شكراً لتحميلك هذا الملف من موقع المناهج البحرينية

دليل المعلم للمرحلة الثانوية البرمجة بلغة python

تاريخ نشر الملف على موقع المناهج: 07-03-2024 07:21:39

التواصل الاجتماعي بحسب الصف الأول الثانوي

روابط مواد الصف الأول الثانوي على تلغرام

التربية الاسلامية	اللغة العربية	اللغة الانجليزية	الرياضيات

المزيد من الملفات بحسب الصف الأول الثانوي والمادة علوم وتقانة في الفصل الثاني

1	مذكرة العلوم والتقانة
2	نموذج للتدريب في أنواع البيانات و الدوال المضمنة
3	شرح درس أنواع البيانات والثوابت والمتغيرات والعوامل الحسابية
4	ملخص الدرس الأوّل الأنظمة العددية
5	مذكرة التقانة

KINGDOM OF BAHRAIN
Ministry of Education

مملكة البحرين
وزارة التربية والتعليم

دليل المعلّم

المرحلة الثانوية

المسار العام/الفني والمهني ف

البرمجة بلغة Python

تقن 106 / تقن804/805/806

وحدة مناهج تقنية المعلومات والاتصال

2024 / 2023

ت	العنوان	الصفحة
	الجانب النظري	
1	الرؤية والرسالة	3
2	مقدّمة الدليل	3
3	أهميّة الدليل	3
4	مقدّمة الكتاب	4
5	تصميم الكتاب المدرسي	5
	الجانب التطبيقي	
7	النتائج التعليمية	5
8	تقييم مدى التقدّم في تحقيق النتائج التعليمية	5
9	كيفيّة استخدام الكتاب المدرسي	7
10	واجهة الكتاب الالكتروني	7
11	الشاشات الافتتاحيّة للكتاب المدرسي	8
12	فهرس الكتاب الالكتروني	9
13	الخطة الأسبوعية	11
14	الأنظمة العدديّة	12
15	التخطيط لحلّ المسائل البرمجيّة	14
16	التعامل مع البيانات	20
17	جمل الادخال والاخراج	21
18	الدوال المضمّنة	27
19	الجمل الخوارزميّة	29
20	الجمل الشرطية	29
21	الجمل التكرارية	37
22	بناء الدوال	47

الرؤية والرسالة

الرؤية

منهج يُعزز الانتماء ويرسخ قيم المواطنة، ويسعى إلى ردم الفجوة في التعلم من خلال مواكبة التطوّر التكنولوجي الوطني والعالمي وتحقيق متطلّبات سوق العمل ليساهم في تحقيق أهداف التنمية المستدامة.

الرسالة

اكساب خريج المرحلة الثانوية مهارات القرن 21 من خلال مناهج دراسية مطوّرة قائمة على استراتيجية المشاريع مما يعزّز لدى الطالب القدرة على التخطيط لحلّ المشكلات وتقييم الذات والتفكير الناقد إلى جانب تحمّل المسؤولية كقائد لإدارة مجموعة العمل أو كعضو ضمن فريق يكتسب من خلالها مجموعة من القيم كالتعايش وقبول الآخر والتعاون مع الزملاء.

مقدّمة الدليل

عزيزي المعلّم:

إن نجاح المنهج يتوقف على أدائك مع طلابك داخل الفصل وخارجه، ولما كان المنهج معدّا بأسلوب جديد يعتمد على فلسفة التعلم عبر المشاريع ويهدف أساسا إلى تنمية المفاهيم والمهارات والقيم والاتجاهات الإيجابية لدى الطلاب لإعدادهم كمواطنين صالحين قادرين على النهوض بالمجتمع، فأنت بحاجة إلى دليل يوضح لك فلسفة المنهج وأهدافه ويرشدك إلى كيفية تنفيذه تنفيذا جيدا.

أهميّة الدليل

عزيزي المعلم تتمثل أهمية الدليل في الآتي:

1. يرشدك الى كيفية استعمال وسائل ومصادر تعلم متعددة.
2. يقدم لك مقترحات تتعلق بتوظيف المعلم للتكنولوجيا
3. يوجهك إلى تطبيق استراتيجيات تدريسية متنوعة.
4. يقدم لك إجراءات واضحة لكيفية تنفيذ أنشطة التعلم.
5. يوضح كيفية معالجة القضايا المتضمنة في أثناء الدرس.
6. يوجهك إلى استعمال أساليب تقويم متنوعة (معرفية – مهارية – وجدانية)
7. يوجهك إلى كيفية الربط والتكامل بين فروع المادة.
8. يوجهك إلى كيفية الربط والتكامل بين المادة والمواد الأخرى.
9. يرشدك إلى كيفية تحديد الأخطاء الشائعة، وعلاجها.
10. يتضمن نماذج استرشادية لأساليب التقويم.
11. يتضمن قائمة بالمصطلحات التربوية والأكاديمية التي تحتاج إليها في تدريسك للمادة.
12. يتضمن قائمة بالمراجع، وبعض المواقع الإلكترونية؛ للاستفادة منها في تدريس المادة.

عزيزي المعلم، يهدف هذا الدليل إلى مساعدتك في:

1. تدريس المنهج على نحو فعّال يحقق الأهداف المتوخاة في إطار التعلم عبر المشاريع.
2. تنظيم وإدارة الفصل المتعدّد المستويات على نحو فاعل.
3. تشكيل بيئة تعليمية قائمة على التعلم عبر المشاريع الذي يجعل المتعلّم محور العملية التعليمية ومشاركا نشطا في عملية التعلّم.
4. إثارة دافعية المتعلمين نحو التعلّم وخلق جو تعاوني بينهم.
5. تأكيد الجوانب المهارية والوجدانية عند تدريس المنهج.
6. استعمال أساليب تقويم متنوعة ومتعددة المستويات

مقدّمة الكتاب

تشغل التكنولوجيا جزءا هاما من الحياة اليوميّة للشباب في ربوع العالم، وهي مجموعة الأدوات التي أصبح يعتمدها الجميع لتسهيل وإنجاز المهام المطلوبة منهم والتواصل مع العالم والتسلية، وحيث أضحت التكنولوجيا الرقمية ضرورة من ضرورات الحياة، فإنّ تعلّمها أيضا أضحى لا مفرّ منه ليستطيع الطالب والمعلّم والموظّف وغيرهم إنجاز أعمالهم وإدارة شؤونهم في مختلف المجالات. كما تسهم في تحسين فرص العمل وتوظيفهم مستقبلا.

ويهدف كتاب تقنية المعلومات والاتصال تقن105 والذي يتناول موضوع "**البرمجة بلغة Python**" إلى ما يلي:

- التعرّف على الأنظمة العدديّة (لنظام العشري والنظام الثنائي).
- يحوّل القيم العدديّة من انظام الثنائي إلى النظام العشري والعكس.
- يتعرّف أهميّة الكود المزيّف في التخطيط لحلّ المشكلات وكلماته المفتاحيّة.
- يتعرّف أشكال الخرائط التدفّقيّة ويوظّفها لحلّ المسائل البرمجيّة.
- يخطّط لحلّ المسائل البرمجيّة عن طريق الكود المزيّف Pseudocode والخرائط التدفّقيّة Flowcharts.
- يتعرّف الجمل البنائيّة syntax للغة Python.
- يحوّل الكود المزيّف/ الخريطة التدفقيّة إلى حلّ برمجيّ بلغة Python.
- يتعرّف المتغيّرات ومعظم أنواع البيانات في لغة Python.
- يتعرّف الكلمات المفتاحيّة للأوامر الادخال والإخراج في لغة Python.
- يتعرّف الجمل الخوارزمية، الشرطية والتكرارية في لغة Python ويوظّفها لحلّ مسائل برمجيّة.
- يتعرّف ماهية الدوال المضمّنة في لغة البرمجة PYTHON ووظيفتها.
- يوظّف الدوال المضمّنة المناسبة لحلّ مسألة برمجيّة.
- يتعلّم بناء الدوال الخاصة ويوظّفها ضمن البرنامج الرئيسي لحلّ مشكل برمجي.
- يتعقّب الأخطاء ويصحّحها.

تصميم الكتاب المدرسي:

يعتمد الكتاب في مختلف محاوره على إبراز مفاتيح البرمجة بلغة بايثون، معتمدا قبل ذلك على التخطيط من خلال الكود المزيّف Pseudocode والخرائط الذهنيّة Flowchart. ويلي كلّ درس من الدروس مجموعة من الأنشطة التكوينيّة يقوم الطالب بمحاولة إيجاد الحلّ منفردا أو بالتعاون مع زملائه في المجموعة قصد التأكّد من إتقان الطالب من المهارة / الكفاية المطلوبة.

يقوم هذا التصميم على:

- التعاون
- الاعتماد على الذات
- الاكتشاف
- الابداع
- اتّخاذ القرار

النتائج التعلّمية

1- القدرة على التخطيط لحلّ المشكلة بوسائل مختلفة قبل التحوّل إلى لغة البرمجة.
2- القدرة على التعامل مع الجمل البنائية للغة البرمجة Python.
3- يعزّز ثقة الطالب بقدراته.
4- يشجّع الطالب على الابداع والابتكار.

تقييم مدى التقدّم في تحقيق النتائج التعلميّة

- أهمّ تقييم لهذا الجانب هو تقييم تقدّم الطالب في تطبيق وحل الأنشطة التكوينية المتوفّرة ضمن الكتاب المدرسي وهو الذي يطبّق فيه الطالب ما اكتسبه من مهارات خلال حلّ الأمثلة المصاحبة لكلّ درس.
- كما يتمّ تقييم تقدّم الطالب من خلال اختبارات قصيرة (تطبيقية / نظرية) نهاية كلّ وحدة. حيث يمكن أن تقسّم الوحدات كالتالي:
  - الأنظمة العدديّة.
  - التخطيط للحلّ من خلال Flowchart /Pseudocode
  - المتغيّرات/أنواع البيانات جمل الإدخال والإخراج
  - الجملة الشرطية
  - الجمل التكراريّة
  - الدوال المضمّنة
  - بناء الدوال الخاصة
  - تعقّب الأخطاء وتصحيحها

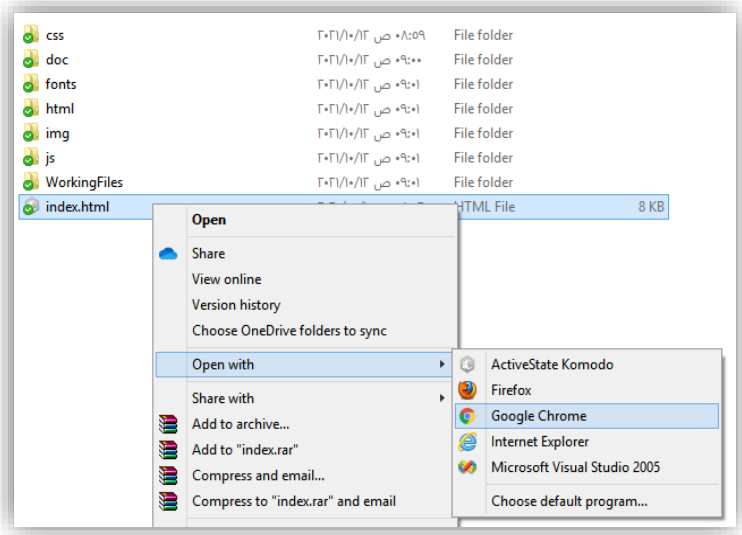
الجانب التطبيقي

كيفيّة استخدام الكتاب المدرسي

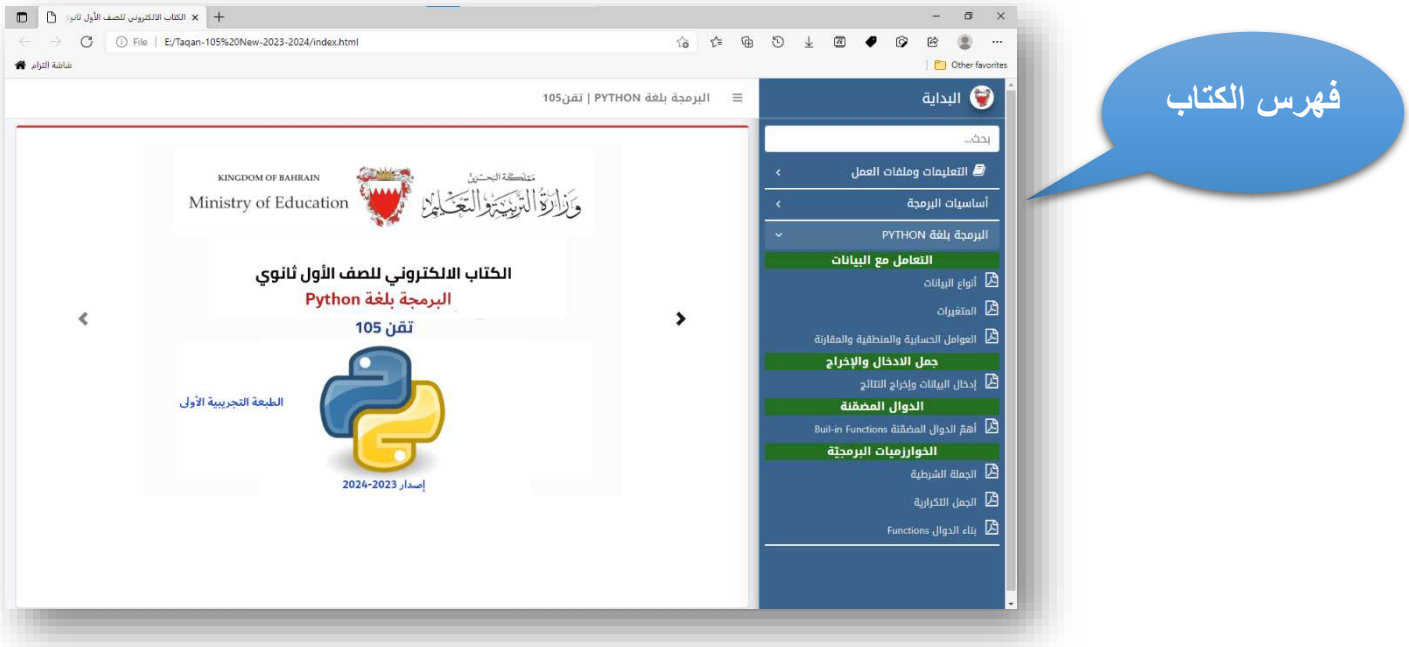
يتمّ تنزيل الكتاب من موقع وزارة التربية والتعليم: https://www.moe.gov.bh/moeecontent.aspx

صمّم هذا الكتاب على شكل موقع ويب، وهذه الطريقة تضمن العديد من الإيجابيات وأهمّها:

1- سهولة استعراض الطالب للكتاب المدرسي الالكتروني حيث أن الجميع قادر على تصفّح مواقع الانترنت في وقتنا الحاضر.
2- سهولة تطوير المحتوى العلمي للكتاب كلّما دعت الحاجة لذلك، حيث لا يتطلّب الأمر برمجيّات خاصة أو مدفوعة الأجر.
3- إمكانيّة الجمع بين الكتاب المدرسي والدروس النموذجيّة والإثراءات والفيديوهات في مكان واحد وعدم تشتيت ذهن الطالب في الانتقال من موقع إلى آخر للتمكّن من الحصول على الموارد التعليمية والإثرائية.
4- لفتح موقع الكتاب الالكتروني:
   أ. فتح مجلّد الكتاب
   ب. بزر الفأرة الأيمن، اضغط على ملف index
   ج. اختر الأمر open with
   د. اختر المتصفح google chrome

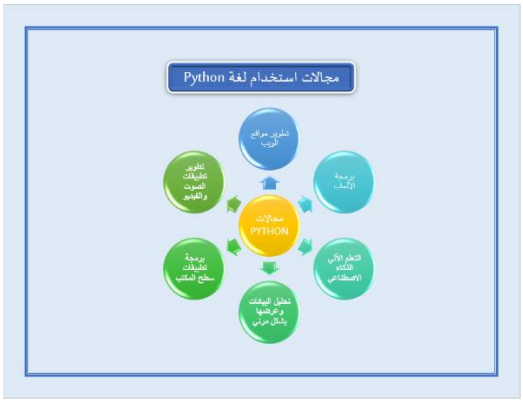


واجهة الكتاب الالكتروني



الشاشات الافتتاحيّة للكتاب المدرسي

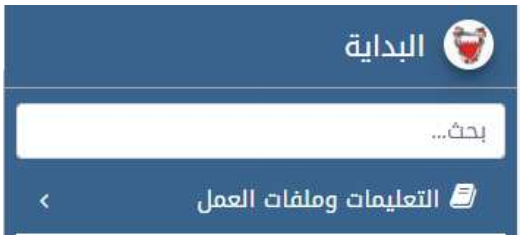
الغلاف	
صفحة حقوق الطبع وسنة الطبعة	
صورة جلالة الملك حمد بن عيسى آل خليفة حفظه الله ورعاه	
مقدّمة الكتاب	

مقدّمة الكتاب	 

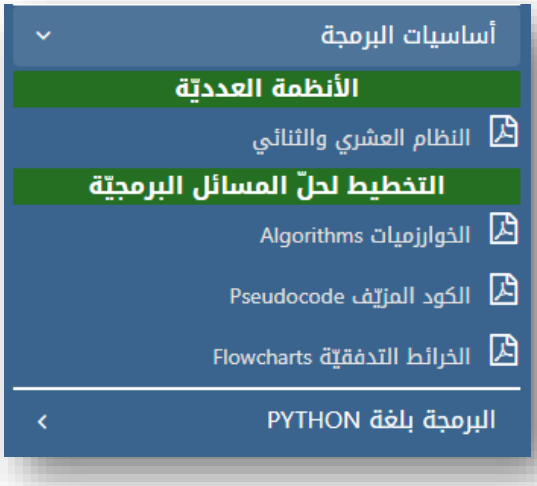
الفهرس

يحتوي الفهرس على 3 محاور أساسيّة للمنهج الدراسي بالإضافة إلى رابط لعرض التعليمات الخاصة بالمنهج الدراسي ورابط لتنزيل الملفات التطبيقية.

1. **التعليمات وملفات العمل:** تضمّ هذه القائمة رابطا إلى ملف التعليمات العامة حيث يعرض أهم محاور الكتاب والبرمجيّات اللازمة لتطبيق المنهج الدراسي بالإضافة إلى هيكل الملفات الخاصة بالدروس وبعض التوصيات. كما يتوفّر رابط "**تنزيل ملفات العمل**" لتنزيل الملفات المتعلّقة بالمشروع التعليمي للطالب والتي سيستخدمها لتطبيق المهارات التي سيتعلّمها.



2. أساسيّات البرمجة



يحتوي على الدروس المتعلّقة بأساسيّات البرمجة بغضّ النظر عن لغة البرمجة المستخدمة، كما تضمّنت القائمة الأنظمة العدديّة لعلاقتها الوثيقة بجهاز الحاسوب ولغات البرمجة بصفة عامّة.

3. البرمجة بلغة Python

يحتوي على الدروس المتعلّقة بلغة البرمجة Python وطريقة تصميم الجمل البنائية لهذه اللغة، والأوامر البرمجيّة الأساسيّة والجمل الخوارزميّة المختلفة الأساسيّة لحلّ المسائل البرمجيّة التي تحقّق الكفايات والمكتسبات المطلوبة.

الخطة الأسبوعية لتنفيذ مقرّر تقن??? خلال فصل دراسي

المادة/ المقرر الدراسي: تقنية المعلومات والاتصال / البرمجة بلغة Python (تقن???)

الرقم	الصف الدراسي	عنوان الدرس	رقم الأسبوع	الملاحظات
1	الصف الأول ثانوي	التعليمات وملفات العمل	1	
2		الأنظمة العدديّة	2	
3		الخوارزميّات	3	
			4	
4		الكود المزيّف	5	
			6	
5		الخرائط التدفّقيّة	7	
6		أنواع البيانات المتغيّرات العوامل الحسابيّة والمنطقية والمقارنة	7	
		جمل الإدخال والإخراج	8	
7		أهمّ الدوال المضمّنة Built-in functions	8	تتبّع الأخطاء وإصلاحها
		الجملة الشرطية	9	
8		الجمل التكرارية	9-10	
		بناء الدوال	11-12	
المهمّة		الانتهاء من إنجاز المهمّة والتقييم	13	
			14	

الأنظمة العدديّة

الدرس1: النظام العشري والنظام الثنائي

يعرض هذا الفصل في الدرس الأول منه: "الأنظمة العدديّة"، تعريفا وجيزا للأنظمة العدديّة ويشرح بالخصوص النظام العددي الثنائي والنظام العددي العشري. إنّ الغاية من التطرّق إلى هذا الموضوع هو تبيان العلاقة بين النظام الثنائي ولغة جهاز الحاسب الآلي بدرجة أولى حتى يفهم الطالب ما يدور داخله وكيف يتعامل مع المعطيات والبيانات التي يعالجها. كما يجب عليك عزيزي المعلّم أن تبيّن بكلّ وضوح ماهية الإشارات الكهربائية داخل الجهاز وسبب ترميزها بالرقمين الصفر والواحد، والذي سيقودك بالتالي إلى التطرّق لوحدات البت والبايت.

بما ان الانسان والعالم يتعامل حاليا مع أرقام النظام العشري، وانّ مستخدم الحاسوب يقوم بإدخال قيم وأعداد من النظام العشري فكيف للحاسوب أن يفهما؟: هنا تبيّن أنّ الحاسوب يقوم بتحويل جميع البيانات المدخلة إلى النظام الثنائي لكي يستطيع الحاسوب فهمها ومن ثمّ يعالجها ويحوّلها مرّة أخرى على الهيئة التي يطلبها المستخدم، إمّا معروضة على الشاشة أو مطبوعة على ورق. وإن كانت النتيجة المطلوبة من نوع رقمي فهو يحوّلها مرّة أخرى إلى النظام العشري قبل عرضها. لذلك ستقوم عزيزي المعلّم بتعليم الطالب آلية التحويل من النظام العشري إلى الثنائي والعكس.

حلّ أنشطة الدرس

النشاط1:

مطلوب تحويل العدد العشري 132_{10} إلى عدد ثنائي.\

سأطبّق طريقة القسمة المتكرّرة ضمن الجدول:

القاسم	132	66	33	16	8	4	2
المقسوم عليه	2	2	2	2	2	2	2
النتيجة	66	33	16	8	4	2	1
الباقي	0	0	1	0	0	0	0

$$(132)_{10} = (10000100)_2 = 1\times2^7 + 1\times2^2 = 128 + 4 = 132$$

النشاط 2:

1- مطلوب تحويل العدد $(1101001)_2$ إلى النظام العشري

العدد الثنائي	1	0	0	1	0	1	1
موقع الرقم ضمن العدد الثنائي	0	1	2	3	4	5	6
النتيجة	1×2^0	0×2^1	0×2^2	1×2^3	0×2^4	1×2^5	1×2^6
	1	0	0	8	0	32	64
	1 + 0 + 0 + 8 + 0 + 32 + 64						
	$(105)_{10}$ =						

2- مطلوب ترتيب الأعداد الآتية من الأكبر إلى الأصغر.

$(1101)_2$ | $(23)_{10}$ | $(10011)_2$ | $(44)_{10}$ | $(11110)_2$ | $(33)_{10}$

- سأقوم بتحويل جميع الأعداد إلى النظام العشري أولا:

$(1101)_2 = (13)_{10}$

$(10011)_2 = (19)_{10}$

$(11110)_2 = (30)_{10}$

فيكون الترتيب كالآتي:

$(44)_{10} > (33)_{10} > (11110)_2 > (23)_{10} > (10011)_2 > (1101)_2$

التخطيط لحلّ المسائل الخوارزميّة

الدرس2: الخوارزميات Algorithm

عزيزي المعلّم، تحاول في هذا الدرس أن تطرح على الطالب عادات حياتيّة يقوم بها ثمّ تطلب منه القيام بكتابة مراحل قيامه بهذه الأعمال كأن يحضّر وجبة معيّنة أو أن يشغّل جهاز ما أو يكتب خطوات معيّنة لحلّ مشكل استثنائي (على سبيل المثال: تشغيل سيارة تعطّلت بطاريّتها / فتح باب مغلق بدون مفتاح / ....).

حلّ أنشطة الدرس:

نشاط 1:

1	2	4	5	3	6	7

الملاحظة: المراحل 3 و4 و5 يمكن التغيير بينها دون التأثير على انطلاق السيارة بالشكل الصحيح.

2	1	3	4	5	7	6

ألاحظ أنّه من المستحيل:

- تقديم الخطوة 2 على الخطوة1: أن أركب السيارة قبل فتحها.
- تقديم الخطوة 7 على الخطوة 6: أنطلق بالسيارة قبل تعشيق مغيّر السرعات.

نشاط 2: مراحل طباعة الصفحة الثانية من مستند ما.

1- أفتح المستند
2- أنتقل إلى الصفحة الثانية من المستند
3- أفتح نافذة أوامر الطباعة
4- أضغط على الخيار: الصفحة الحاليّة
5- أضغط زر الطباعة

نشاط 3:

مثال: إجراء عمليّة قسمة

1- التحليل:
   أ. تحديد المعطيات: إدخال عددين صحيحين
   ب. تحديد القاسم والمقسوم عليه
   ج. التأكّد من أنّ المقسوم عليه لا يساوي صفرا
   د. تحديد العمليّة الرياضيّة لإيجاد النتيجة
2- التصميم
   أ. إدخال القاسم
   ب. كرّر إدخال قيمة المقسوم طالما كانت تساوي صفرا.
   ج. احسب النتيجة = القاسم / المقسوم عليه
   د. عرض النتيجة

الدرس3: الكود المزيّف Pseudocode

عزيزي المعلّم، في هذا الدرس سيتعلّم الطالب طريقة حلّ المسائل البرمجيّة عن طريق الكود المزيّف، ومن هنا يبدأ التركيز على اتقان الطالب لعمليّة التخطيط لحلّ المشكلات. وتعتبر هذه الطريقة حجر الأساس لتكوين المبرمجين حيث أنّ المبرمج الماهر هو من يجد الحلّ الأمثل للمشكل البرمجيّ وليس من يستخدم لغة البرمجة الأفضل. إنّ المبرمج المتهر من يخطّط جيّدا لإيجاد الحلّ وأمّا عن لغة البرمجة فيمكن تعلّم أيّ لغة كانت من خلال مواقع الويب.

حلّ النشاط 1:

variable/Constant	Correct	Incorrect	If incorrect:Why?
12Amount		☑	يبدأ برقم
T3b2	☑		
Integer		☑	كلمة محجوزة
Size1	☑		
Am%nt		☑	يحتوي رمز لا يستخدم

حلّ النشاط 2:

Variables	Data Types
power	integer
speed	Reel/float/decimal
tatement	string
Sale	date

power ← 120

speed ← 3.22

statement ← "sky"

Sale ← #31-12-2022#

حلّ النشاط 3:

أكتب كودا مزيّفا Pseudocode يجعل الحاسوب يستقبل قيمتان صحيحتان ويحفظ نتيجة مجموعهما في متغيّر باسم SumVar ويعرض النتيجة على الشاشة.

```
INPUT ("Enter first integer:", in1)
INPUT ("Enter second integer:", in2)
SumVar ← in1 + in2
DISPLAY ("Result=", SumVar)
```

حلّ النشاط 4:

```
VAR age AS INTEGER
VAR NewPrice AS FLOAT
INPUT ("PLZ ENTER YOUR AGE:", age)
IF age <= 18 THEN
        NewPrice = 35 – (35*0.25)
ENDIF
DISPLAY ("Discounted price = ", NewPrice)
```

حلّ النشاط 5:

```
VAR a, b as INTEGER

a=1, b=1

WHILE (a * b) <> 1000

        INPUT("Plz re-enter a:", a)

        INPUT("Plz re-enter b:", b)

ENDWHILE
```

حلّ النشاط 6:

```
VAR i , multi as INTEGER

multi = 8

FOR i=2 TO 6 DO

        multi = multi × i

        DISPLAY (multi)

ENDFOR
```

الدرس4: الخرائط التدفقيّة Flowcharts

عزيزي المعلّم، الخرائط التدفقيّة هو نوع آخر من أنواع التصميم للحلّ البرمجي وهو حلّ رسوميّ يجعل أكثر وضوحا وتنظيما من الحلّ الخوارزمي أو الحلّ عن طريق الكود المزيّف.

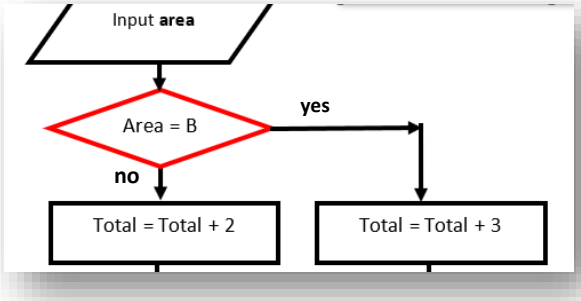
تعتمد الخرائط التدفقيّة على أشكال هندسيّة محدّدة بحيث يرمز كلّ شكل إلى إجراء محدّد (صفحة 2 من درس الخرائط التدفقيّة).

حلّ أنشطة الدرس

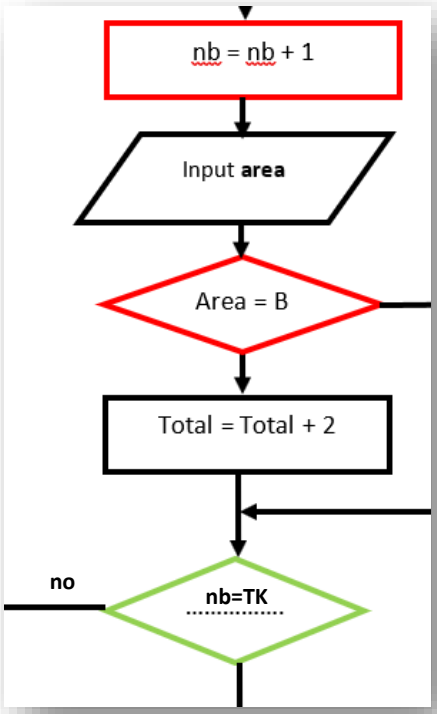
حلّ النشاط 1:

1- الخريطة السابقة صمّم الحلّ بحيث يبدأ احتساب عدد التذاكر المباعة (nb=nb+1) بعد عمليّة الشراء، بينما في الخريطة الحاليّة يبدأ احتساب عدد التذاكر المباعة قبل عمليّة الشراء.

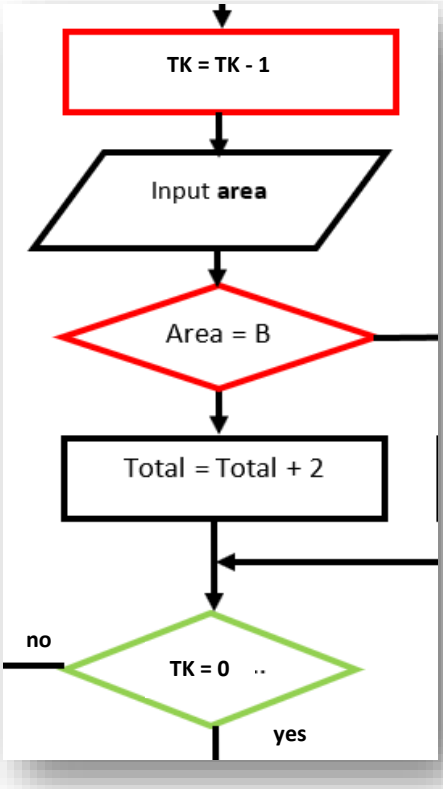
2-



-3


nb = nb + 1
Input area
Area = B
Total = Total + 2
no
nb=TK

-4


TK = TK - 1
Input area
Area = B
Total = Total + 2
no
TK = 0
yes

حلّ النشاط 2:

حساب رسوم الفحص الفني حسب عمر السيارة:

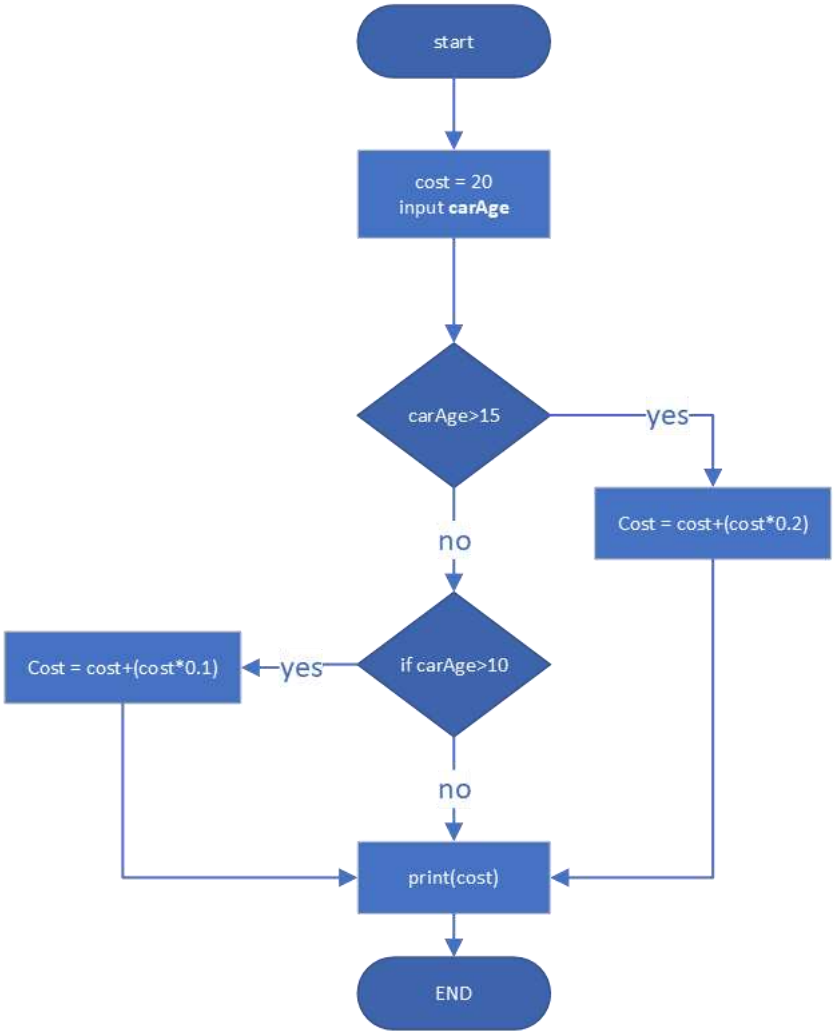
يتمّ إدخال عمر السيارة

إذا كان عمر السيارة أكبر من 15 سنة ← رسوم الفحص = 20+(20*20%)

إذا كان عمر السيارة أكبر من 10 سنوات ← رسوم الفحص = 20+(20*10%)

إذا لم تتحقّق الشروط أعلاه (أي أن عمر السيارة لم يتجاوز العشر سنوات) ← تكون رسوم الفحص = 20

الخريطة التدفقية:



التعامل مع البيانات

الدرس 5: أنواع البيانات

عزيزي المعلّم، يعتبر التعرّف على أنواع البيانات أو تعدادها من الكفايات السابقة التي اكتسبها الطالب في المرحلة الأولى من خلال تطرّقه إلى برمجيّة سكراتش وإلى برمجيّة Php ولكن بدرجة بسيطة.

يمكن أن تطلب من الطلبة كعمل تعاوني أن تقوم كلّ مجموعة بكتابة قائمة من أنواع البيانات وعرضها فيما بينهم. أذكر لهم كبداية البيانات الرقمية كمثال.

يمكن الاكتفاء بالأنواع الآتية: الرقمي الصحيح – الرقمي العشري – النصي – التاريخ – حرف – المنطقي (integer – float or decimal – string – date – char - Boolean)

كما يمكنك عزيزي المعلّم التعمّق قليلا في الدرس وذكر أسباب ضرورة تصنيف البيانات في البرمجة كأن أحتاج مثلا إلى دمج أرقام مع بعضها: "5" + "2" = "25" أو لتجنّب الوقوع في بعض الأخطاء التي تودّي إلى نتائج غير المطلوبة.

الدرس 6: المتغيّرات

عزيزي المعلّم، يتعامل الطالب كثيرا في مادّة الرياضيات مع المتغيّرات، وخاصة في المعادلات الرياضيّة. انتق أية معادلة بسيطة: a2 + b2 ثمّ استبدل a وb بقيم بسيطة وبيّن بكلّ بساطة أنّ a وb ليست لهما قيم ثابتة لذلك نسمّيهم متغيّرات. كما لا يفوتك أنّ الطلاب قد تطرّقوا سابقا إلى مفهوم المتغيّرات ضمن صفوف المرحلة الابتدائية والإعدادية.

أمّا بالنسبة للثوابت constants فإنّ لغة البرمجة بايثون لا تعرّف الثوابت ولكن يمكن الاعتماد على طرق أخرى لتثبيت قيمة متغيّر ما.

أمّا بالنسبة لتعاملنا مع الثوابت في هذا الكتاب المدرسي وبناء على عديد المراجع الدولية فإنّنا سنعتمد على المتغيّرات المسمّاة بحروف انجليزية كبيرة كثوابت بحيث لا تقوم بتغيير قيمها أثناء تنفيذ البرنامج. هذه الطريقة هي طريقة متّفق عليها وليست قاعدة من قواعد لغة البايثون.

مثال: MY_NUMBER أو MY_NAME

حلّ النشاط 1:

حدّد في الجدول الآتي، ما إذا كانت تسميات المتغيّرات/الثوابت صحيحة أو لا! ولماذا؟

variable/Constant	Correct	Incorrect	If incorrect:Why?
12Amount		☑	يبدأ برقم

T3b2	☑		
Integer		☑	كلمة محجوزة
Size1	☑		
Am%nt		☑	يحتوي رمز لا يستخدم

الدرس 7: العوامل الحسابية والمنطقية والمقارنة

حلّ النشاط 1:

جِدْ نتيجة الجُمل المنطقيّة الآتية، مع العلم أنّ (x=4 و y=2 و z=6):

x > y or z>y : True

not(x=y) and z<y : false

y <> z and not(x=y or x > (z-y)) :True

حلّ النشاط 2:

جِدْ نتيجة الجُمل الحسابيّة الآتية في لغة Python:

13/2 : 6.5

13 % 2 : 1

13 // 2 : 6

10 ** 2 : 100

جمل الادخال والإخراج / Input & Output statements

الدرس 8: إدخال البيانات وإخراج النتائج

عزيزي المعلم، بدءا من هذا الدرس يجب عليك التركيز على تطبيق الكود المزيف Pseudocode والخرائط التدفقية قبل التطبيق على لغة البرمجة Python.

استخدم جملة الادخال **INPUT** لإدخال بعض القيم لمتغيرات مختلفة واعرضها واعرض نوعها على الشاشة.

مثال هام جدا:

1. أدخل عددا صحيحا في المتغير n بلغة Python:
2. لنفترض أنك أدخلت العد 5

```
n=input("Enter an integer:")
Enter an integer:5
```

3. اعرض قيمة المتغيّر n ← المُخرج هو الرقم 5

```
print(n)
5
```

4. سأعرض الان نوع المتغيّر n ← المُخرج هو النوع str

```
>>> print(type(n))
    <class 'str'>
```

ألاحظ أنّ نوع المتغيّر n حسب لغة Python هو str يعني نصي. لذلك إذا أردت استخدام المتغيّر n كقيمة رقمية يجب استخدامه ضمن الدالة int() كالآتي مثلا:

```
print(int(n) + 5)
10
```

وألاحظ الخطأ البرمجي في حال استخدمت المتغيّر n بدون الدالة int():

```
print(n+5)
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    print(n+5)
TypeError: can only concatenate str (not "int") to str
```

بحيث يهيّأ للبرنامج أنّني أردت حساب مجموع نص مع رقم.

بينما إذا دمجت نصا مع المتغيّر n ←

```
print("The number that was entred is:" + n)
```

تكون النتيجة كالآتي: ←

```
The number that was entred is:5
```

حلّ النشاط 1:

ضمن الملف السابق، أكمل إدخال بقية الأوامر البرمجية بلغة بايثون لإدخال الاسم الأول ضمن المتغيّر FirstName .

```
FirstName=input("Enter your first name:")
```

أمّا بخصوص جملة الإخراج OUTPUT في الكود المزيّف أو print في لغة بايثون فهي لإخراج نتيجة معالجة بيانات محدّدة. ويجب إبراز مختلف حالات الإخراج ضمن الدالة print كالاتي:

- أن يكون المُخرج متغيّرا واحدا مثل: print(x)

- أن يكون المُخرج قيمة، مثل:print("Welcome Bahrain") || print(10)
- أن يكون المُخرج عبارة عن معادلة حسابيّة: print(5**2+10)
- أن يكون المخرج نتيجة دالة، مثل: print(int(degree))

حلّ النشاط 2:

إظهار محتوى المتغيّر degree.

```
print(degree)
```

حلّ النشاط 3:

Variables	Data Types
power	integer
speed	Reel/float/decimal
statement	string
Sale	date

1. أسند كلّ قيمة من القيم الموجودة في الدائرة إلى المتغيّر المناسب من الجدول.

power = 120 / speed = 3.22 / statement = sky / Sale = #31-12-2022#

حلّ النشاط 4:

I. أكتب كودا مزيّفا **Pseudocode** ينفّذ الآتي:

1. يعيّن متغيّران صحيحان
2. يجعل الحاسوب يستقبل قيمتان صحيحتان يحفظان في هذين المتغيّرين.
3. يحفظ نتيجة مجموعهما في متغيّر باسم **SumVar** ويعرض النتيجة على الشاشة.
4. يحفظ باقي قسمة **SumVar** على العدد 3 في المتغيّر **Multi3**.

```
start

DECLARE n1 as INTEGER

DECLARE n2 as INTEGER
```

```
INPUT ("Enter number 1:", n1)

INPUT ("Enter number 2:", n2)

SumVar = n1 + n2

OUTPUT("sum=", SumVar)

Multi3 = SumVar MOD 3

END
```

II. نفّذ الآتي:

1. افتح ملف **Python** جديد واحفظه باسم **InputOutput.py**
2. ضمن الملف الجديد، حوّل الكود المزيّف إلى لغة **Python**.
3. نظريّا: هل العدد **SumVar** من مكرّرات الرقم 3؟

```python
n1=input("Enter number1:")
n2=input("Enter number2:")
SumVar=int(n1)+int(n2)
print("Sum=",SumVar)
Multi3=SumVar % 3
```

حلّ النشاط 5:

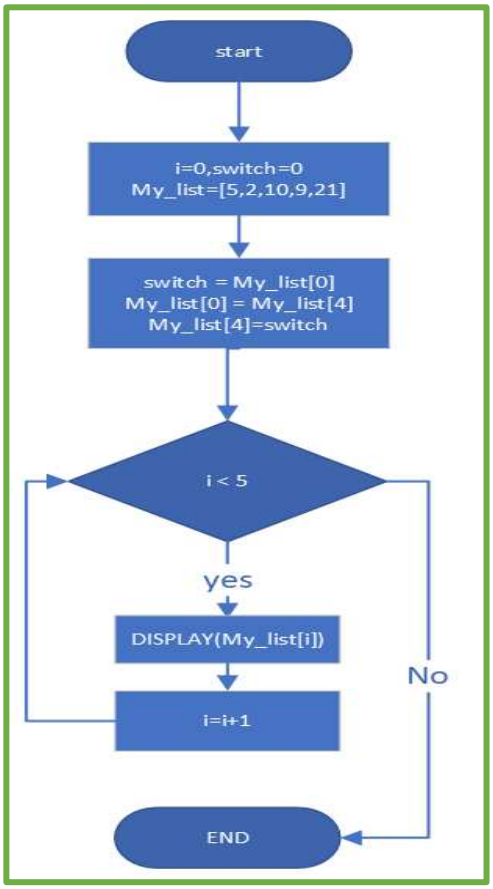
أكتب كودا مزيّفا **Pseudocode** ينفّذ الآتي:

1. يعيّن سلسلة من 5 أعداد صحيحة.
2. يعكس قيمة العنصر الأول مع العنصر الأخير من السلسلة
3. يظهر عناصر السلسلة بعد التبديل.
4. أرسم الخريطة التدفقية للحل بالاستعانة ببرمجيّة visio.
5. حوّل الكود إلى لغة Python.

Pseudocode

```
START
DECLARE My_list AS array[5]
DECLARE switch, i as INTEGER
i=0, switch = 0
My_list = [5,2,10,9,21]
switch = My_list[0]
My_list[0] = My_list[4]
My_list[4] = switch
FOR i=0 to 4 DO
        DISPLAY My_list[i]
ENDFOR
```

Flowchart



Code Python

```
File  Edit  Format  Run  Options  Window  Help
1 My_list=[5,2,10,9,21]
2 i=0
3 switch=0
4 switch=My_list[0]
5 My_list[0]=My_list[4]
6 My_list[4]=switch
7 for i in range(5):
8     print("the ", i+1," of the list=",My_list[i])
9
```

Output:

```
========== RESTART: C:/Users/
the  1  of the list= 21
the  2  of the list= 2
the  3  of the list= 10
the  4  of the list= 9
the  5  of the list= 5
```

الدوال المضمّنة / Built-in Functions

الدرس 9: أهمّ الدّوال المضمّنة

عزيزي المعلّم بالإمكان ربط هذا الدرس مع مقرّر الرياضيات وتمثيل الدوال المضمّنة ببعض الأمثلة الرياضيّة كالآتي:

في الرياضيات		في لغة بايثون
الجذر التربيعي	$\sqrt{9}$	*import math* *math.sqrt(9)*
القيمة المطلقة	$\|x\|$	*abs(x)*

كذلك استخدمنا الدوال المضمنة في الدروس السابقة كـ: print() و input() و int()

ابدأ الدرس بشرح ماهي الدوال المضمنة في لغات البرمجة وأهميّتها في مساعدة المبرمج على تسهيل عمليّة البرمجة وحلّ المسائل.

حلّ النشاط 1:

inputs	instructions	results
x=−2	y=abs(x)	y= 2
a=[2,5,9,10,5,6,7,8,9]	b = len(a)	b= 9
z="123"	n = int(z)	n= 123
z=123	s=str(z)	s= "123"
r=range(6)		r contains(0,1,2,3,4,5)
r=range(0,6,2)		r contains (0,2,4)
n=4	f = float(n)	f=4.0
name="othman"	l=len(name)	l= 6
L=[5,6,2,0,7]	M=**min**(L)	M= 0
L=[5,6,2,0,7]	M=**max**(L)	M= 7
L=[5,6,2,0,7]	M=**sum**(L)	M= 20
L=[5,6,2,0,7]	M=sorted(L)	M= [0,2,5,6,7]
L=[5,6,2,0,7]	M=sorted(L, reverse=True)	M= [7,6,5,2,0]
L=[5,6,2,0,7]	i=L.index(0)	i = 5
L=[5,6,2,0,7]	L.remove(5)	L=[6,2,0,7]
L=[5,6,2,0,7]	L.pop(3)	L=[5,6,2,7]

r is a variable	r=random.random()	r is between 0 and 1
r is a variable	r=random.randint(15,20)	r is between 15 and 20
r is a variable	r=random.ranrange(15,20)	r is between 15 and 20 not included
r is a variable	r=random.randrange(15,25,5)	r is between 15 and 25 step 5

الجمل الخوارزميّة /

الدرس 10: الجملة الشرطيّة

عزيزي المعلّم، في هذا الدرس سوف تتطرّق إلى ثلاثة أنواع من الجملة الشرطيّة: البسيطة والكاملة والمتداخلة، وأفضل طريقة لشرحها كونك تحاول توظيفها في مسائل حياتيّة وتركّز على تحديد الشرط ونتائجه.

مثال:

1- دعهم يبحثون عن مخالفات تجاوز السرعة المسموح بها: إذا كانت في حدود 10% ثم 30% ثمّ أكثر من 30%
2- متى يكون عدد ما فرديّ.
3- متى يكون عدد ما أوّليّا.

الجملة الشرطية البسيطة

حلّ النشاط 1:

1. أكتب بلغة **Pseudocode**، برنامجا يمكّنك من إدخال المعدّل الفصلي للطالب وإظهار الرسالة "**ناجح**" إذا كان المعدّل يساوي 60 فما فوق.
2. حوّل الكود في السؤال الأول إلى كود بلغة Python.

1-

```
DECLARE deg AS DECIMAL

INPUT("Plz Enter the student degree:",deg)

IF deg >=60 THEN DISPLAY("ناجح")

ENDIF
```

2-

```
deg=input("Plz enter the student degree:")
if int(deg)>=60:
    print("ناجح")
```

```
Plz enter the student degree 70
ناجح
```

```
Plz enter the student degree 50
```

الجملة الشرطية الكاملة

حلّ النشاط 2:

1. أكتب بلغة **Pseudocode** برنامجا يمكّنك من عرض نتيجة عمليّة قسمة لعددين صحيحين ويتأكّد قبل إجراء العمليّة من:
   أ. إذا كان العدد المقسوم عليه يساوي صفرا، يظهر الرسالة ("لا يمكن القسمة على صفر").
   ب. إذا كان العدد المقسوم عليه لا يساوي صفرا، يعرض نتيجة القسمة.

2. حوّل الحلّ من pseudocode إلى لغة Python.
3. غيّر الشرط بحيث يكون المقسوم عليه لا يساوي صفرا

Pseudocode

```
START
// إدخال القاسم
INPUT(“Enter 1st integer:”,int1)
// إدخال المقسوم عليه
INPUT(“Enter 2nd integer:”,int2)
IF int2 ==0 then
        DISPLAY(“لا يمكن القسمة على صفر”)
ELSE
        DISPLAY(“result=”,int1/int2)
ENDIF
```

Python Solution

```
// condition1: devisor equal zero

nb1=input("Enter denominator:")
nb2=input("Enter devisor:")
if nb2==0:
    print("division by 0 !!!!")

else:
    print(int(nb1)/int(nb2))


// condition2: devisor not equal to zero

nb1=input("Enter denominator:")
nb2=input("Enter devisor:")
if nb2 != 0:
    print(int(nb1)/int(nb2))
else:
    print("division by 0 not possible!!!!")
```

حلّ النشاط 3:

1. أكتب بلغة **Pseudocode** برنامجا يمكّنك من حساب سعر تذكرة الدخول إلى حلبة سباق البحرين للفورمولا1 بناء على المنطقة وعمر الشخص بحيث يتمتّع الشخص الذي يبلغ من العمر 18 سنة أو أقلّ بتخفيض 25%.

المعطيات:

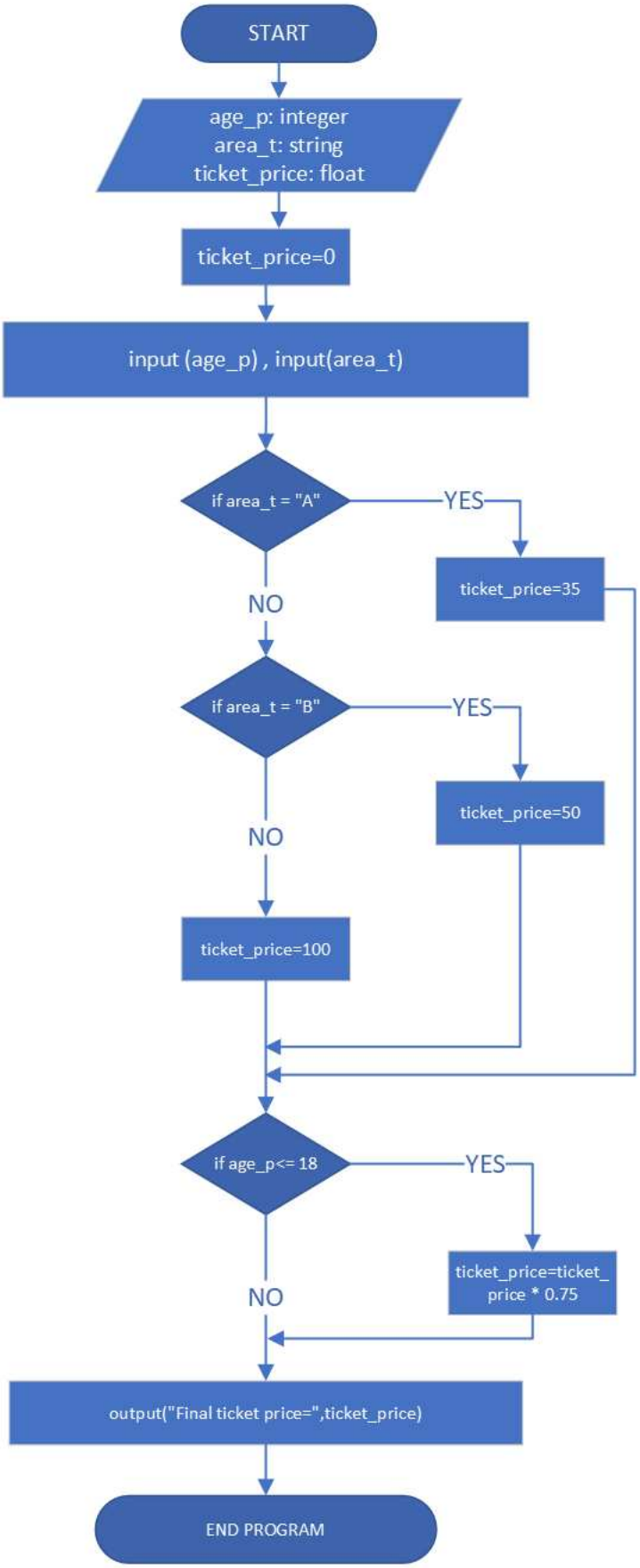
- المنطقة A: سعر التذكرة 35 دب
- المنطقة B: سعر التذكرة 50 دب
- المنطقة VIP: سعر التذكرة 100 دب

2. حوّل الحلّ من pseudocode إلى خريطة تدفقية.
3. حوّل الحلّ إلى لغة Python.

1- Pseudocode solution:

```
START
DECLARE age_p as INTEGER
DECLARE area_t as STRING
DECLARE ticket_price as DECIMAL
ticket_price = 0
INPUT("Enter the age of the person:",age_p)
INPUT("Enter the Ticket Area:",area_t)
IF area_t = A then ticket_price = 35
        ELSE
        IF area_t = B then ticket_price = 50
        ELSE
        ticket_price = 100
        ENDIF
ENDIF
IF age_p <=18 then
        ticket_price = ticket_price *0.75
ENDIF
DISPLAY("TICKET PRICE = ",ticket_price)
PROGRAM END
```

2-Flowchart solution:



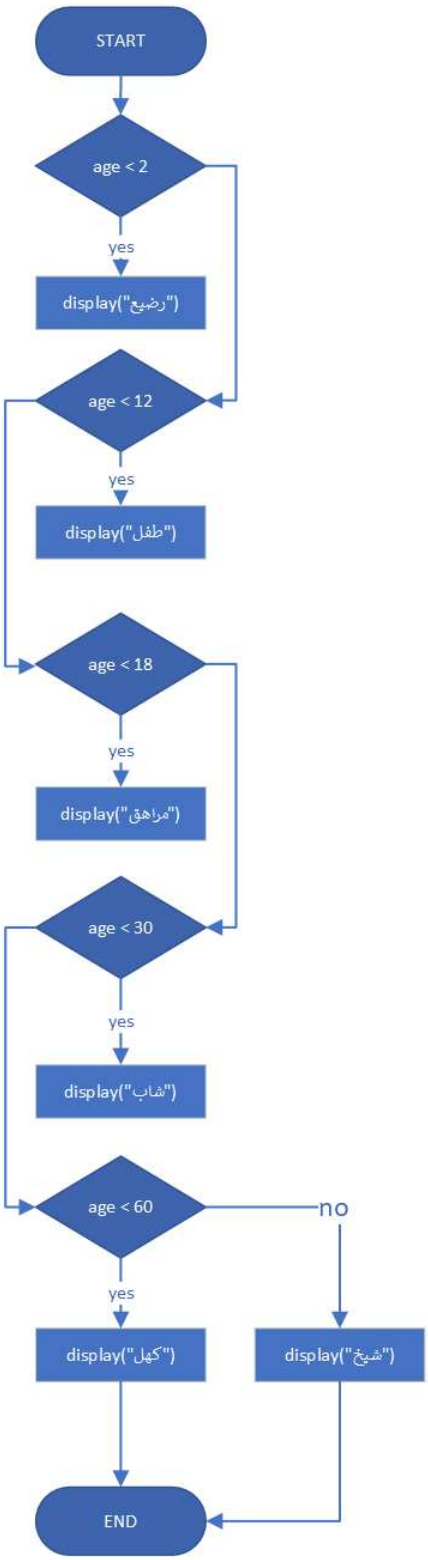
3-Python solution

```
1  age_p=input("Plz Enter the person age:")
2  area_t = input("Plz Enter the Area:")
3  if area_t=="A":
4      ticket_price=35
5  elif area_t=="B":
6      ticket_price=50
7  else:
8      ticket_price=100
9
10 if int(age_p)<=18:
11     ticket_price=ticket_price*0.75
12 print("Final ticket price=",ticket_price)
13
```

حلّ النشاط 4:

أرغب في كتابة برنامج بلغة Python لعرض الوصف المناسب لشخص ما حسب عمرهِ بناء على الشروط الآتية:

الشرط	الوصف
أصغر من 2 سنة	رضيع
أصغر من 12 سنة	طفل
أصغر من 18 سنة	مراهق
أصغر من 30 سنة	شاب
أصغر من 60 سنة	كهل
أخرى	شيخٌ



1- هل الخريطة التدفقيّة أمامك تمثّل الحلّ المناسب؟ لا

2- أجب عن الآتي:

a. ما هي نتيجة إسناد القيمة 17 إلى المتغيّر age؟
"كهل"

b. في حال عكسنا علامة المقارنة إلى أكبر من، ماذا ستكون النتيجة؟ "شيخ"

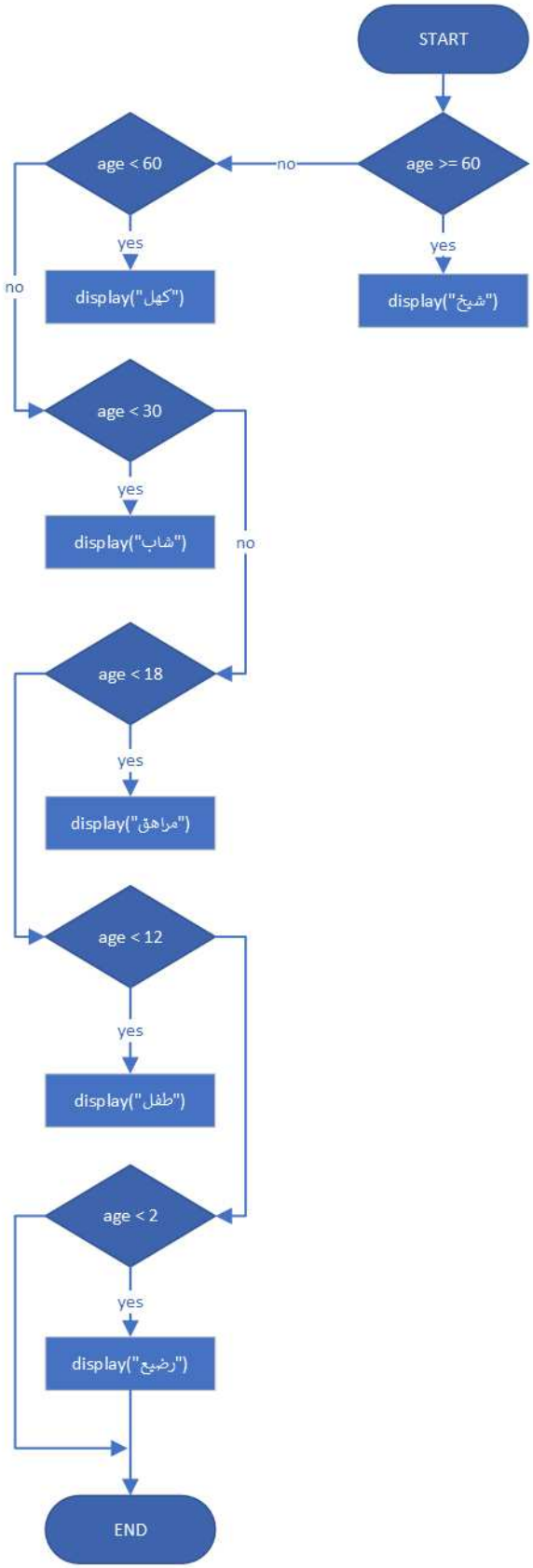
c. برّر سبب الخطأ!: أيّ قيمة ندخلها أصغر من 60 ستكون النتيجة دائما = "كهل"، لأنّ الجمل الشرطية جاءت متتالية ببنيتها البسيطة وفي كلّ جملة يتحقّق الشرط.

3- ما هو الحلّ الذي تراه مناسبا؟

i. البدء بالمعيار أكبر أو يساوي 60.

ii. اعتماد الجملة الشرطية المتداخلة.

4- قم بتصميم الخريطة التدفقيّة الصحيحة.



الدرس 11: الجمل التكراريّة

1- جملة for loop

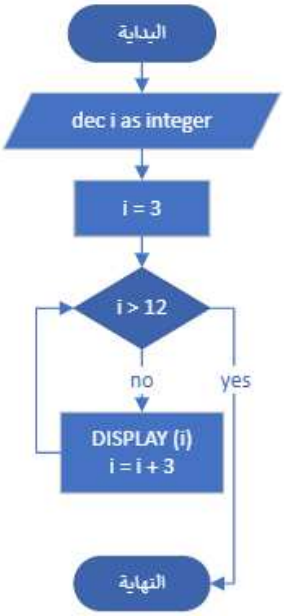
عزيزي المعلّم: قم باختيار مثال من الحياة يقوم فيه شخص بتكرار تنفيذ مجموعة من المهام، مثال القفز، الركض حيث تكون حركة الجسم في تكرار مستمرّ حتى التوقف.

اجعل الطلبة يبحثون عن مواقف تكرار أخرى واسألهم كيف سينتهي هذا التكرار؟ بمعنى إيجاد الشرط المناسب لوقف التكرار.

حلّ النشاط 1:

1- صمّم الخريطة التدفقية **Flowchart**، تمكّنك من عرض مضاعفات الرقم 3 الآتية:

OUTPUT:
3
6
9
12



2- الحلّ بلغة PYTHON

a. **الحلّ الأول**: المرور على جميع القيم للمتغيّر (i) والتأكد من أنّ باقي القسمة على 3 يساوي 0.

```
for i in range(3,13):
    if i % 3 == 0:
        print(i)
```

b. **الحلّ الثاني**: استخدام المتغيّر الثالث للدالة range() والذي يمثّل النمط.

```
for i in range(3,13,3):print(i)
```

c. **الحلّ الثالث**: استخدام المتغيّر (i) لمثّل عدد مرّات التكرار. مرّات التكرار ضمن الدالة range (1,5) تساوي 4 مرّات بما أنّ القيمة 5 لا تحسب ضمن المجال.

```
for i in range(1,5):
    print(i*3)
```

حلّ النشاط 2:

```
sum=0
s3 = [6, 5, 12, 7, 9]
for i in range(len(s3)):
    if (s3[i] % 3) != 0:
        continue
    else:
        sum=sum+s3[i]
print(sum)
```

توضيح الحلّ:

السطر 3: **len(s3)** ← لحساب عدد عناصر السلسلة s3

السطر4: في حال كان العنصر لا يقبل القسمة على 3 يتجاوزه البرنامج بفضل الأمر continue.

حلّ النشاط 3:

لنعتبر السلسلة الآتية:

names = ['ahmed', 'salah', 'Ali', 'ridha', 'amal']

1- أكتب برنامجا بلغة Python يعرض الأسماء التي تبدأ بحرف "a" أو "A" فقط.

الحلّ والتوضيح:

```
names = ['ahmed', 'salah', 'Ali', 'ridha', 'amal']
for i in names:
    if i[0] == "a" or i[0]=="A":
        print(i)
```

- المتغيّر i هنا يمثّل عناصر السلسلة names
- i[0] يمثّل أول حرف من الاسم.

حلّ النشاط 4:

لنعتبر السلسلتان الآتيتان:

Races = ['BahrainGP', 'SaudiGP', 'AustralianGP', 'ChineseGP', 'AzerbaijanGP']

Dates = ['2023-03-05', '2023-03-19', '2023-04-02', '2023-04-16','2023-04-30']

تمّ إلغاء سباق "**ChineseGP**" الذي كان سيقام في تاريخ "**2023-04-16**"، أكتب برنامجا بلغة Python يمكّن من:

1- حذف السباق الملغى من السلسلتين "Races" و "Dates".

2- عرض كل سباق متبوعا بتاريخه.

```
Races = ['BahrainGP', 'SaudiGP', 'AustralianGP',
         'ChineseGP', 'AzerbaijanGP']
Dates = ['2023-03-05', '2023-03-19', '2023-04-02',
         '2023-04-16','2023-04-30']
ind = Races.index('ChineseGP')
Races.pop(ind)
Dates.pop(ind)
l=len(Races)
for i in range(l):
    print(Races[i])
    print(Dates[i])
```

2- جملة **while** *condition is True* **loop**

تختلف هذه الجملة التكرارية عن الجملة التكرارية السابقة، أنّها تستمرّ بتكرار الأوامر حتى تحقّق الشرط، بخلاف الجملة التكراريّة السابقة التي تعتمد على عدد محدّد من التكرارات يتمّ تحديدها.

حلّ النشاط 5:

أريد أن يستمرّ الحاسوب في حساب مجموع أعداد مختلفة مدخلة حتى يفوق مجموعها الـ 100 ثمّ يعرض المجموع.

1- أكتب الحلّ بلغة Pseudocode.

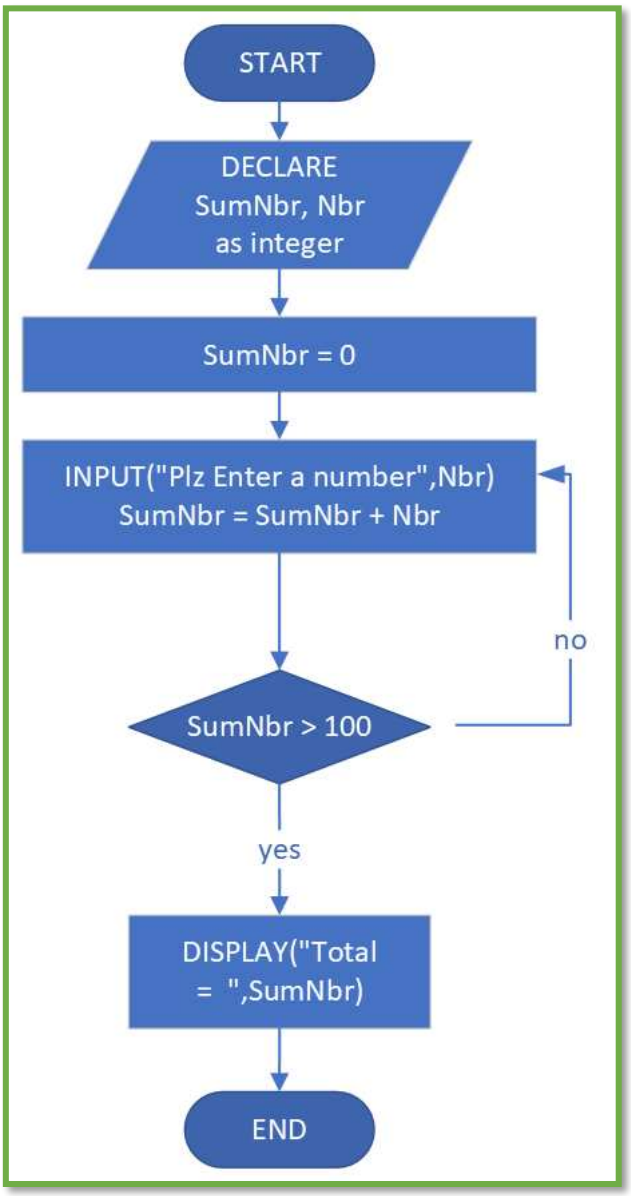
2- ارسم الخريطة التدفقية للحلّ.

3- حوّل الحلّ إلى لغة Python.

Pseudocode

```
START
DECLARE SumNbr as INTEGER
WHILE SumNbr < 100 DO
        INPUT("Plz enter a number",nbr)
        SumNbr = SumNbr + Nbr
END WHILE
DISPLAY("Total= ",SumNbr)
```

Flowchart



Code Python

```
1  SumNbr = 0
2  while SumNbr<=100:
3      Nbr=input("Enter a number:")
4      SumNbr=SumNbr+int(Nbr)
5
6  print("Total=",SumNbr)
7
```

حلّ النشاط 6:

أرغب في برمجة لعبة بسيطة تتمثّل في:

1- يختار الحاسوب عددا عشوائيا صحيحا ضمن مجال تحدّده) .راجع الدالة(random.randint(start, end

2- أخمّن العدد الذي اختاره الحاسوب من خلال إدخال قيمة في متغيّر أعيّنه.

3- إذا كانت القيمة التي أدخلتها أصغر من القيمة العشوائية تظهر الرسالة ("هذه القيمة أصغر من العدد المجهول")

4- إذا كانت القيمة التي أدخلتها أكبر من القيمة العشوائية تظهر الرسالة ("هذه القيمة أكبر من العدد المجهول")

5- إذا أدخلت القيمة الصحيحة تظهر الرسالة ("ممتاز تخمينك صحيح.") ويظهر العدد المجهول.

6- أكتب الحلّ بلغة Python موظّفا الجملة الشرطية والجملة التكرارية المرفقة بخاصيّة else.

Code Python

```
1  import random
2  x=random.randint(20,60)
3  y=1
4  tries = 0
5
6  while int(y) != x:
7      y=input("Gess x:")
8      tries+=1
9      if int(y)<x:
10         print("Your try is less than x.")
11     elif int(y)> x:
12         print("Your try is greater than x.")
13
14 else:
15     print("Excelent the answer is:",int(y))
16     print("Number of tries=",tries)
17
```

ملاحظة" يتطلّب منك عزيزي المعلّم شرح السطر رقم1. وأنّ استخدام بعض الدوال تحتاج إلى استدعاء المكتبة الخاصة بها, مثل مكتبة الدوال الرياضيّة math ومكتبة التواريخ datetime ومكتبة الأعداد العشوائية random وغيرها ...

حلّ النشاط 7:

أكتب برنامجا بلغة Python يمكّنك من:

1- إدخال كلمة ضمن متغيّر باسم "word".

2- التأكّد من أنّ هذه الكلمة متناظرة أو لا.

3- الكلمة المتناظرة هي التي تقرأ من الاتجاهين. (racecar – madam – civic - ...etc)

Code Python

```
1  word=input("Enter a word")
2  bool=0
3  start=0
4  end=len(word)-1
5  mid=len(word)//2
6  while int(start)<=int(mid):
7      if word[start] == word[end]:
8          bool=1
9          start+=1
10         end-=1
11     else:
12         bool=0
13         break
14 if bool==1:
15     print(word," is palindrom")
16 else:
17     print(word," is not palindrom")
```

line	Code Python	Explanation
1	word=input ("Enter a word")	لإدخال كلمة معيّنة عبر لوحة المفاتيح
2	bool=0	تهيئة متغيّر منطقي بالقيمة صفر
3	start=0	تهيئة متغيّر يمثّل أول حرف من الكلمة بالقيمة صفر
4	end=len(word) - 1	تهيئة متغيّر يمثّل (**طول الكلمة - 1**) بما أنّ أوّل حرف موقعه صفر
5	mid=len(word)//2	تهيئة متغيّر يمثّل نصف طول الكلمة والذي سيمثّل أعلى قيمة للمتغيّر start وأقلّ قيمة للمتغيّر end
6	while int(start)<=int(mid):	الجملة التكرارية: سنكرّر مقارنة الحرف الأول بالحرف الأخير ثم

		الحرف الثاني بالحرف قبل الأخير وهكذا حتى بلوغ نصف الكلمة.
7	if word[start] == word[end]:	في حال الحروف متشابهة
8	bool=1	تغيير قيمة المتغيّر المنطقي إلى 1 كمؤشّر بأن الحرةف التي تمّت مقارنتها متشابهة
9	start+=1	الانتقال إلى موقع الحرف الموالي من بداية الكلمة
10	end-=1	الانتقال إلى موقع الحرف السابق من نهاية الكلمة
11	else:	في حال عدم تحقّق الشرط
12	bool=0	تسند إلى المتغيّر المنطقي القيمة 0 للإشارة إلى أنّ الحرفين التي تمّت مقارنتهما ليسا متشابهين.
13	break	للخروج الاجباري من الجملة التكرارية
14	if bool==1:	إذا حافظ المتغيّر bool على القيمة 1 يعني أنّ جميع أزواج الحروف التي تمّت مقارنتها متشابهة.
15	print(word," is palindrome")	عرض رسالة على الشاشة بأنّ الكلمة المدخلة متناظرة.
16	else:	وفي حال كانت قيمة المتغيّر bool صفرا أي أنّ هناك على الأقل زوج من الحروف غير متشابهان
17	print(word," is not palindrome")	عرض رسالة على الشاشة بأنّ الكلمة المدخلة غير متناظرة.

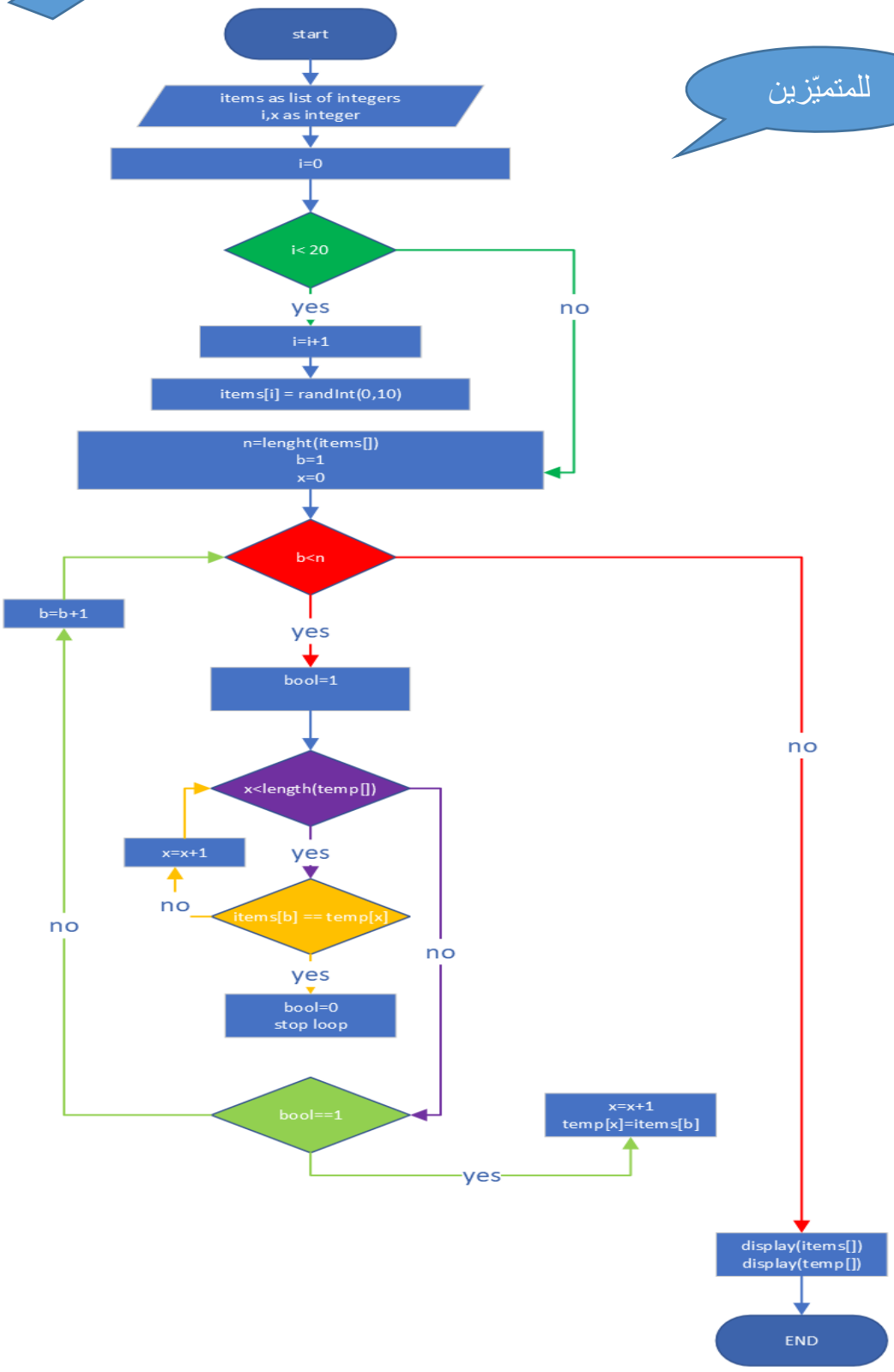
حلّ النشاط 8:

أكتب حلاّ برمجيّا لحذف القيم المكرّرة من سلسلة list. متّبعا التوجيهات الآتية:

1- يتمّ إدخال 20 عددا صحيحا عشوائيا ضمن السلسلة items.
2- أكتب الحل عن طريق الخريطة التدفقيّة.
3- حوّل الكود إلى لغة Python.

Flowchart

للمتميّزين



Code Python

للمتميّزين

```python
import random
items=[]
temp=[]
b=0
'إدخال مائة عنصر عشوائي من الأعداد الصحيحة'
'fill the list with 20 integer'
for i in range(20):
    'the integers are between 0 and 10 '
    items.append(random.randint(0, 10))
'Display the list items after filling'
print(items)
'assign the first item from items to temp'
temp.append(items[0])

n=len(items)
'compare each item of (items starting from the 2nd) by the content of (temp)'
for b in range(b+1,n):
    bool=1
    'to verify if the value in items list exists in temp list'
    for x in range(len(temp)):
        if items[b]==temp[x]:
            'the value exists in temp'
            bool=0
            'exit the loop and return to the next value in items'
            break
    'the value in items dose not exit in temp list'
    if bool==1:
        ' add the new value to temp list'
        temp.append(items[b])
'Display the list temp without duplicates'
print(temp)
```

الحلّ البرمجي لجميع الطلبة

```python
import random
items=[]

'إدخال مائة عنصر عشوائي من الأعداد الصحيحة'
for i in range(20):
    'the integers are between 0 and 10 '
    items.append(random.randint(0, 10))
'Display the list items after filling'
print(items)
temp = []
'insert the new value from items to temp if not already exists '
[temp.append(x) for x in items if x not in temp]

# printing list after removal
print ("The list after removing duplicates : " + str(temp))

```

توفّر برمجيّة بايثون أدوات وطرق مبسّطة لحلّ مثل هذه المسائل التي كانت سابقا تحتاج إلى برمجة في حال اعتمدنا لغة برمجة مختلفة، وفي هذه المسألة سنعتمد الحلّ الآتي:

حلّ النشاط 9:

الحلّ البرمجي الأول "تمايز": بدون الاستعانة بالدالة split()

```
'the variable to store each word'
word = ""
txt="Welcome To Bahrain"
'check text until finding the space between words'
for i in range(len(txt)):
    'if not space add this character to the variable word'
    if txt[i]!=" ":
        word=word+txt[i]

    else:
        'if space display the word'
        print(word)
        'reset the variable word for the next one'
        word=''
    'the last word has no space after it will be written when i reach the end of text'
    if i == len(txt)-1:
        print(word)

```

الحلّ البرمجي الثاني: الاستعانة بالدالة split():

```
txt="Welcome To Bahrain"

words=txt.split()

'The result will be a list'
print(words)

```

الدرس 12: بناء الدّوال / Building Functions

عزيزي المعلّم، ربّما يعتبر هذا الدرس ذا مستوى عالي من الصعوبة بالنسبة للطالب، لذا عليك البدء بأمثلة بسيطة جدّا والتركيز على الهدف من هذا الدرس، وهو أن يتقن الطالب بناء الدالة ومن ثمّ توظيفها قبل العمل على حلّ مسائل ذات درجة صعوبة أعلى.

حلّ النشاط 1:

أكتب برنامجا بلغة Python يمكّنك من برمجة نموذج لآلة حاسبة متّبعا الخطوات الآتية:

1- يتمّ إدخال العددين اللذين سيتمّ إجراء العملية الحسابية عليهما من خلال الجزء الرئيسي من البرنامج.

2- يتمّ إدخال المعامل الحسابي المطلوب من العوامل الرياضيّة الآتية فقط: (+ أو - أو / أو *)

3- في حال إدخال علامة الجمع: يتمّ استدعاء دالة الجمع وتنفيذ العمليّة وإرجاع النتيجة إلى البرنامج الرئيسي.

4- في حال إدخال علامة الطرح: يتمّ استدعاء دالة الطرح وتنفيذ العمليّة وإرجاع النتيجة إلى البرنامج الرئيسي.

5- وكذلك الحال لبقيّة العوامل الحسابية.

6- يتمّ عرض نتيجة العملية الحسابية من البرنامج الرئيسي من خلال القيمة المُرْجَعَة من الدالة المعنيّة.

-1

```
'Main program'
x=input("Plz Enter first number:")
y=input("Plz Enter second number:")
x=int(x)
y=int(y)
```

-2

```
'Main program'
x=input("Plz Enter first number:")
y=input("Plz Enter second number:")
x=int(x)
y=int(y)
Op=''
while Op not in ['+','-','*','/']:
    Op=input("Plz Enter right operator (+ or - or * or /):")
```

-3 -4 / -5 /

```
while Op not in ['+','-','*','/']:
    Op=input("Plz Enter right operator (+ or - or * or /):")
if Op=='+':
    Result=summ(x,y)
if Op=='-':
    Result=minus(x,y)
if Op=='*':
    Result=multip(x,y)
if Op=='/':
    Result=devis(x,y)
```

-6

```
if Op=='+':
    Result=summ(x,y)
if Op=='-':
    Result=minus(x,y)
if Op=='*':
    Result=multip(x,y)
if Op=='/':
    Result=devis(x,y)

print("The result of ",Op," =",Result)
```

حلّ النشاط 2:

1- عيّن ضمن سلسلة list مجموعة هامة من الكلمات.

2- اختر كلمة عشوائية من القائمة.

3- ادخل حرفا من الحروف.

4- أظهر عدد مرات تواجد هذا الحرف في الكلمة ومواقعها.

5- حلّ المسألة مستعينا بالدوال والخوارزميات اللازمة.

مثال: الكلمةmanama :

الحرفa :

النتيجة: عدد مرات تواجد الحرف a ضمن الكلمة manama = 3

المواقع index: (2، 4، 6)

توصيف للحلّ:

1- تعيين سلسلة list وتعبئتها يدويا بمجموعة من الكلمات.

2- اختيار كلمة عشوائية:

a. يمكن الاختيار من خلال ادخال موقع الكلمة index يدويا.

b. يمكن الاختيار من خلال دالة randint() لموقع الكلمة index عشوائيا.

3- ادخال حرف عن طريق الدالة input()

4- إيجاد عدد مرات تواجد الحرف في النص من خلال الدمج بين الجملة التكرارية للمرور على جميع حروف النص والجملة الشرطية للتأكد من الحرف الحالي للنص إن كان يساوي الحرف المطلوب أو لا.

a. في حال تطابق الحرفين يتمّ زيادة 1 لعداد يتمّ تعيينه بداية البرنامج.

b. أما بالنسبة لموقع الحروف المتطابقة مع الحرف المطلوب: يتمّ تعيين سلسلة list وفي كل تطابق للحرفين يتمّ إدراج موقع الحرف عن طريق الدالة index()

5- الدوال التي يمكن بناؤها وتوظيفها:

a. دالة لاختيار موقع الكلمة العشوائي

b. دالة للبحث عن الحرف المطلوب في الكلمة المختارة وتحديد موقع كلّ حرف ضمن هذه الكلمة.

Code Python

1- ضمن البرنامج الرئيسي: تعبئة المتغيّر WORDS بمجموعة من الكلمات والعبارات.

```
24 #Main Program
25 import random
26 # list of words
27 WORDS = ["bahrain","welcome","Artificial Intelligence","robotics","internet of things",]
```

2- بناء الدالة **RanWordInd()** والتي ستحدّد الكلمة المختارة من خلال موقعها ضمن السلسلة **WORDS** والذي بدوره يتمّ تحديده عشوائيا من خلال الدالة **randint()**.

```
2 #Function to choose a word from the list
3 def RanWordInd(ls):
4     n=len(ls)
5     ind=random.randint(0,n-1)
6     TheWord=ls[ind]
7     return TheWord
```

3- ضمن البرنامج الرئيسي يتمّ استدعاء مكتبة الدوال العشوائية random. ثمّ استدعاء الدالة RanWordInd() ويحفظ النتيجة (وهي الكلمة التي تمّ اختيارها) في المتغيّر wordCh، ثمّ يحدّد الحرف الذي يريد البحث عنه ضمن الكلمة من خلال الدالة **input** ويحفظه في المتغيّر ch.

```
24 #Main Program
25 import random
26 # list of words
27 WORDS = ["bahrain","welcome","Artificial Intelligence",
28         "robotics","internet of things",]
29 #Call 1st function where the word will be choosen
30 wordCh=RanWordInd(WORDS)
31 # Enter a character
32 ch=input("Plz Enter a character:")
```

4- بناء الدالة FindChOcc()

```
#function to find occurance and locations
def FindChOcc(wrd,c):
    #count: to Determine the number of occurances
    count = 0
    #declare a list to store the c indexes
    occ=[]
    #indCh to save the current index
    indCh=0
    for i in wrd:
        #Each loop indCh is incremented by 1
        indCh+=1
        if i == c:
            count = count + 1
            occ.append(indCh)
    return(count,occ)
```

حيث []occ ستحتفظ بقيم مواقع الحرف المطلوب البحث عنه.

المتغيّر indCh يمثّل الموقع الحالي للحرف ضمن الكلمة wrd.

return(count,occ): النتيجة التي سيتمّ ارجاعها إلى البرنامج الرئيسي وتتكوّن من جزئين: 1- عدد الحروف 2- قائمة مواقع الحرف ضمن الكلمة.

5- يتمّ استدعاء الدالة FindChOcc() بعواملها المكوّنة من متغيّر الكلمة ومتغيّر الحرف حيث تخزّن النتيجة المرجعة في المتغيّر cntCh

```
#call the 2nd function and the returned values are inside a tupel
cntCh=FindChOcc(wordCh,ch)

# printing result
print ("Count of ",ch, "in ",wordCh,": ",cntCh[0])  # cntCh[0] = 1st parameter returned
print("locations:",cntCh[1]) ## cntCh[1] = 2nd parameter returned
```

سأتعرّف على محتوى المتغيّر cntCh، لذا سأقوم بعرضه على الشاشة في السطر 36 حسب الصورة أعلاه:

print(cntCh)

ثمّ سأقوم بتشغيل البرنامج واختيار الحرف e، ستكون إحدى النتائج كالآتي:

```
>>> %Run CountChar.py

  Plz Enter a character:e
  (3, [15, 20, 23])
  Count of  e in  Artificial Intelligence :  3
  locations: [15, 20, 23]
```

محتوى المتغيّر هو: (3,[15, 20, 23])؛ حيث 3 هو عدد مرات تواجد الحرف e ضمن الكلمة و[15, 20, 23] هي مواقع هذا الحرف.

```
37  # printing result
38  print ("Count of ",ch, "in ",wordCh,": ",cntCh[0])  # cntCh[0] = 1st parameter returned
39  print("locations:",cntCh[1]) ## cntCh[1] = 2nd parameter returned
40
```

ألاحظ أنّني أستطيع طباعة كلّ عنصر من المتغيّر cntCh على حده من خلال الـindex الخاص بكلّ جزء.

cntCh[0] = 3

cntCh[1] = [15, 20, 23]

انتهى

والله ولي التوفيق