العمل التطبيقي 3:

الهدف: التمرن على كتابة الخوارزميات ثم ترجمتها إلى برامج بلغة (بايثون) وتصحيحها وتنفيذها.

المسألة 1 : نظام مبسط لتسيير معلومات ونتائج الطلبة "*Student Management System*":

يهدف هذا التمرين إلى تطوير برنامج بلغة Python يقوم بمعالجة "معلومات مجموعة من الطلبة"، حيث يتمكن الطالب من خلاله من التعامل مع مختلف أنماط البيانات من خلال القراءة والإظهار، مع التمرن على استعمال مختلف تراكيب الشرط والحلقات، بالإضافة إلى معرفة كيفية تصحيح وتنفيذ البرنامج.

المرحلة الأولى: إنشاء البرنامج.

في البداية يطلب البرنامج من المستخدم إدخال المعطيات التالية لكل طالب:

- الاسم (سلسلة نصية)
- الجنس (حرف "M" :للذكر أو "F" للأنثى)
- العمر (عدد صحيح موجب)
- المعدل (عدد حقيقي)

في المرحلة الثانية، يقوم البرنامج بعد إدخال هذه البيانات بتحديد نتيجة كل طالب اعتمادًا على معدله، وفق القواعد التالية:

- إذا كان المعدل أكبر أو يساوي 10 "النتيجة: ناجح" (Result: Pass) .
- إذا كان المعدل أكبر أو يساوي 7 وأقل من 10 "النتيجة: استدراك" (Result: Retake) .
- إذا كان المعدل أقل من 7 " النتيجة: راسب" (Result: Fail) .

في المرحلة الأخيرة، يقوم البرنامج بحساب النسبة المئوية لكل فئة من الطلبة (الناجحين، المستدركين، والراسبين) ثم عرضها بشكل واضح.

ملاحظة:

يستحسن استعمال التعاليق (Comments) داخل البرنامج لشرح كل مرحلة من مراحل الحل وتوضيح عمل التعليمات المستعملة.

المرحلة الثانية:

إعادة صياغة تعليمات القراءة في البرنامج من أجل التحقق من القراءة الصحيحة لكل معلومة.

المرحلة الثالثة:

استعمال جداول للاحتفاظ بمعلومات الطلبة، ثم إظهار النتائج بالتفاصيل إذا أراد المستخدم ذلك.

المرحلة الرابعة:

بعد تصحيح كل الأخطاء ومعالجة كل الحالات الاستثنائية "الحدية" (*Edge cases*) يطلب إنشاء البرنامج التنفيذي (*Executable*).

المسألة 2: يوم من الأسبوع

1. اكتب خوارزم لحساب يوم الأسبوع الموافق لتاريخ ميلادي ما بالاعتماد على طريقة "**زيلر**" (***Zeller's Congruence***) :

$$h = \left(q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lfloor \frac{J}{4} \right\rfloor + 5J\right) mod\ 7$$

حيث:

- **q** : يمثل اليوم.
- **m** : يمثل رقم الشهر (3 : شهر مارس ، .. 12 : شهر ديسمبر ، شهري جانفي وفيفري يعتبران 13 و 14 على التوالي من السنة الماضية).
- **K** : يمثل الرقمين الأخيرين (الآحاد والعشرات) من السنة.
- **J** : يمثل رقمي المئات والآلاف من السنة.
- $\lfloor\ \rfloor$: يمثلان حاصل القسمة الصحيحة.
- **mod** تحسب باقي القسمة.

- ترجمة النتيجة: بما أن قيمة h تكون محصورة بين 0 و 6 فإن: 0 يمثل يوم السبت، 1 يمثل يوم الأحد، .. 6 يمثل يوم الجمعة.

2. حول هذا الخوارزم إلى برنامج في لغة بايثون ثم جربه على تواريخ تعرفها: مثل تاريخ اليوم، أو تاريخ 2026/02/28 الذي يوافق يوم السبت، تاريخ 2023/11/01 الذي يوافق يوم الأربعاء، تاريخ 1900/01/01 الذي يوافق يوم الإثنين.

3. اكتب نسختين من البرنامج:

- نسخة باستعمال التركيبة "if-elif" أو "match" من أجل تحويل الأرقام إلى أسماء الأيام.
- نسخة أخرى باستعمال قائمة بها أسماء الأيام وبالتالي الاستغناء عن التركيبتين المذكورتين.

مثال 1:

يوم 2026/02/15 (q = 15، m = 2، y = 2026) :

بما أن الشهر هو فيفري فإن m تصبح 14 و y تصبح 2025. إذن k = 25 و j = 20.

$$h = \left(15 + \left\lfloor \frac{13(14+1)}{5} \right\rfloor + 25 + \left\lfloor \frac{25}{4} \right\rfloor + \left\lfloor \frac{20}{4} \right\rfloor + 5 * 20\right) mod\ 7 = 190\ mod\ 7 = 1$$

النتيجة : هذا اليوم يو افق يوم الأحد.

مثال2:

يوم 2026/07/16 : q = 15، m = 7، y = 2026، k = 26 و j = 20 :

$$h = \left(16 + \left\lfloor \frac{13(7+1)}{5} \right\rfloor + 26 + \left\lfloor \frac{26}{4} \right\rfloor + \left\lfloor \frac{20}{4} \right\rfloor + 5 * 20\right) mod\ 7 = 173\ mod\ 7 = 5$$

النتيجة: هذا اليوم يو افق يوم الخميس.

التطبيق 1: تمارين بسيطة (السلسلة رقم 2)

اكتب برنامج في لغة بايثون :

- يطلب من المستخدم إدخال اسمه ثم يقوم بإظهار جملة ترحيبية، مثل : "Welcome to Python, Ahmed ".
- يطلب من المستخدم إدخال عدد طبيعي N ثم يقوم بإظهار الأعداد من 1 إلى N وفي الأخير يظهر مجموع هذه الأعداد (من 1 إلى N).
- يطلب من المستخدم إدخال عدد صحيح N ثم يقوم بالبحث عنه إن كان موجودا أم لا داخل قائمة غير مرتبة من الأعداد (القائمة تدخل مباشرة في البرنامج).
- لقراءة عناصر جدول (قائمة من الأسماء)، ثم يقوم بترتيبها وإظهارها حسب الترتيب الأبجدي.
- يطلب من المستخدم إدخال عدد طبيعي N ثم يقوم بحساب N ! وإظهار النتيجة.

- -

التطبيق 2: لعبة "حجر-ورقة-مقص".

1. اكتب برنامجا في لغة بايثون لمحاكاة لعبة "حجر – ورقة – مقص" (***Rock-Paper-Scissors***) . حيث يمكن هذا البرنامج المستخدم من لعب هذه اللعبة ضد الكمبيوتر.

   المبدأ:
   - يقوم اللاعب أولا بإدخال اختياره من بين الاختيارات الثلاثة ("حجر"،"ورقة"،"مقص") ("Rock"،"Paper"،"Scissors")
   - بعدها يقوم الكمبيوتر باختيار عشوائي من بين نفس الاختيارات (استعمال .
   - يقوم بعدها بحساب وإعلان النتيجة، حيث "الحجر يغلب المقص"، "المقص يغلب الورقة" و"الورقة تغلب الحجر".
2. حسن في هذا البرنامج بحيث يحسب النتيجة النهائية بعد 10 محاولات.

ملاحظة: يجب جلب المكتبة "random" إلى البرنامج عن طريق التعليمة "import random" من أجل استعمال الاختيارات العشوائية. ثم كتابة التعليمة "x = random.choice(list)" للسماح للكمبيوتر بالاختيار من بين الاختيارات الموجودة في list.

- -

التطبيق 3: حل المعادلات من الدرجة 1 ومن الدرجة 2.

1. اكتب خوارزم لحل معادلة من الدرجة الأولى (Ax+B=0) في ***R***، ثم حوله إلى برنامج في لغة بايثون.
2. اكتب خوارزم لحل معادلة من الدرجة الثانية في ***R***. ادرس مختلف الحالات الاستثنائية (***Edge cases***). ترجمه إلى برنامج في لغة بايثون.
3. حسن في هذا البرنامج بحيث يسمح للمستخدم بحل عدة معادلات (بعد حل كل معادلة يطلب من المستخدم إن كان يريد الاستمرار أم يريد أن ينهي العمل).

- -

التطبيق 4: حساب المعدل

اكتب برنامجا في لغة بايثون لحساب معدلك الفصلي مع الأخذ بعين الاعتبار لأسماء المواد ومعاملاتهم، حيث:

- يقوم أولا بقراءة اسم كل مادة ومعاملها ويحتفظ بهذه المعلومات في جداول (قوائم).

مثال:

Modules	"Maths"	"Informatics"	"Arabic"	"Physics"	"Chemistry"	"English"
Weights	2	2	3	2	1	1

- بعد الانتهاء من إدخال أسماء المواد ومعاملاتهم يطلب من المستخدم إدخال علامة كل مادة على حده، ومن ثم يقوم بحساب المعدل وإظهاره.
- حسب قيمة المعدل المحصل عليه يظهر النتيجة "ناجح" أو "راسب" (استعمل إجراء "دالة بدون إرجاع").

- -

التطبيق 5: الكلمة-المتناظرة.

1. اكتب خوارزم للتأكد إن كانت كلمة متناظرة أم لا (***Palindrome word***). الكلمة المتناظرة هي التي تساوي الكلمة المحصل عليها عند عكس ترتيب حروفها مثل "***radar***" أو "***level***".
2. اكتب برنامج في لغة بايثون يطلب من المستخدم إدخال جملة متكونة من عدة كلمات مفصولة بمسافات بينها، ويقوم بعدها البرنامج بالبحث عن الكلمات المتناظرة وإظهارها إن كانت موجودة، مع إظهار العدد الكلي للكلمات المكونة لهذه الكلمات المتناظرة. استعمل دالة من أجل التعرف على كلمة إن كانت متناظرة أم لا.

- -