السؤال الأول:

**12**

|  |  |  |  |
| --- | --- | --- | --- |
| م | المهارة المطلوبة | الدرجة | الدرجة المستحقة |
| 1 | تشغيل برنامج DB Browser for SQlite | 1 |  |
| 2 | فتح قاعدة البيانات  | 1 |  |
| 3 | تحميل ملف سجل المحفوظات  | 2 |  |
| 4 | عرض جدول عناوين Urls  | 1 |  |
| 5 | اختاري أحد العناوين الموجودة وحددي وقت الزيارة الأخير  | 1 |  |
| 6 | قومي بقراءة ختم الوقت الخاص بالزيارة الأخيرة التي قمت بتحديدها  | 2 |  |
| 7 | عرض جدول التحميلات  | 1 |  |
| 8 | اختيار أحد الملفات وتحديد وقت انتهاء التحميل  | 1 |  |
| 9 | قومي بقراءة ختم الوقت الخاص بانتهاء تحميل الملف الذي قمت باختياره  | 2 |  |
| المجموع | 12 |  |

باستخدام متصفح DB قومي بالآتي :

السؤال الثاني:

**6**

أمامك مجموعه من الأذونات صلي كل إذن بالوصف الخاص به:

|  |  |  |  |
| --- | --- | --- | --- |
| م | الإذن | الاجابة | الوصف |
| 1 | Full control | 4 | يسمح للمستخدمين بعرض الملفات والمجلدات |
| 2 | Modify | 3 | يسمح للمستخدمين بقراءة وعرض الملفات والمجلدات وتنفيذها |
| 3 | Read Execute | 5 | يسمح بإنشاء ملفات ومجلدات جديده |
| 4 | Read | 2 | يسمح للمستخدمين بتعديل الملفات والمجلدات بما في ذلك إنشاء ملفات ومجلدات فرعية جديده |
| 5 | Write | 1 | يوفر للمستخدم تحكم كاملاً في الملف او المجلد |
| 6 | List Folder Content |

السؤال الثاني:

**8**

أمامك خوارزمية تشفير للنصوص تعرفي عليها وأجيبي على المطلوب :

**def caesar\_encrypt**(**message, KEY**):

 alphabet\_lower = "abcdefghijklmnopqrstuvwxyz"

 alphabet\_upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

 encrypted\_message = ""

 **for** char in message:

 **if** char in alphabet\_lower:

 char\_index = alphabet\_lower.find(char)

 new\_char\_index = (char\_index + key) % 26

 encrypted\_message += alphabet\_lower[new\_char\_index]

 **elif** char in alphabet\_upper:

 char\_index = alphabet\_upper.find(char)

 new\_char\_index = (char\_index + key) % 26

 encrypted\_message += alphabet\_upper[new\_char\_index]

 **else:**

 encrypted\_message += char

 **retu**rn **encrypted\_message**

1. اسم الخوارزمية: شفرة قيصر (1 درجة)
2. أكملي الفراغات الموجودة في الكود: (3 درجات) **message, KEY ، encrypted\_message**
3. باستخدام مفكرة جوبيتر قومي بكتابة أكواد التشفير وفك التشفير (3 درجات)
4. باستخدام مفتاح 4 قومي بتشفير كلمة (Cybersecurity) (1 درجة) (Hdgjwxjhzwnyd)

def caesar\_encrypt(message, KEY):

 alphabet\_lower = "abcdefghijklmnopqrstuvwxyz"

 alphabet\_upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

 encrypted\_message = ""

 for char in message:

 if char in alphabet\_lower:

 char\_index = alphabet\_lower.find(char)

 new\_char\_index = (char\_index + key) % 26

 encrypted\_message += alphabet\_lower[new\_char\_index]

 elif char in alphabet\_upper:

 char\_index = alphabet\_upper.find(char)

 new\_char\_index = (char\_index + key) % 26

 encrypted\_message += alphabet\_upper[new\_char\_index]

 else:

 encrypted\_message += char

 return encrypted\_message

كود التشفير

def caesar\_decrypt(encrypted\_message, KEY):

 alphabet\_lower = "abcdefghijklmnopqrstuvwxyz"

 alphabet\_upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

 decrypted\_message = ""

 for char in encrypted\_message:

 if char in alphabet\_lower:

 char\_index = alphabet\_lower.find(char)

 new\_char\_index = (char\_index - key) % 26

 decrypted\_message += alphabet\_lower[new\_char\_index]

 elif char in alphabet\_upper:

 char\_index = alphabet\_upper.find(char)

 new\_char\_index = (char\_index - key) % 26

 decrypted \_message += alphabet\_upper[new\_char\_index]

 else:

 decrypted \_message += char

 return decrypted \_message

كود فك التشفير