهو العدد الذي نريد حساب العامل له، إذا كانت $n=0$ فيعيد الرقم 1 وإلا فيعيد قيمة العدد n مضروب بالتابع بعد إنقاص قيمة n بمقدار واحد بشكل تكراري حتى يصل العدد إلى قيمة 0، يطلب البرنامج من المستخدم إدخال عدد ويحول هذا الإدخال إلى عدد صحيح ثم يتحقق من أن الرقم الذي أدخله المستخدم ليس سالباً، إذا كان العدد سالباً فيطبع رسالة العدد السالب ليس له عاملي وإلا يطبع ما تم حسابه عن طريق التابع المعروف.

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.



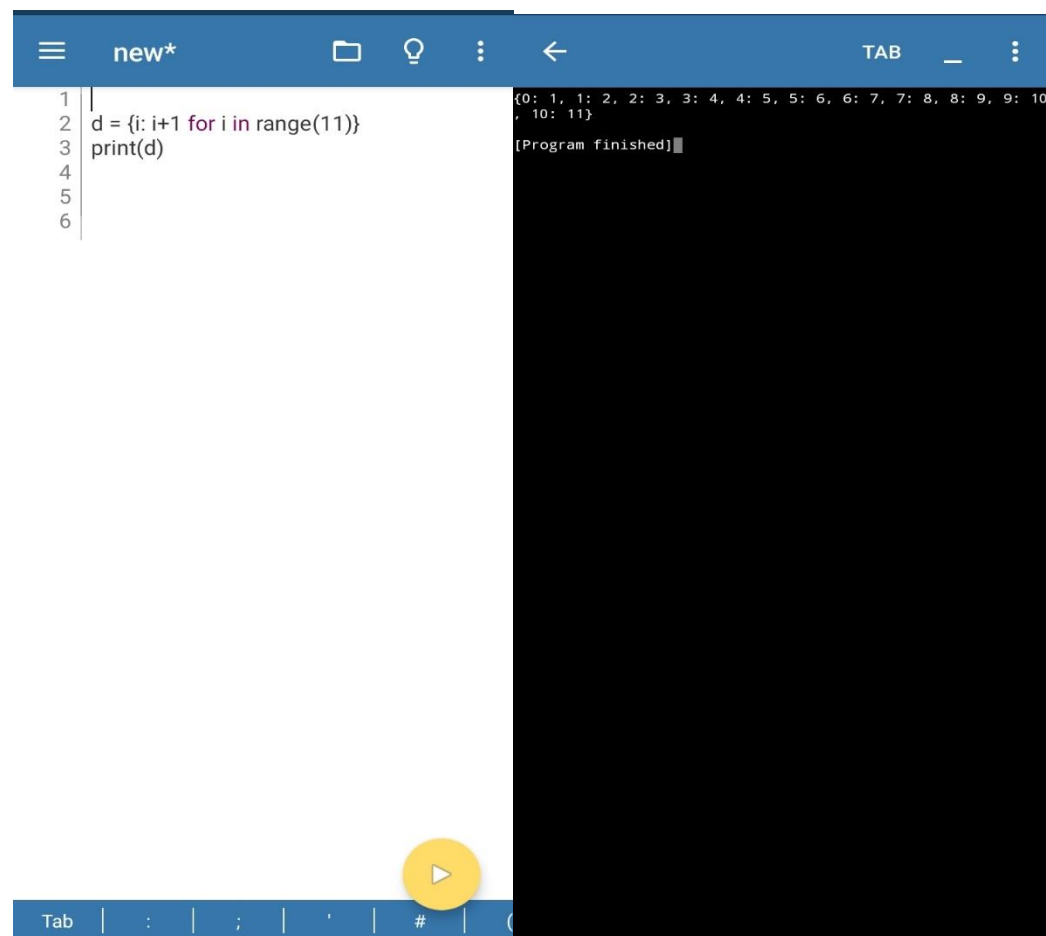
```
1 L = ['Network', 'Bio', 'Programming',  
2   'Physics', 'Music']  
3 for item in L:  
4     if item.startswith('B'):  
5         print(item)
```

Bio
[Program finished]

تم تعريف قائمة تحتوي على العناصر المذكورة سابقاً، ثم أنشأت حلقة for تمر على عناصر هذه القائمة، إذا كان العنصر يبدأ بحرف B تطبع هذا العنصر.

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}`.



The screenshot shows a code editor with a blue header bar containing icons for menu, file, search, and navigation. The editor has a tab labeled 'new*'. The code in the editor is:

```
1 |
2 | d = {i: i+1 for i in range(11)}
3 | print(d)
4 |
5 |
6 |
```

The output on the right side of the editor is:

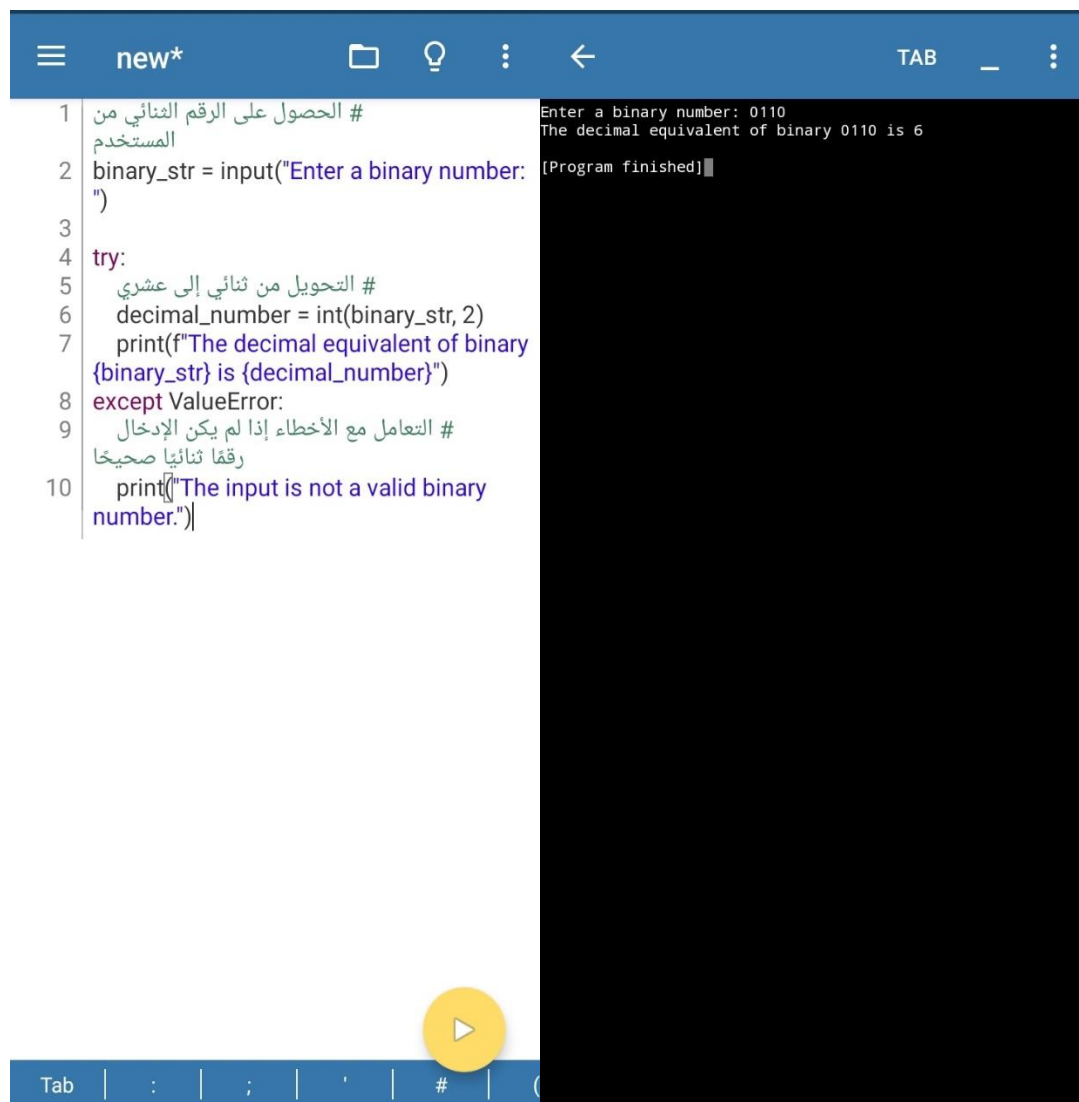
```
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
[Program finished]
```

A yellow play button icon is visible at the bottom center of the editor.

أنشأت قاموس يحتوي على أزواج مفتاح -قيمة ، كل مفتاح i له قيمة مرتبطة به هي العدد الصحيح التالي له $i+1$ ، ثم أنشأت حلقة for للمفتاح من 0 حتى 10 ضمناً وتقوم بطباعة المفتاح مترافقاً مع قيمته.

Question 2:

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.



```
1 # الحصول على الرقم الثنائي من المستخدم
2 binary_str = input("Enter a binary number: ")
3
4 try:
5     # التحويل من ثنائي إلى عشري
6     decimal_number = int(binary_str, 2)
7     print(f"The decimal equivalent of binary {binary_str} is {decimal_number}")
8 except ValueError:
9     # التعامل مع الأخطاء إذا لم يكن الإدخال رقمًا ثنائيًا صحيحًا
10    print("The input is not a valid binary number.")
```

Enter a binary number: 0110
The decimal equivalent of binary 0110 is 6
[Program finished]

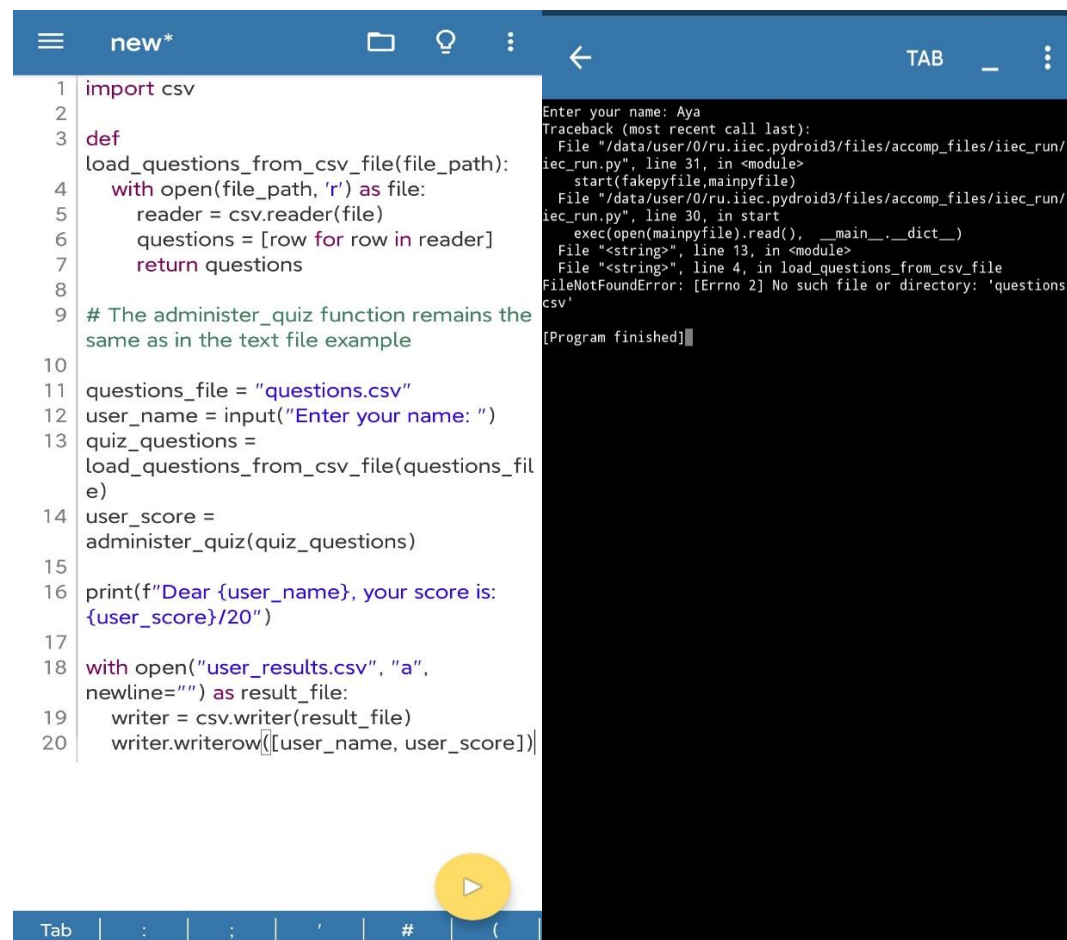
في البداية يطلب البرنامج من المستخدم إدخال رقم ثنائي ويخزن هذا الإدخال كنص (String) في المتغير `binary_str`، بعد ذلك يبدأ بـ `try` ويضع ضمنه الكود الذي قد يولد أخطاء ثم يقوم

بتحويل النص الثنائي إلى عدد عشري باستخدام int مع تمرير 2 كبارامتر ثانٍ ويخزن العدد العشري الناتج في المتغير decimal_number.

إذا كان الإدخال صحيحاً وتم التحويل بنجاح يطبع البرنامج النتيجة العشرية على الشاشة أما إذا حصل خطأ ValueError بسبب إدخال غير صحيح فإنه يستخدم except للتعامل معها ويطبع البرنامج رسالة تقول بأن الإدخال ليس رقماً صحيحاً.

Question 3:

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.



```
1 import csv
2
3 def
4 load_questions_from_csv_file(file_path):
5     with open(file_path, 'r') as file:
6         reader = csv.reader(file)
7         questions = [row for row in reader]
8         return questions
9
10 # The administer_quiz function remains the
11 # same as in the text file example
12
13 questions_file = "questions.csv"
14 user_name = input("Enter your name: ")
15 quiz_questions =
16 load_questions_from_csv_file(questions_fil
17 e)
18 user_score =
19 administer_quiz(quiz_questions)
20
21 print(f"Dear {user_name}, your score is:
22 {user_score}/20")
23
24 with open("user_results.csv", "a",
25 newline="") as result_file:
26     writer = csv.writer(result_file)
27     writer.writerow([user_name, user_score])
```

```
Enter your name: Aya
Traceback (most recent call last):
  File "/data/user/0/ru.iiec.pydroid3/files/accomp_files/iiec_run/i
  iec_run.py", line 31, in <module>
    start(fakepyfile,mainpyfile)
  File "/data/user/0/ru.iiec.pydroid3/files/accomp_files/iiec_run/i
  iec_run.py", line 30, in start
    exec(open(mainpyfile).read(), __main__.__dict__)
  File "<string>", line 13, in <module>
  File "<string>", line 4, in load_questions_from_csv_file
FileNotFoundError: [Errno 2] No such file or directory: 'questions.
csv'

[Program finished]
```

هذا الكود يقوم بتحميل أسئلة اختبار من ملف csv، يدير الاختبار ويحفظ نتائج المستخدم في ملف csv آخر :

1 -استيراد مكتبة csv :

-import csv :

هذا السطر يقوم باستيراد مكتبة csv والتي تستخدم للتعامل مع بيانات csv في بايثون.

2- تعريف التابع

load_questions_from_csv_file :

هذا التابع يقوم بتحميل الأسئلة من ملف csv.

-file_path:

هو المسار إلى ملف csv الذي يحتوي على الأسئلة.

-with open(file_path, 'r') as file:

يُستخدم لفتح الملف للقراءة ('r' تعني قراءة) ويتم التعامل مع الملف تحت المتغير file .

-reader = csv.reader(file):

يُنشئ كائن قارئ (reader) يستخدم لقراءة البيانات من الملف .

-questions = [row for row in reader]:

تقوم بإنشاء قائمة من الصفوف حيث كل صف يمثل سؤال .

-return questions :

تعيد الأسئلة المحملة .

3 – التعامل مع ملف الأسئلة :

-questions_file ="questions .csv ":

يُعرف مسار ملف csv الذي يحتوي على الأسئلة .

-quiz_questions = load_questions _from_csv_file(questions_file):

يتم تحميل الأسئلة من الملف وتخزينها في المتغير quiz_questions .

4 -أخذ اسم المستخدم وتشغيل الاختبار :

-user_name = input ("Enter your name : "):

يطلب من المستخدم إدخال اسمه.

-user_score = admininister_quiz(quiz_questions):

يُفترض أن هناك تابع `admininister_quiz` يقوم بإدارة الاختبار ويُرجع النتيجة .

5 -طباعة نتيجة المستخدم : يتم طباعة نتيجة المستخدم مع رسالة تبين النتيجة التي حصل عليها من أصل 20 .

6 – حفظ نتائج المستخدم في ملف `.csv`.

Question 4:

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.


```

new*
1 class BankAccount:
2     def __init__(self, account_number,
3         account_holder, balance=0.0):
4         self.account_number =
5             account_number
6         self.account_holder = account_holder
7         self.balance = balance
8
9     def deposit(self, amount):
10        if amount > 0:
11            self.balance += amount
12            print(f"Deposited ${amount}. New
13                balance is ${self.balance}.")
14        else:
15            print("Deposit amount must be
16                positive.")
17
18    def withdraw(self, amount):
19        if amount > 0 and amount <= self.
20            balance:
21            self.balance -= amount
22            print(f"Withdrew ${amount}. New
23                balance is ${self.balance}.")
24        else:
25            print("Insufficient balance or
26                invalid withdrawal amount.")
27
28    def get_balance(self):
29        return self.balance
30
31    def __str__(self):
32        return f"Account Number: {self.
33            account_number}, Account Holder: {self.
34            account_holder}, Balance: ${self.balance}"
35
36 SavingsAccount تعريف الفئة الفرعية #
37 class SavingsAccount(BankAccount):
38     def __init__(self, account_number,
39         account_holder, interest_rate, balance=0.
40         0):
41         super().__init__(account_number,
42             account_holder, balance)
43         self.interest_rate = interest_rate
44
45     def apply_interest(self):
46         interest = self.balance * self.
47             interest_rate / 100
48         print(f"Applying interest: ${interest}")
49         self.deposit(interest)
50
51     def __str__(self):
52         return super().__str__() + f", Interest
53             Rate: {self.interest_rate}%"
54
55 BankAccount إنشاء مثال من #
56 account = BankAccount("123456", "John
57     Doe")
58 account.deposit(1000)
59 account.withdraw(500)
60 print(account)

```

```

46 print(account)
47 SavingsAccount إنشاء مثال من #
48 savings_account =
49     SavingsAccount("789012", "Jane Doe",
50         interest_rate=5)
51 savings_account.deposit(1000)
52 savings_account.apply_interest()
53 print(savings_account)

```



Tab | : | ; | ' | # | (

يتم تعريف كلاس BankAccount الذي يمثل حساب مصرفي وله الخصائص التالية :

-account_number: رقم الحساب

-account_holder: اسم صاحب الحساب

-balance:0.0 الرصيد في الحساب ،والذي يبدأ بقيمة افتراضية

يحتوي على الطرق (الوظائف) التالية :

init: تابع باني للكلاس، تُستخدم لإعداد الكائنات الجديدة

-deposit: لإيداع مبلغ في الحساب ،يجب أن يكون المبلغ موجباً

-withdraw: لسحب مبلغ من الحساب ،لايمكن سحب مبلغ أكبر من الرصيد

-get_balance: لاسترجاع الرصيد الحالي للحساب

str:

لطباعة معلومات الحساب بتنسيق من print على كائن محدد عند استدعاء bankaccount.

الكلاس savingsaccount :

هو كلاس فرعي يرث من bankaccount ويمثل حساب توفير يضيف فائدة ، يحتوي على

خاصية إضافية : interset_rate- وهي نسبة الفائدة ، ويحتوي على طريقة إضافية وهي : -

apply_interest لتطبيق الفائدة على الرصيد وإيداعها في الحساب

إنشاء الكائنات واستخدام الكلاسات :

يتم إنشاء كائن account من كلاس bankaccount وإجراء عمليات إيداع وطباعة المعلومات.

يتم إنشاء كائن savings_account من كلاس savingsaccount وإجراء عملية إيداع وتطبيق

الفائدة وطباعة المعلومات .