



# مقدمة في البرمجة Introduction To Programming

**IPG101**

## الفصل الثاني تصميم البرمجيات Programs Design

## الكلمات المفتاحية

خوارزمية، برنامج، خوارزمية حسابية، خوارزمية غير حسابية، خوارزمية تسلسلية، خوارزمية غير تسلسلية، قرار، تكرار، مخطط تدفقي، شبه الشيفرة.

## ملخص الفصل

يتضمن هذا الفصل مدخلاً إلى البرمجة، حيث يركز على مهارات التفكير البرمجي من خلال شرح منهجية تصميم البرامج عبر تحديد مدخلاتها ومخرجاتها وعملياتها وتصميم خوارزميات الحل لها، ومن ثم يقدم تعريفاً بالخوارزميات وأنواعها وطرق التعبير عنها حيث يركز على أسلوب المخططات التدفقية وأسلوب شبه الشيفرة.

## أهداف الفصل

بنهاية هذا الفصل سيكون الطالب قادراً على:

- وصف منهجية حل المسائل باستخدام مفاهيم هندسة البرمجيات.
- معرفة مفهوم الخوارزمية.
- معرفة أنواع الخوارزميات.
- التعبير عن الخوارزميات بطرق مختلفة.
- رسم المخطط التدفقي المعبر عن خوارزمية حل مسألة ما.
- تصميم خوارزمية الحل لمسألة ما.

## محتويات الفصل

1. مقدمة.
2. منهجية تطوير البرمجيات.
3. مدخل إلى الخوارزميات.
4. أنواع الخوارزميات.
5. طرق التعبير عن الخوارزميات.
6. التحكم بمسار التنفيذ.
7. تمارين وأنشطة.

## 1- مقدمة

تعتبر مهارة الانتقال من المسألة المطروحة إلى تصميم الحل اللازم لها واحدة من أهم المهارات التي يجب على المبرمج أن يمتلكها وهذه المهارة تدعى مهارة التفكير البرمجي وتقوم على فهم وتحليل المسألة واستخراج مكوناتها الأساسية وتصميم خوارزمية الحل.

إن مفهوم الخوارزمية Algorithm مفهوم قديم يعود إلى مطلع القرن التاسع الميلادي يعود إلى العالم العربي محمد بن موسى الخوارزمي الذي برز في علوم الرياضيات والفلك وكان أول من وضع مبادئ علم الجبر، إلا أن هذا المفهوم ازداد شيوعاً وتعزز به كثيراً في المدة الأخيرة. ومنذ ظهور الحواسيب، شاع استخدامها وتركيز الاهتمام على مبادئها في الكتب والأبحاث وميادين متعددة من النشاطات العلمية والتطبيقية.

يعتبر الحاسوب أداة مذهلة في حل طيف واسع من المسائل. لجعل الحاسوب يقوم بأي شيء (حل أي مسألة)، لابد من كتابة برنامج حاسوبي computer program، تقوم في هذا البرنامج بإخبار الحاسوب، خطوة خطوة بما ترغب منه أن يقوم به بدقة. يقوم الحاسوب عندئذ بتنفيذ البرنامج متتبعاً كل خطوة لإنجاز الهدف النهائي.

يعرف هذا التتالي من الخطوات الواجب تنفيذها بالترتيب لحل المسألة من قبل الحاسوب باسم الخوارزمية algorithm. أما المخطط التدفقي flowchart فهو عبارة عن تمثيل رسومي أو ترميزي للخوارزمية. إنه التمثيل المخططاتي للحل المعطى للمسألة خطوة خطوة. في حين يتألف تصميم البرنامج program design من الخطوات الواجب على المبرمج تنفيذها قبل البدء بكتابة شيفرة البرنامج بلغة البرمجة المرغوبة. إن التصميم الملائم للبرنامج يساعد المبرمجين الآخرين على صيانة وتعديل البرنامج لاحقاً.

## 2- منهجية تطوير البرمجيات

تعتمد هندسة البرمجيات طرائق أساسية محددة للحصول على حلول برمجية للمسائل. وعلى الرغم من أن المسائل بعد ذاتها تتنوع وأن لغات البرمجة المستخدمة في الحل تتنوع كذلك، إلا أن هناك عدة مراحل مشتركة في عملية تطوير وبناء البرمجيات:

1- **مرحلة التصميم design:** وفيها يتم تحليل المسألة وتصميم الحل، وينتج عنها خوارزمية algorithm لحل المسألة.

2- **مرحلة كتابة الشيفرة coding:** يتم كتابة الحل وفق قواعد لغة عالية المستوى (مثل C#) فينتج برنامج.

3- **مرحلة الاختبار، التنفيذ والتنقيح testing, execution and debugging:** وفيها يتم اختبار البرنامج وفق قواعد صارمة وإزالة أي أخطاء.

4- **مرحلة الصيانة maintenance:** وفيها يتم تحديث وتعديل البرنامج، بحسب الحاجة، لتلبية المتطلبات المستجدة للمستخدمين.

تتعدد المنهجيات المعتمدة في تنفيذ المرحلة الأولى (أي تصميم الحل) إلا أننا سنعتمد منهجية التصميم المتمركز على الأغراض Object-Centered Design، حيث تتألف هذه المنهجية في أبسط أشكالها من المراحل التالية:

- 1- **مرحلة توصيف السلوك Behavior** حيث يتم تحديد كيف نرغب من برنامجنا أن يتصرف بأكبر قدر ممكن من الدقة.
- 2- **مرحلة تحديد الأغراض Objects** حيث يتم تحديد أغراض العالم الحقيقي في المسألة وتصنيفها وتوصيفها.
- 3- **مرحلة تحديد العمليات Operations** حيث يتم تحديد العمليات اللازمة لحل المسألة وتوصيفها.
- 4- **مرحلة كتابة الخوارزمية Algorithm** حيث يتم ترتيب الأغراض والعمليات الموصفة وفق ترتيب يؤدي إلى حل المسألة بما يحقق السلوك الموصوف للبرنامج.

### 3- مدخل إلى الخوارزميات

قبل كتابة أي برنامج لحل مسألة ما، يجب أن يتوفر لدينا فهمٌ شاملٌ للمسألة المطروحة. وهذا يتضمن تحديد وتوصيف المعطيات التي نعتمد عليها أو ننطلق منها، والنتائج التي نريد الوصول إليها. قبل كتابة البرنامج نتبع أسلوباً منهجياً للحل، ونعبر عن هذا الحل بطريقة مؤطرة مهيكلة لا لبس فيها كافية لتتقنا فيما بعد إلى لغة برمجة معينة دون عناء كبير.

يتم حل أي مسألة برمجية وضع خطة للحل ندعوها الخوارزمية Algorithm، حيث يمكن تعريف الخوارزمية على أنها **مجموعة متتالية ومنتهية من الخطوات التي تعبر عن حل مسألة ما**. تتضمن الخوارزمية إذا خطوات حل مسألة عبر تحديد:

- 1- الأفعال الواجب تنفيذها.
  - 2- الترتيب الواجب إتباعه من أجل تنفيذ الأفعال السابقة.
- يجب أن تكون الأفعال وترتيب التنفيذ موصفاً على وجهٍ لا يدعو إلى اللبس أو التأويل
- إن إنجاز أي أمر بشكل سليم يتطلب وضع خوارزمية صحيحة وفعالة له. على سبيل المثال، لنتناول مسألة طرح سحب كرة حمراء اللون من صندوق يحوي عدداً كبيراً من الكرات الملونة. يمكن التعبير عن إنجاز مثل هذه المسألة من خلال الخوارزمية التالية:

- 1- البداية.
- 2- سحب كرة من الصندوق.
- 3- اختبار لون الكرة المسحوبة فإن كان أحمر، الانتقال إلى الخطوة 4 وإلا تكرار الخطوات 2 و 3.
- 4- النهاية.

تعتبر الخوارزميات مكوناً جوهرياً في مجال الحوسبة وذلك لأنها تساعد على بناء الإجراءات اللازمة بطريقة نظامية. وتوجد عدة خوارزميات لحل المسألة الواحدة ولكن أفضل هذه الخوارزميات هي التي تصل إلى النتيجة بأقل جهد وزمن ممكنين "أي سهولة الفهم وسريعة التنفيذ".

إن استخدام الخوارزمية الجيدة يشبه استخدام الأداة الصحيحة في الورشة. فهو يتيح إنجاز العمل المطلوب بطريقة صحيحة وبالمقدار الصحيح من الجهد. أما استخدام الخوارزمية الخاطئة أو الخوارزمية الغير معرفة بشكل واضح فهو أمر يشبه محاولة قطع شجرة بسكين إذا على الرغم من أنك قد تتمكن من إنجاز العمل إلا أنك تتساءل عن مدى الفعالية التي أنجز فيها الأمر.

على سبيل المثال، لنفرض أن لديك صديقاً يقيم في مدينة بعيدة ويرغب بزيارتك، فهناك عدد كبير من الخوارزميات التي تتيح لك إرشاده إلى كيفية إنجاز هذا الأمر والوصل بشكل سليم إلى منزلك، نبين فيما يلي بعضها:

#### ❖ خوارزمية سيارة الأجرة

- إذهب إلى موقف سيارات الأجرة.
- إركب سيارة أجرة.
- أعط العنوان للسائق.

#### ❖ خوارزمية إتصل بي

- عندما تصل إلى محطة الباصات، إتصل برقم هاتفي.
- انتظرنني على مدخل محطة الباصات.

#### ❖ خوارزمية حافلات النقل العام (السرفيس)

- إتجه إلى مدخل محطة الباصات.
- إركب حافلة النقل رقم xxx.
- إنزل من هذه الحافلة في الموقع YYY.
- إركب حافلة النقل ZZZ..
- إنزل من هذه الحافلة في الموقع TTT..
- قم بالسير ثلاث بنايات نحو الغرب.

إن جميع هذه الخوارزميات تحقق الهدف ذاته، إلا أن كل خوارزمية تتجز هذا الأمر بطريقة مختلفة. كما أن لكل خوارزمية كلفة مختلفة وتحتاج إلى زمن مختلف. فمثلاً خوارزمية سيارة الأجرة هي الأسرع ولكنها الأكثر تكلفة مادية. في حين أن خوارزمية حافلة النقل العام هي الأقل كلفة مادية لكنها الأبطأ. يتوقف نوع الخوارزمية التي عليك اختيارها حسب أولوية الاعتبارات (الوقت، الكلفة المادية).

بطريقة مشابهة تماماً، عند برمجة الحاسوب هناك غالباً العديد من الخوارزميات لتحقيق مهمة معينة. كل خوارزمية لها حسنات ومساوئ بحسب ظروف الحل.

هناك ثلاثة أسباب لاستخدام الخوارزميات وهي الجدوى efficiency، التجريد abstraction وقابلية إعادة الاستخدام reusability.

- **الجدوى efficiency:** نواجه بعض الأنواع من المسائل، مثل مسائل الفرز، بشكل متكرر. يجب عندها استخدام خوارزمية مجدية لحل مثل هذه المسائل بزمان مقبول، إذا أن عامل الكلفة هنا أمر أساسي في كل خوارزمية نقوم بتصميمها.
- **التجريد abstraction:** تمكنا الخوارزميات من تحقيق مستوى معيناً من التجريد في حل المسائل وذلك لأن الكثير من المسائل التي قد تبدو معقدة يمكن أن تتم تجزئتها إلى مسائل أبسط، يوجد لكل منها خوارزمية حل شهيرة.
- **قابلية إعادة الاستخدام reusability:** غالباً ماتكون الخوارزميات قابلة لإعادة الاستخدام في العديد من الأوضاع المختلفة. وبما أن الكثير من الخوارزميات المعروفة تمثل تعميماً لخوارزميات أكثر تعقيداً، وبما أن الكثير من المسائل المعقدة يمكن تجزئتها إلى خوارزميات أبسط، فإنه من الممكن حل مسألة معقدة بواسطة استخدام حلول مسابقة لمسائل أبسط.

#### 4- أنواع الخوارزميات

كخلاصة لكل ماسبق، يمكن تعريف الخوارزمية بأنها توصيف دقيق وكامل على شكل خطوات متسلسلة معدودة ومعرفة تحدد طريقة إنجاز عمل ما، أو حل مسألة ما. و يمكن تقسيم الخوارزميات بشكل عام إلى حسابية وغير حسابية (عامة).

#### الخوارزميات غير الحسابية:

يقصد بالخوارزمية غير الحسابية، الخوارزمية التي لا تتضمن التعامل مع مقادير حسابية ولا عمليات حسابية أو رياضية ولا تعيد نتائج رقمية. ربما كانت الخوارزميات غير الحسابية هي أكثر الخوارزميات استخداماً، ونذكر منها تلك التي تقوم بمعالجة النصوص، وتخزين المعلومات واستعادتها، وإدارة قواعد البيانات، والمساعدة في اتخاذ القرار في جميع نواحي الحياة.

#### مثال 1: خوارزمية تشغيل برنامج حاسوبي:

- 1- اضغط على زر التشغيل؛
  - 2- انتظر ظهور شاشة الاستقبال؛
  - 3- إذا كان من الضروري أن تُعرّف عن نفسك: أدخل إسم حسابك وكلمة مرورك؛
  - 4- ابحث عن أيقونة البرنامج الذي تريد تشغيله وانقر عليها نقرتين بالفأرة.
- (لاحظ هنا أن الأفعال قد تكون ملتبسة حسب الحالات: شاشة الاستقبال قد تكون الشاشة البيانية : أيقونات ونوافذ....، وقد تكون الشاشة السوداء : نصية فقط، وبالتالي خوارزمتنا لا تغطي جميع الحالات!! ولكنها تتفد المطلوب أي تشغيل برنامج حاسوبي ، في الحالة الأكثر شيوعاً)

مثال 2: خوارزمية تحضير بيضة مقليه:

- 1- ضع الوعاء على النار؛
  - 2- أضف مقدار نصف ملعقة صغيرة من الزبدة؛
  - 3- انتظر ذوبان الزبدة؛
  - 4- اكسر البيضة وصع محتواها ضمن الوعاء؛
  - 5- انتظر حتى تتضج البيضة.
- (لاحظ أن ترتيب الخطوات يغير في إنجاز العمل المطلوب من الخوارزمية وقد يعطي نتائج خاطئة، فلو وضعنا الخطوة 4 قبل الخطوة 2؛ أي وضع البيض قبل الزبدة، لما حصلنا على "العجة" اللذيذة!!)

## الخوارزميات الحسابية

يقصد بالخوارزمية الحسابية تلك التي تتعامل مع المقادير الرياضية. وقد شاع لدى الرياضيين تقديم الأمثلة على هذه الخوارزميات حتى ارتبط مفهوم الخوارزمية عند الكثيرين بهذا النوع.

مثال 1: خوارزمية حساب قيمة المقدار result في المعادلة الرياضية التالية:

$$result = \frac{4a + b}{3ab - b + a}$$

- 1- البداية.
- 2- الحصول على قيمة a.
- 3- الحصول على قيمة b.
- 4- حساب قيمة البسط: num=4a+b
- 5- حساب قيمة المقام: denum=3ab-b+a.
- 6- حساب قيمة المقدار result=num/denum.
- 7- النهاية.

مثال 2: خوارزمية تحديد العدد الأكبر من مجموعة أعداد:

1. أدخل العدد الأول إلى المتحول KMax
2. أدخل العدد التالي إلى المتحول K
3. إذا (K تساوي 999) اذهب إلى الخطوة 6
4. إذا K أكبر من KMax أسند K إلى KMAX
5. اذهب إلى الخطوة 2
6. اكتب "قيمة العدد الأكبر"، KMax

## 5- طرق التعبير عن الخوارزميات

يمكن التعبير عن الخوارزميات بأساليب عدة مختلفة، منها اللغات الطبيعية natural languages، شبه الشيفرة pseudocode، المخططات التدفقية flowcharts، ولغات البرمجة programming languages. كما أن من الممكن استخدام مزيج من الأساليب السابقة لدى التعبير عن خوارزمية ما.

نبين فيما يلي شرحاً لكل طريقة من هذه الطرق، حيث سنقوم بتوضيح الفارق بينها من خلال بناء خوارزمية لإيجاد مجموع عددين وإظهار الناتج.

### التعبير عن الخوارزميات باستخدام اللغات الطبيعية Natural Language

وهي الطريقة التلقائية التي نقوم باستخدامها في التعبير عن خطوات إنجاز أي أمر، حيث نقوم باستخدام اللغة الطبيعية التي نستخدمها في المحادثة ( كاللغة العربية مثلاً ) لتحقيق هذا الأمر.

تتميز هذه الطريقة بالسهولة، وإمكانية استخدامها من قبل جميع الناس ولهذا فهي تستخدم على نطاق واسع في الحياة اليومية لصياغة بعض أنواع الخوارزميات الموجهة للجمهور كتعليمات تشغيل الأجهزة أو وصفات تحضير أطباق الطعام. إلا أن من مساوئ هذا الأسلوب في التعبير عن الخوارزميات أنه يميل لأن يكون إنشائياً وغامضاً، وهو نادراً ما يستخدم من أجل الخوارزميات المعقدة أو التقنية.

على سبيل المثال، لو رغبتنا بإيجاد مجموع عددين وإظهار ناتج الجمع:

يمكن التعبير عن مثل هذه الخوارزمية باللغة الطبيعية كما يلي:

- 1- البداية
- 2- اقرأ (أدخل) الرقم الأول وقم بتخزينه في الحاسوب.
- 3- اقرأ الرقم الثاني واحفظه في الحاسوب.
- 4- أضف الرقم الأول إلى الرقم الثاني للحصول على المجموع وقم بتخزينه في الحاسوب.
- 5- إعرض (أخرج) قيمة المجموع.
- 6- النهاية

### التعبير عن الخوارزميات باستخدام شبه الشيفرة Pseudo code

تقوم هذه الطريقة على استخدام عبارات خاصة في التعبير عن خطوات الخوارزمية قريبة جداً من العبارات المستخدمة في لغات البرمجة إلا أنها مستقلة عن أي لغة برمجة. واستخدام صياغات رياضية شبه برمجية للعمليات وبنى تخزين القيم.



يبين الجدول التالي قائمة بأهم العبارات أو الأوامر التي يمكن أن تستخدم في صياغة التعابير على أوامر الخوارزمية المكتوبة باستخدام لغة شبه الشيفرة:

العملية	أوامر شبه الشيفرة
البداية والنهاية	<b>START, BEGIN, STOP, END</b>
الإدخال	<b>INPUT, READ, GET</b>
الإخراج	<b>PRINT, DISPLAY, SHOW, PROMPT, WRITE, OUTPUT</b>
التهيئة والإسناد	<b>SET, INIT</b>
العملية الحسابية	<b>COMPUTE, CALCULATE, DETERMINE</b>
القرار أو الشرط	<b>TEST, IF/ELSE</b>
الحلقة أو التكرار	<b>WHILE, FOR, REPEAT, LOOP</b>

للتعبير عن الخوارزمية السابقة لإيجاد مجموع عددين باستخدام هذا الأسلوب يمكن أن نكتب كما يلي:

```

1- START
2- READ VALUE1
3- READ VALUE2
4- CALCULATE SUM = VALUE1 + VALUE2
5- WRITE SUM
6- STOP

```

نشير أخيراً إلى أن لغة شبه الشيفرة ليست لغة قياسية وقد تجد أشكالاً مختلفة في التعبير تختلف من مرجع لآخر.

### التعبير عن الخوارزميات باستخدام المخططات التدفقية

في بعض الأحيان يكون من المفيد أثناء وصف الخوارزمية أن نقوم ببناء مخطط تدفقي بسيط يكتب في داخل رموزه أو صناديقه عبارات باللغة الطبيعية أو تعابير رياضية لتلخيص ماينجزه هذا المخطط التدفقي.

**المخطط التدفقي flowchart هو نوع من المخططات (الرسومية أو الرمزية) التي تعبر عن خوارزمية أو عملية ما.** حيث يتم تمثيل كل خطوة في الخوارزمية بواسطة رمز مختلف يتضمن وصفاً مبسطاً لهذه الخطوة. يتم ربط رموز المخطط التدفقي مع بعضها من خلال أسهم لتبين تدفق أو مسار تنفيذ الخطوات.

يقوم المخطط التدفقي بوصف ماهي العمليات (وما هو ترتيبها ) اللازمة لحل مسألة مطروحة.

تستخدم المخططات التدفقية في تحليل، تصميم، توثيق أو إدارة البرامج. وهي ترسم عادة في المراحل الأولى لتصميم البرنامج. كما أن من مزاياها تسهيل التواصل والتفاهم بين المصمم والمبرمج من جهة، والمبرمج والآخرين من جهة ثانية.

تعتمد هذه الطريقة في التعبير عن الخوارزميات على توضيح خطوات تنفيذ الخوارزمية باستخدام أشكال هندسية خاصة وأسهم تصل بينها، إضافة إلى عبارات باللغة الطبيعية، و/أو بتعابير رياضية أو منطقية.

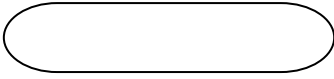
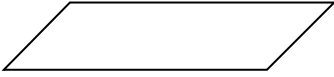

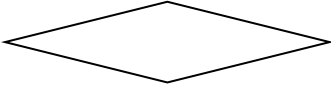
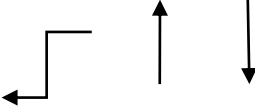
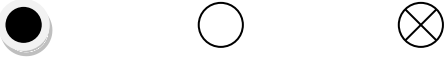
#### من أهم حسنات وفوائد استخدام المخططات التدفقية:

- تسهيل التواصل بين مختلف الأشخاص المعنيين بالمسألة (المصمم، المبرمج، صاحب العمل ...).
- يمكن بمساعدة المخططات التدفقية تحليل المسائل المطروحة بفعالية أكبر.
- التوثيق الملائم للبرنامج.
- تساهم في كتابة شيفرات برمجية فعالة وملائمة للمسألة المطروحة.
- تسهيل عملية تنقيح وصيانة البرامج.

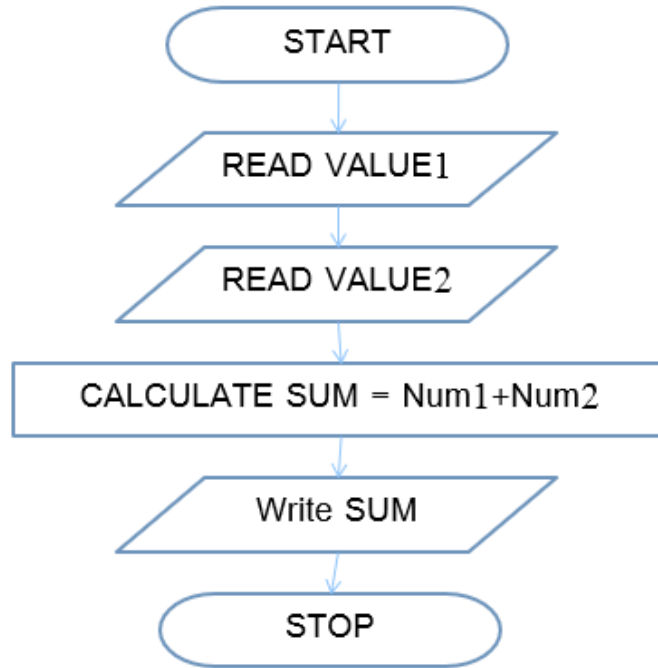
وعلى الرغم من أن المخططات التدفقية تعتبر أداة مفيدة، إلا أنها تواجه بعض المحدودية في التعبير عن جميع المسائل وذلك في الحالات التالية:

- في المسائل ذات الطبيعة المعقدة.
- في حال كثرة التعديلات على المسألة المطروحة التي تتطلب إعادة بناء الخوارزمية من جديد.
- وجود بعض العمليات التي لا يمكن التعبير عنها بواسطة رموز المخططات التدفقية.

نبين فيما يلي قائمة بأهم الرموز التي يمكن استخدامها في بناء مثل هذه المخططات (يكتب بداخل هذه الرموز توصيف للعملية التي تمثلها):

الرمز التخطيطي المستخدم	العملية
	البداية والنهاية
	الإدخال والإخراج
	العملية الحسابية والإسناد
	الاختبار أو القرار
	مسار التدفق
	نقاط التوصيل والربط
إضافة إلى طيف واسع من الرموز الأخرى	

يعبر المخطط التالي عن خوارزمية إيجاد مجموع عددين:



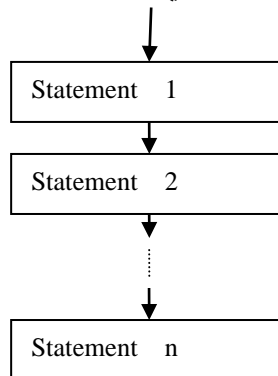
## 6- التحكم بمسار التنفيذ

يمكن تصنيف الخوارزميات من حيث التحكم بمسار التنفيذ إلى نوعين:

- 1- خوارزميات تسلسلية (تتابعية).
- 2- خوارزميات غير تسلسلية (تفرعية)

## الخوارزميات التسلسلية

تستخدم في حل المسائل التي يكون حلها مؤلفاً من تتال محدد للخطوات التي تؤدي إلى النتيجة دون الحاجة لتغيير سياق التنفيذ، تنفذ هذه الخطوات خطوة خطوة حتى الوصول إلى النهاية دون تجاهل أو تكرار لأي من الخطوات الموجودة. يبين الشكل التالي بنية المخطط التتابعي:



### مثال خوارزمية تسلسلية:

احسب المصروف اليومي لطفلك بحسب عمر كل منهما، بمعدل 75 ليرة لكل عام من أعوام عمر الطفل.

#### تنويه هام:

قبل البدء بكتابة أي خوارزمية، تأكد من أن المهمة محددة تمامًا وبالتالي لا بد من البدء بتحليل المسألة عبر طرح مجموعة من الأسئلة ومحاولة الإجابة عنها بشكل واضح ودقيق.

#### أسئلة يجب طرحها:

- ما هي البيانات المعروفة قبل تشغيل البرنامج؟
- ما هي البيانات التي يجب إدخالها من قبل المستخدم؟
- ما هي الحسابات التي سيتم إجراؤها على البيانات؟
- ما هي البيانات التي سيتم إخراجها (المعروضة) للمستخدم؟

#### تحليل المسألة:

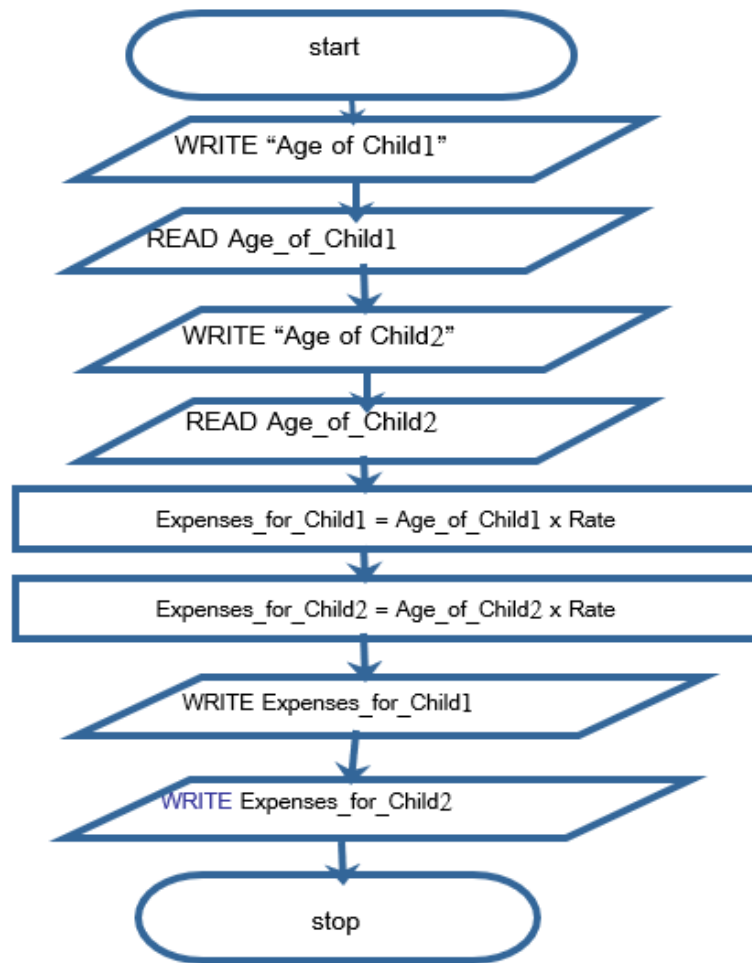
القيم المعروفة	• المعدل 75 ليرة لكل سنة.
المدخلات	• أعمار الأطفال.
الحسابات	• المصروف = العمر x المعدل
المخرجات	• المصروف لكل طفل.

#### التعبير عن الحل باستخدام شبه الشيفرة

- 1- **START**
- 2- **WRITE** "Age of Child1"
- 3- **READ** Age\_of\_Child1
- 4- **WRITE** "Age of Child2"
- 5- **READ** Age\_of\_Child2
- 6- **CALCULATE** Expenses\_for\_Child1 = Age\_of\_Child1 x Rate
- 7- **CALCULATE** Expenses\_for\_Child2 = Age\_of\_Child2 x Rate
- 8- **WRITE** Expenses\_for\_Child1
- 9- **WRITE** Expenses\_for\_Child2
- 10- **STOP**

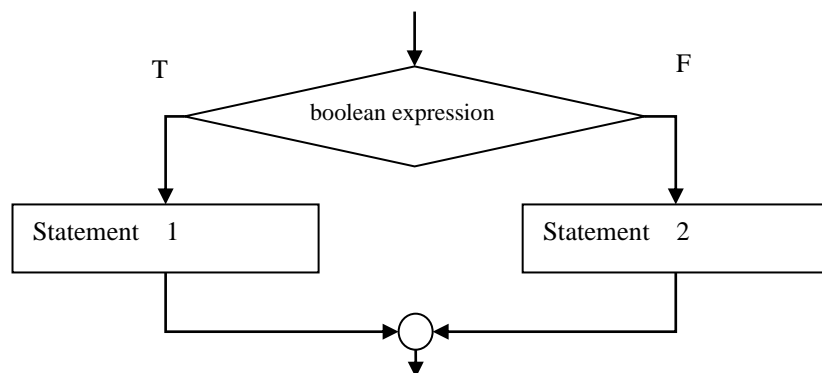
ملاحظة: إن إخراج الرسائل التوضيحية ورسائل التخاطب ليس إلزامياً ولكنه ممارسة جيدة وخاصة أثناء الانتقال من الخوارزمية المصممة إلى كتابة البرنامج.

التعبير عن الحل باستخدام المخططات التدفقية

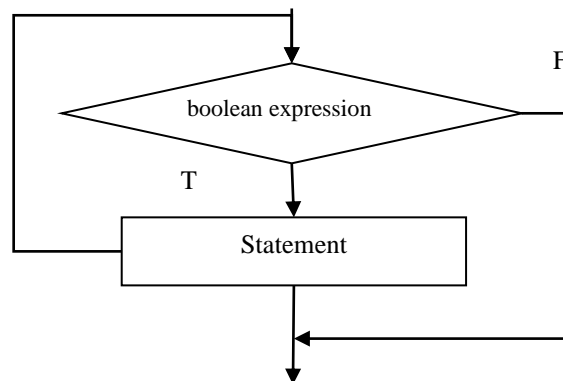


### الخوارزميات غير التسلسلية

ان الأمر ليس دائماً على النحو الوارد في الحالة التسلسلية، فقد يتطلب تنفيذ برنامج أن لا يتم تنفيذ جميع الأوامر وإنما اختيار تنفيذ أوامر وتجاهل تنفيذ أوامر أخرى بحسب ناتج اختبار شروط ما، كما يوضح المخطط التدفقي التالي:



وبالمثل، قد يتطلب تنفيذ برنامج أن لا يكتفى بتنفيذ الأمر لمرة واحدة وإنما تكرر تنفيذه لعدد من المرات بحسب ناتج اختبار شروط ما، كما يوضح المخطط التدفقي التالي:



تدعى المخططات التي تتضمن اختياراً بين تنفيذ أمرين مختلفين بمخططات القرار، أما المخططات التي تتضمن تكراراً لتنفيذ مجموعة من الخطوات لأكثر من مرة باسم المخططات الحلقية.

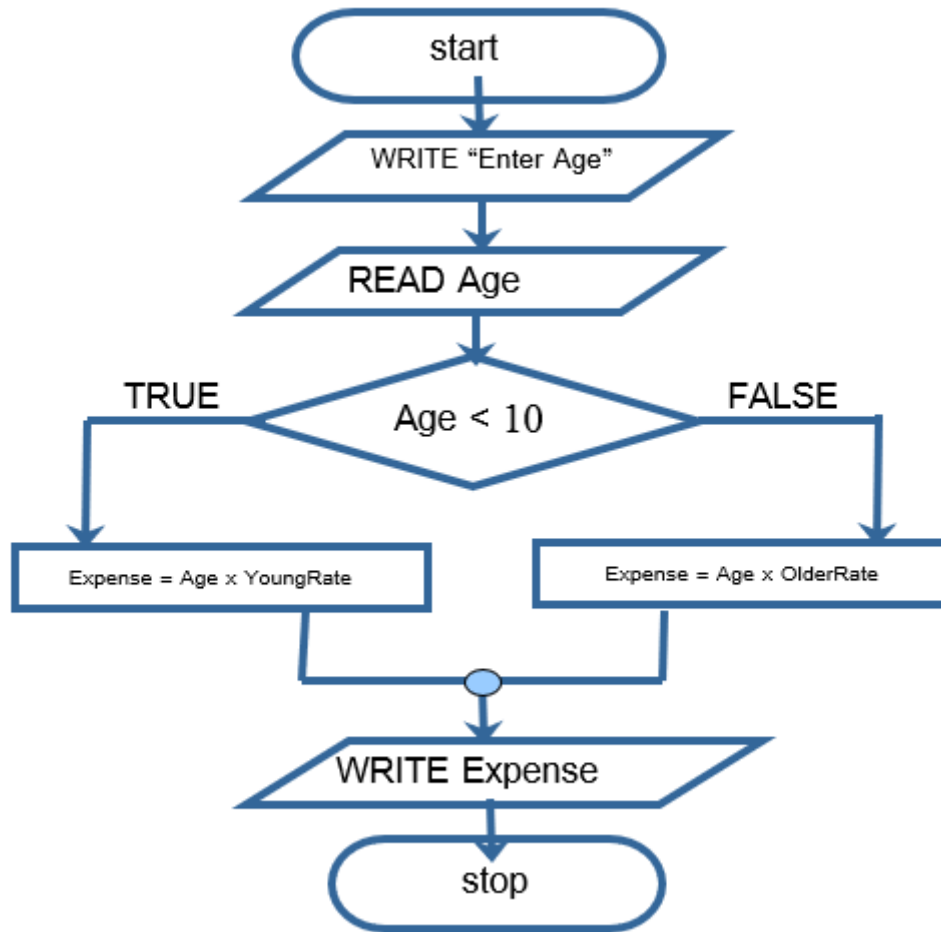
مثال خوارزمية غير تسلسلية (حالة الاختيار أو القرار):

احسب المصروف اليومي للطفل بحسب عمر الطفل، بمعدل 75 ليرة لكل عام من أعوام عمر الطفل إذا كان عمره أقل من 10 سنوات و 100 ليرة إذا كان عمره أكبر أو يساوي 10 سنوات.

التعبير عن الحل باستخدام شبه الشيفرة:

1. **START**
2. **WRITE** "Enter Age"
3. **READ** Age
4. **IF** Age < 10 **THEN**
  - 4-1 **CALCULATE** Expense = Age x YoungRate
- ELSE**
  - 4-1 **CALCULATE** Expense = Age x OlderRate
5. **WRITE** Expense
6. **STOP**

التعبير عن الحل باستخدام المخططات التدفقية



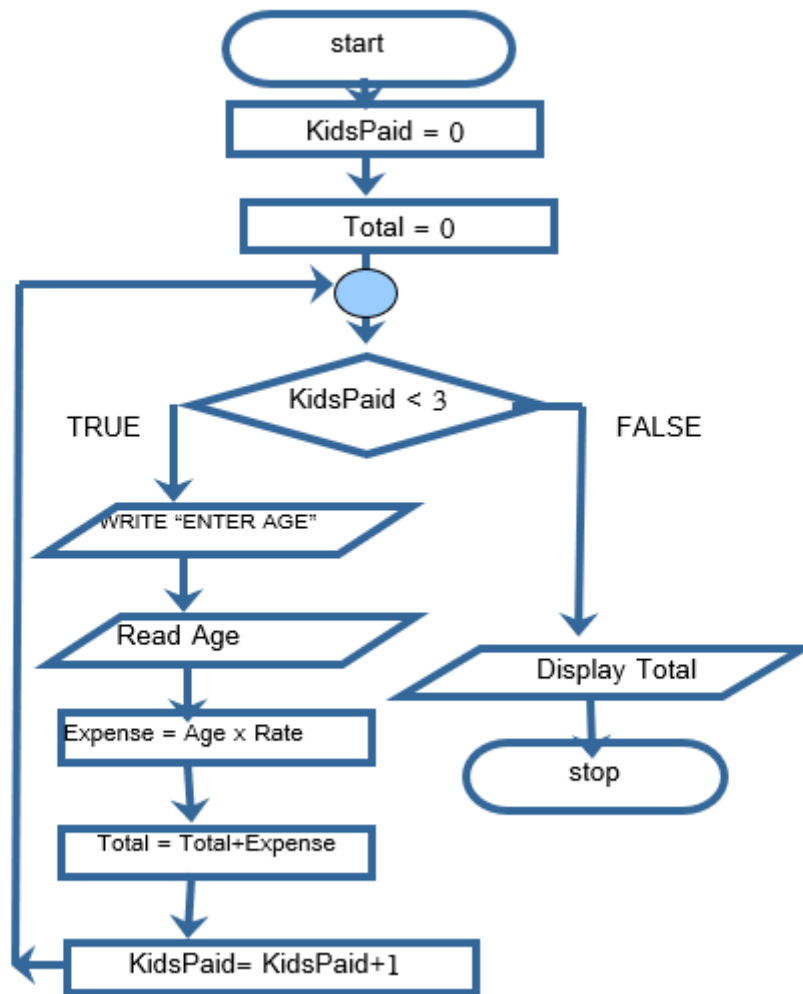
مثال خوارزمية غير تسلسلية (حالة التكرار):

التعبير عن الحل باستخدام شبه الشيفرة:

احسب المصروف اليومي اليومي لثلاثة أطفال بحسب عمر الطفل، بمعدل 100 ليرة لكل عام من أعوام عمر الطفل.

1. START
2. SET KidsPaid = 0
3. SET Total = 0
4. WHILE KidsPaid < 3
  - 4-1 WRITE "ENTER AGE"
  - 4-2 READ Age
  - 4-3 CALCULATE Expense = Age x Rate
  - 4-4 ADD Expense to Total
  - 4-5 SET KidsPaid = KidsPaid + 1
5. WRITE Total
6. STOP

التعبير عن الحل باستخدام المخططات التدفقية:





## 7- تمارين وأنشطة

ملاحظة: يطلب التعبير عن حلول جميع هذه التمارين باستخدام شبه الشيفرة والمخططات التدفقية بعد إجراء تحليل المسألة المطروحة وتحديد القيم المعروفة والمدخلات والحسابات والمخرجات.

### التمرين الأول:

صمم خوارزمية لحساب مجموع الأعداد المحصورة بين 1 و 50.

### التمرين الثاني

صمم خوارزمية لإيجاد العدد الأكبر بين ثلاثة أعداد A,B,C يدخلها المستخدم.

### التمرين الثالث

صمم خوارزمية لإيجاد عاملي عدد صحيح موجب N! حيث أن:

$$N! = 1 * 2 * 3 * ..... * (N-1) * N$$

### التمرين الرابع

صمم خوارزمية لإيجاد العدد الأكبر بين مجموعة من الأعداد.

### التمرين الخامس

صمم خوارزمية حساب المتوسط الحسابي:

$$\text{Average} = \frac{\sum_{i=1}^n x_i}{n}$$

حيث تعبر  $x_i$  عن معدلات طلاب صف من صفوف الجامعة الافتراضية. مع العلم أن عدد طلاب الصف الواحد (المشار إليه بالمتحول n) يُعطى من الدخل، وأن المعدلات محسوبة من 100 علامة وأن المستخدم يقوم بإدخال المعدلات عند تنفيذ البرنامج.

### التمرين السادس

صمم خوارزمية حساب الانحراف المعياري:

$$\text{StandardDeviation} = \frac{\sum_{i=1}^n (x_i - \text{Average})^2}{n}$$

حيث تعبّر  $x_i$  عن معدلات طلاب صف من صفوف الجامعة الافتراضية، ويعبر  $\text{Average}$  عن المتوسط الحسابي للمعدلات. مع العلم أن عدد طلاب الصف الواحد (المُشار إليه بالمتحول  $n$ ) يُعطى من الدخل وأن المعدلات محسوبة من 100 علامة وأن المُستخدم يقوم بإدخال المعدلات عند تنفيذ البرنامج.