علي غازي زيني ١٠٧١

السؤال الأول:

```
print("السؤال الأول")
print("الطلب A")
d={}
def func(k,v):
    d[k]=v
L1=['HTTP','HTTPS','FTP','DNS']
L2=[80,443,20,53]
for i in range(len(L1)):
    func(L1[i],L2[i])
print(d)

print(50*"*")
print("الطلب B")

def f(n):
    x=1
    if n>0:
        for i in range(1,n+1):
            x=x*i
        return x
    elif n==0:
        return 1
    else:
        return "error"
i=int(input("Enter number: "))
print(f(i))


print(50*"*")
print("الطلب C")

L=['Network','Bio','Programming','Physics','Music']
for i in range(len(L)):
    if L[i][:1]=='B':
        print(L[i])

print(50*"*")
print("الطلب D")

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

الخرج الخاص بالسؤال الأول:

```
السؤال الأول
A الطلب
{'HTTP': 80, 'HTTPS': 443, 'FTP': 20, 'DNS': 53}
************************************************
B الطلب
Enter number: 5
120
************************************************
C الطلب
Bio
************************************************
D الطلب
{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
```

السؤال الثاني:

```python
def converter(binary_num):
    decimal_num=0
    l=[]
    try:
        for i in binary_num:
            l.append(int(i))
        l.reverse()
        for i in range(len(l)):
            decimal_num+=l[i]*2**i
        return decimal_num
    except ValueError as e:
        return e

b=input("input binary number: ")
print(converter(b))
```

تم تعريف تابع "converter" الذي يأخذ عدد ثنائي كمدخل وقيمة عشرية كمخرج. هنا كيفية عمل هذا التابع:

1. **تم تعريف متغير "decimal_num" لتخزين القيمة العشرية للرقم الثنائي.**
2. **تم إنشاء قائمة فارغة "l" لتخزين أرقام الرقم الثنائي.**
3. **تم استخدام تعبير try-except لمعالجة أي قيم غير صالحة (غير رقمية) في الرقم الثنائي. إذا تم إدخال قيمة غير صالحة، يتم إرجاع الخطأ.**
4. **تم تحويل الرقم الثنائي إلى قائمة من الأرقام الصحيحة وعكس ترتيبها.**
5. **تم تكرار القائمة واحتساب القيمة العشرية باستخدام الصيغة.decimal_num += l[i] * 2^i :**
6. **تم إرجاع القيمة العشرية النهائية.**

الخرج الخاص بالسؤال الثاني:

```
input binary number: 110011
51

Process finished with exit code 0
```

السؤال الثالث:

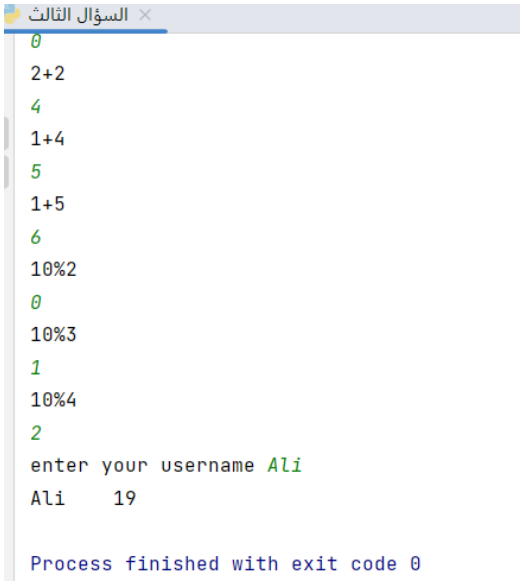
```
import json
filename="t.txt"
infile=open(filename,'r')
s=infile.readlines()
infile.close()
c=0
for i in s:
    qa=i.rstrip().split(",")
    print(qa[0])
    a=input()
    if a==qa[-1]:
        c+=1
sname=input("enter your username ")
print(sname,'  ',c)
outfile=open("Ali.json",'w')
json.dump({sname:c},outfile)
outfile.close()
```

يتم فتح الملف النصي في وضع القراءة وقراءة جميع السطور وتخزينها في القائمة s وبعدها يتم إغلاق الملف. يتم تهيئة متغير c إلى 0 لتتبع عدد الإجابات الصحيحة. ثم يتم تكرار كل سطر في القائمة "s" بالنسبة لكل سطر، يتم فصل السطر إلى قائمة من السلاسل باستخدام الطريقة"split(',')" ، ويتم عرض السلسلة الأولى (الفهرس 0) على المستخدم كسؤال. يُطلب من المستخدم إدخال إجابة، وإذا كانت إجابة المستخدم تطابق السلسلة الأخيرة في القائمة (الفهرس -1)، يتم زيادة "c" بمقدار 1.

بعد الانتهاء من تكرار جميع السطور، يُطلب من المستخدم إدخال اسم المستخدم الخاص به، ثم يتم عرض اسم المستخدم والعدد النهائي للإجابات الصحيحة.

يتم إنشاء ملف JSON بالاسم "Ali.json" في وضع الكتابة، ويتم تخزين معلومات المستخدم (الاسم وعدد الإجابات الصحيحة) في وكتابته في الملف.

الخرج الخاص بالسؤال الثالث:



ملف الأسئلة:

```
1+1,2
2+2,4
1+4,5
1+5,6
3+6,9
5+2,7
1-1,0
9%3,0
9%4,1
8/2,4
1+5,6
3+6,9
5+2,7
1-1,0
2+2,4
1+4,5
1+5,6
10%2,0
10%3,1
10%4,2
```

ملف العلامة:

```
{"Ali": 19}
```

السؤال الرابع

```
# تعريف كلاس BankAccount مع الخصائص والطرق المطلوبة
class BankAccount:
    def __init__(self, account_number, account_holder):
        self.account_number = account_number # رقم الحساب
        self.account_holder = account_holder # اسم صاحب الحساب
        self.balance = 0.0 # رصيد الحساب، يبدأ بصفر

    def deposit(self, amount):
        # إيداع طريقة المبلغ في الحساب
        self.balance += amount

    def withdraw(self, amount):
        # سحب طريقة المبلغ من الحساب
        self.balance -= amount

    def get_balance(self):
        # للحصول طريقة على رصيد الحساب الحالي
        return self.balance

# إنشاء كائن من كلاس BankAccount
account = BankAccount("1071", "زيني علي")

# إيداع مبلغ 1000 دولار
account.deposit(1000)
print("الرصيد الحالي:", account.get_balance()) # 1000.0

# سحب مبلغ 500 دولار
account.withdraw(500)
print("الرصيد الحالي:", account.get_balance()) # 500.0
print("صاحبُ الحساب:", account.account_holder)
# تعريف كلاس SavingsAccount التي ترث من كلاس BankAccount
class SavingsAccount(BankAccount):
    def __init__(self, account_number, account_holder, interest_rate):
        super().__init__(account_number, account_holder)
        self.interest_rate = interest_rate # معدل الفائدة

    def apply_interest(self):
        # تطبيق طريقة على الفائدة لحساب رصيد
        self.balance += self.balance * self.interest_rate

    def __str__(self):
        # المخصصة print() طريقة تعريف
        return f"الرصيد الحالي: {self.get_balance():.2f} دولار، معدل الفائدة:
{self.interest_rate:.2%}"

# إنشاء كائن من كلاس SavingsAccount
saccount = SavingsAccount("11010", "زيني علي", 0.05)

# تطبيق الفائدة على رصيد الحساب
saccount.apply_interest()
print(saccount) # الرصيد الحالي: 0.00 دولار، معدل الفائدة: 5.00%
```

الخرج الخاص بالسؤال الرابع:

```
الرصيد الحالي: 1000.0
الرصيد الحالي: 500.0
صاحب الحساب: علي زيني
الرصيد الحالي: 0.00 دولار، معدل الفائدة: %5.00

Process finished with exit code 0
```