

Name: _____ حيدرة بديع صالح _____, Number: 1862 _____, Submitted To
GitHub: _____

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 }

Answer:

```
L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
```

```
L2 = [80, 443, 21, 53]
```

```
d1 = dict(zip(L1, L2))
```

```
print(d1)
```

الشرح:

قمنا بإنشاء قائمتين، L1 و L2 تحتوي L1 على سلاسل و L2 تحتوي أعداد صحيحة. ثم استخدمت zip الدالة المضغوطة لوضع عناصر L1 و L2 معاً في قاموس يسمى d1. سيحتوي القاموس الناتج d1 على مفاتيح من L1 وقيم من L2 في هذه الحالة، ستكون المفاتيح هي "HTTP" و "HTTPS" و "FTP" و "DNS"، وستكون القيم المقابلة 80 و 443 و 21 و 53.

الخرج :

```
==== RESTART: C:\Users\we\AppData\Local\Programs\Python\Python312\quiz.py ====
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
```

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

Answer:

```
num = int(input("Enter a number: "))
```

قمنا بتعريف num من النوع int يطلب من المستخدم ادخال رقم صحيح

```
def factorial(n):
```

تعريف دالة 'factorial' تأخذ المعامل n وظيفة هذه الدالة هو حساب العامل ل العدد المدخل n

```
if n == 0:
```

```
    return 1
```

استخدام الشرط if حيث اذا قام المستخدم بإدخال الرقم 0 لا يتوقف الكود بل يعيد القيمة 1

```
else:
```

```
    return n * factorial(n-1)
```

اذا كان العدد المدخل غير 0 يقوم الكود بحساب العملية للرقم المدخل عن طريق ضرب العدد n بالعامل ل n-1

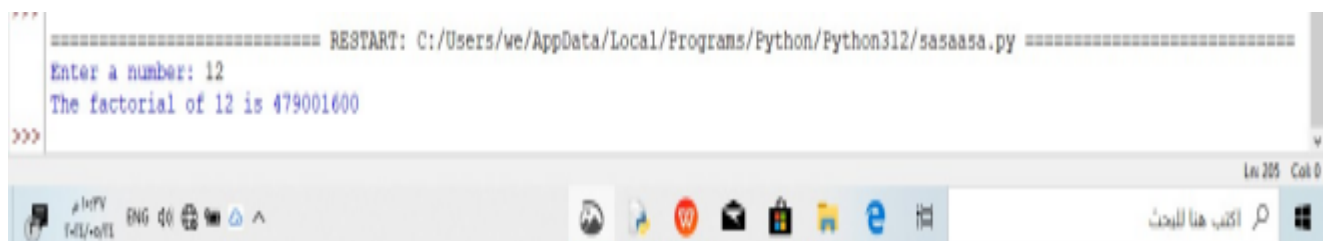
```
fact = factorial(num)
```

استدعاء الدالة 'factorial' بالعدد المدخل 'num' وتخزين النتيجة في المتغير fact

```
print(f"The factorial of {num} is {fact}")
```

طباعة قيمة fact , num باستخدام السلسلة f حيث تقوم بدمج المتغيرات مباشرة داخل السلسلة string باستخدام الأقواس المتعرجة {} وبالتالي يظهر في الخرج قيمة كل من fact , num ضمن السلسلة

الخرج:



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.

Answer:

L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']

قمنا بإدخال القائمة المعطاة في نص السؤال التي تحوي على خمس عناصر من نوع string

for x in L:

ثم قمنا بتعريف حلقة for لكي نقوم باختبار الشرط على كل عناصر القائمة L

if x.startswith('B'):

print(x)

بالنسبة لكل عنصر 'x' في القائمة، فإنه يتحقق مما إذا كان العنصر يبدأ بالحرف 'B' باستخدام 'startswith'. إذا تحقق الشرط (أي إذا كان العنصر يبدأ بـ "B")، فسيقوم الكود بطباعة هذا العنصر.

الخرج:

```
===== RESTART: C:/Users/we/AppData/Local/Programs/Python/Python312/sasaasa.py =====
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}

Answer:

```
d2 = {i: i+1 for i in range(11)}
```

```
print(d2)
```

الشرح :

قمنا بإنشاء قاموس 'd2' باستخدام Dictionary comprehension في هذه الحالة، يتم تكرار الأرقام من 0 إلى 10 باستخدام حلقة for

وبالتالي يصبح range(11) حيث يمثل 'i' كل رقم ضمن المجال. و لكل 'i' في المجال يقوم Dictionary comprehension

بإنشاء زوج مفتاح - قيمة في القاموس 'd2' حيث يكون المفتاح 'i' والقيمة 'i+1' هذا يعني أن كل مفتاح في القاموس سيكون رقم من المجال، و القيمة المقابلة ستكون الرقم نفسه مضاف اليه واحد .

وفي النهاية نقوم بطباعة القيم ضمن قاموس وتظهر بشكل مفتاح - قيمة

الخرج:

```
===== RESTART: C:/Users/we/AppData/Local/Programs/Python/Python312/sasaasa.py =====
{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

Answer:

```
def binary_to_decimal(binary_str):
```

```
    try:
```

```
        for digit in binary_str:
```

```
            if digit not in ['1', '0']:
```

```

        raise ValueError("Input must be only 1s or 0s.")

    c

    return dec

except ValueError as s:

    print(s)

    return None

binary_number = input("Enter a binary number: ")

decimal_number = binary_to_decimal(binary_number)

if decimal_number is not None:

    print(f"The decimal number is: {decimal_number}")

```

الشرح:

يطلب الكود من المستخدم ادخال رقم ثنائي بعد ادخال الرقم يتم اسناده الى التابع **binary_n** **umber** ثم يقوم الكود باستدعاء الرقم المدخل ويقوم بتخزينه في التابع **decimal_number**

يتم اسناد القيمة المدخلة الى التابع **binary_to_decimal** الذي يأخذ مدخل واحد من نوع string "binary_str".

ضمن التابع **binary_to_decimal** قمنا بتعريف حلقة ال for لتكرار اختبار الارقام المدخلة من المستخدم نتسخدم التابع **try** لكي لا يتوقف البرنامج عن العمل في حال قام المستخدم بإدخال رقم غير 0 او 1 يتم و الانتقال الى **except ValueError** ويطلع رسالة الخطأ ولا يعيد قيمة **return None**

يتم قبول الرقم اذا كان يحوي 0 او 1 فقط و اي رقم اخر يتم اظهار رسالة Input must be only 1s or 0s. اذا كانت كافة الأرقام صحيحة، فإنه يحول السلسلة الثنائية إلى رقم عشري باستخدام التابع **int(binary_str, 2)** ذات الأساس 2 (ثنائي). ويتم اسناد القيمة بعد التحويل الى

المتغير dec يقوم التابع binary_to_decimal بإرجاع الرقم العشري من خلال المتغير dec في حالة نجاح التحويل. إذا لم يكن dec=None (أي كان التحويل ناجحًا)، فسيتم طباعة الرقم العشري باستخدام تنسيق string f.

الخرج:

```
= RESTART: C:/Users/we/AppData/Local/Programs/Python/Python312/0000.py
Enter a binary number: 1100
The decimal number is: 12
```

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.

Answer:

```
import json
```

```
import csv
```

```
def calculate_score(user_answers, correct_answers):
```

```
    score = 0
```

```
    for question, answer in user_answers.items():
```

```
        if answer == correct_answers.get(question):
```

```
            score += 1
```

```
    return score
```

```

def save_results(user_name, user_answers, score, file_format):

    file_name = f'{user_name}_quiz_results.{file_format}'

    if file_format == 'json':

        with open(file_name, 'w') as json_file:

            json.dump({'name': user_name, 'resu': f'{score}/20', 'ans': user_answers},
            json_file)

            print('save is done as .json')

    elif file_format == 'csv':

        with open(file_name, 'w', newline='') as csv_file:

            csv_writer = csv.writer(csv_file)

            csv_writer.writerow(['Question', 'User Answer'])

            for question, answer in user_answers.items():

                csv_writer.writerow([question, answer])

            csv_writer.writerow(['Score', f'{score}/20'])

            print('result save as csv.')

    else:

        print('not save the file dosnt support.')

user_name = input('your name: ')

file_name = input('file ques and ans: ')

with open(file_name, 'r') as file:

```

```
w = file.read()

questions_and_answers = json.loads(w)

user_answers = {}

for question in questions_and_answers.keys():

    user_answer = input(f'Your answer for "{question}": ')

    user_answers[question] = user_answer

score = calculate_score(user_answers, questions_and_answers)

print(f'Your score is: {score}/20')

file_format = input('Choose the file format to save the results (JSON or CSV): ')

save_results(user_name, user_answers, score, file_format)
```

الخروج :


```

your name: haedara
file ques and ans: ques.txt
Your answer for " Sharks are mammals.? ":
Your answer for " It takes a sloth two weeks to digest a meal?": g
Your answer for " Greenland is the largest island in the world ?":
Your answer for " You can find the 'Desert of Death' in California.?":
Your answer for " China has the longest coastline in the world? ":
Your answer for " The human body has four lungs.?":
Your answer for " The human body is about 60% water? ":
Your answer for " Cheesecake comes from Italy.? ": f
Your answer for " Hawaiian pizza comes from Canada.? ": f
Your answer for " lattakia is the name of the capital of Syria? ": f
Your answer for " City of Angels is the meaning of the name Los Angeles? ": f
Your answer for " Pepperoni is the most popular pizza topping in the US.? ": f
Your answer for " Palestine is the name of the land that is the cradle of the heavenly religions that God Almighty swore by?": fq
Your answer for "Asia is the largest continent in the world by area? ": t
Your answer for "A dog sweats by panting its tongue? ": f
Your answer for "The total length of the Great Wall of China adds up to 13,171 miles? ":
Your answer for "Thomas Edison Who invented the light bulb?? ": f
Your answer for "Humans lose an average of 75 strands of hair a month? ": f
Your answer for "The most common blood type is 0- negative.?": t
Your score is: 2/20
Choose the file format to save the results (JSON or CSV): csv
result save as csv.

```

الشرح :

يقوم الكود باستيراد مكتبتي json و csv لمعالجة البيانات ذات تنسيقات json و csv

عند التشغيل يطلب من المستخدم ان يقوم بادخال اسمه ثم يطلب اسم الملف الذي يحوي على الأسئلة والأجوبة

with open(file_name, 'r') as file: تقوم بفتح ملف باسم **file_name** و ('r') تعني قراءة الملف دون التعديل عليه. باستخدام **with** لإغلاق الملف بعد الانتهاء من استخدامه

w = file.read() : يقوم بقراءة كامل محتوى الملف وتخزينه في المتغير **w**.

questions_and_answers = json.loads(w): هذا يقوم بتحويل محتوى الملف المخزن في **w** من تنسيق JSON إلى قاموس باستخدام التابع (**json.loads()**). النتيجة تُخزن في المتغير **questions_and_answers**

يصبح لدينا الان قاموس يحوي على الاسئلة ولأجوبة

نقوم بإنشاء قاموس فارغ بالاسم **user_answers** لتخزين إجابات المستخدم ثم باستخدام حلقة **for** نقوم ب تكرار التخزين على جميع مفاتيح القاموس ، والتي تمثل الأسئلة ثم يقوم بالطلب من المستخدم إدخال إجابته على السؤال الحالي و يقوم بتخزين إجابة المستخدم في القاموس **user_answers** مقابل المفتاح الذي يمثل السؤال وفي نهاية هذه الحلقة، سيكون قاموس يحتوي على إجابات المستخدم لجميع الأسئلة

يقوم بإرسال القاموسين `questions_and_answers` و `user_answers` إلى التابع `calculate_score` والذي يقوم بحساب درجة المستخدم. حيث ان النتيجة يتم تخزينها في المتغير `score` بقيمه 0 و عن طريق حلقة `for` تختبر كل إجابة فإذا كانت صحيحة تقوم بزيادة `score` بقيمة 1 ثم تعيد قيمة `score` ويقوم بطباعة عدد الاجابات الصحيحة بالنسبة لعدد الاسئلة ثم يقوم بالطلب من المستخدم اختيار صيغة الملف المراد ان تظهر النتائج به إما `json` أو `csv` و يقوم التابع `save_results` بأختبار صيغة الملف المراد الحفظ به إذا كانت من النوع `json` او `csv` يقوم بحفظ اسم المستخدم والاجابات والنتيجة وصيغة الملف في ملف من النوع الذي تم اختياره ويقوم بطباعة تم الحفظ بالصيغة التي ادخلها المستخدم وإذا ادخل المستخدم صيغة غير `json` او `csv` يقوم بطباعة الملف غير مدعوم لم يتم الحفظ .

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.

Answer:

class `BankAccount`:

```
def __init__(s, acc_num, acc_hol):
```

```
    s.acc_num = acc_num
```

```

    s.acc_hol = acc_hol

    s.balance = 0.0

def deposit(s, amount):

    s.balance += amount

def withdraw(s, amount):

    if s.balance >= amount:

        s.balance -= amount

    else:

        print("Insu funds.")

def get_balance(s):

    return s.balance

class SavingsAccount(BankAccount):

    def __init__(s, acc_num, acc_hol, interest_rate):

        super().__init__(acc_num, acc_hol)

        s.interest_rate = interest_rate

    def apply_interest(s):

        s.balance += s.balance * s.interest_rate

    def __str__(s):

        return f"Account Holder: {s.acc_hol}\nAccount Balance: ${s.balance:.2f}\nInterest Rate: {s.interest_rate*100:.2f}%"

bank_account = BankAccount("123456789", "haedara s")

bank_account.deposit(1000)

print(f"Current Balance: ${bank_account.get_balance():.2f}")

bank_account.withdraw(500)

```

```

print(f"Current Balance: ${bank_account.get_balance():.2f}")

savings_account = SavingsAccount("0000000", "hadi s", 0.05)

savings_account.apply_interest()

print(savings_account)

```

الشرح:

لدينا كلاس **BankAccount** يحوي التابع الباني :
def __init__(s, acc_num, acc_hol) يأخذ اربع متغيرات

و التابع **deposit(s, amount)** و التابع **account.withdraw** يقوم باختبار الرصيد الذي تم ايداعه اذا كان الرصيد الموجود اكبر من الرصيد المسحوب يقوم بطرح الرصيد الحالي من الرصيد المسحوب والا يطبع Insu funds.

و التابع **get_balance(s):** يعيد قيمة الرصيد الموجود في الحساب

الكلاس **SavingsAccount**: يرث من **BankAccount** ويضيف معدل الفائدة كخاصية حيث يستطيع الوصول الى الكلاس الاب باستخدام **super().__init__(acc_num, acc_hol)**

ويحوي على تابع لحساب الرصيد بعد ضربه ب الفائدة

عند التنفيذ يقوم البرنامج ب اظهار الرصيد الحالي الموجود في الحساب و قيمة الرصيد بعد السحب ثم يظهر اسم العميل و الرصيد بعد تطبيق الفائدة و قيمة الفائدة

الخرج:

```

===== RESTART: C:/Users/we/AppData/Local/Programs/Python/Python312/0000.py =====
Current Balance: $1000.00
Current Balance: $500.00
Account Holder: hadi s
Account Balance: $0.00
Interest Rate: 5.00%
>>

```