

الخوارزميات Algorithm وخرائط المسار Flowchart:

١- الخوارزميات (Algorithms):

كلمة خوارزم تعني الوصف الدقيق لتنفيذ مهمة من المهمات أو حل مسألة من المسائل . واشتقت هذه الكلمة من اسم عالم الرياضيات العربي محمد بن موسى الخوارزمي . وتستخدم كلمة خوارزم على نطاق واسع في علوم الرياضيات والحاسب الآلي.

الخوارزم Algorithm: هي مجموعة من التعليمات (الخطوات) المرتبة ترتيباً منطقياً بشكل تتابعي متسلسل ومنظم لتنفيذ عمليات حسابية أو منطقية أو غيرها.

مثال: اكتب الخوارزم اللازم لحساب المتوسط الحسابي لثلاثة أرقام مدخلة بواسطة المستخدم T_1, T_2, T_3 ؟

Write an Algorithm to calculate an Average of three numbers T_1, T_2, T_3 input by the user?

الحل:

- | | |
|--|---|
| 1- Start | ١- ابدأ |
| 2- Read T_1, T_2, T_3 | ٢- اقرأ قيم الأعداد T_1, T_2, T_3 |
| 3- calculate
$AV = (T_1 + T_2 + T_3) / 3$ | ٣- احسب المتوسط الحسابي للأعداد AV من
المعادلة
$AV = (T_1 + T_2 + T_3) / 3$ |
| 4- Print AV | ٤- اطبع النتيجة |
| 5- Stop | ٥- توقف |

ملاحظات:

- ❖ عند كتابة الخوارزم يجب أن تبدأ الخطوة الأولى بكلمة " ابدأ Start " على أن تنتهي آخر خطوات الخوارزم بكلمة " توقف Stop " ونضع بينهما الخطوات الرئيسية لحل المشكلة.
- ❖ الرموز AV, T_1, T_2, T_3 جميعها رموز اختيارية تسمى بالمتغيرات Variables وهي عبارة عن مساحات يتم حجزها عن طريق المترجم ونظام التشغيل على الذاكرة لاستقبال البيانات المختلفة.




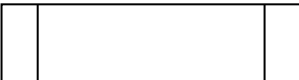
خرائط المسار او التدفق Flowchart:

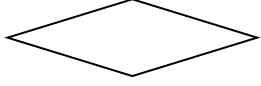
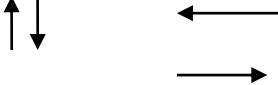
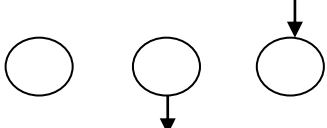
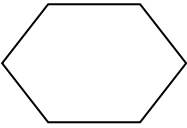

هي عبارة عن مجموعة من الرموز والأشكال الاصطلاحية تستعمل لتوضيح الخطوات الرئيسية لحل مسألة ما بشكل رسومي متفق عليه (ومعلوم) لجميع مبرمجي الحاسب لتبين جميع الخطوات والعلاقات المنطقية للبرنامج.

فوائد استخدام خرائط سير العمليات أو خرائط المسار Flowchart:

1. تعطي صورة متكاملة للخطوات المطلوبة لحل المسألة في ذهن المبرمج بحيث تمكنه من الإحاطة الكاملة بكل أجزاء المسألة من بدايتها وحتى نهايتها.
2. تساعد المبرمج على تشخيص الأخطاء المنطقية التي تقع عادة في البرنامج والتي يعتمد اكتشافها على وضوح التسلسل المنطقي لحل المسألة عن طريق المبرمج.
3. تسير للمبرمج إدخال أي تعديلات في أي جزء من أجزاء المسألة بسرعة وبدون الحاجة لإعادة دراسة المسألة بكاملها من جديد.
4. تساعد المبرمج على متابعة دقة التسلسل خاصة في المسائل التي تكثر فيها الاحتمالات والتفرعات حيث يصبح هذا الأمر شاقا إذا لم يستعن بخرائط المسار.
5. تعتبر خرائط المسار المستعملة في حلول بعض المسائل مرجعا للمسائل المشابهة ومفتاحا لحل مسائل جديدة لها علاقة بالمسائل القديمة المحولة .

الجدول التالي يوضح الرموز والأشكال الاصطلاحية المستخدمة في خرائط المسار Flowchart:

الشكل الاصطلاحي(الرمز)	معنى الرمز
	يستخدم هذا الرمز للدلالة على بداية البرنامج ونهايته (Start / Stop)
	يمثل هذا الرمز كل من عمليتي الإدخال (قراءة البيانات تمهيدا لمعالجتها) والإخراج (عرض النتائج على الشاشة، طباعة، ...) (Input / Output)
	رمز المعالجة، وقد يحتوي هذا الرمز على عملية حسابية أو عملية تخزين (Calculate and Store)
	عملية استدعاء (نداء) لبرنامج فرعي (Call Sub-Routine)

الشكل الاصطلاحي (الرمز)	معنى الرمز
	رمز اتخاذ قرار، ويستخدم هذا الرمز في خطوات المعالجة التي تتطلب إجراء عملية منطقية كالمقارنة أو عملية اختبار Decision
	اتجاه سير البرنامج. تستخدم الأسهم لبيان حركة واتجاه خريطة التدفق. Directions
	نقطة توصيل وربط (Connector).
	تكرار أو دوران (Looping)
	تعليق وإيضاح (Comment)

أنواع خرائط سير العمليات:

١. خرائط سير النظم System Flowchart:

يستخدم هذا النوع من الخرائط عند تصميم الأجهزة الهندسية ، وعند تصميم نظام للمصانع أو الشركات وغيرها، والتي تستعمل أنظمة تحكم ذاتية ، مثل العوامة في خزانات المياه، وإشارات المرور الضوئية، وأجهزة ضبط الضغط ودرجات الحرارة في أبراج تقطير البترول، فتعتبر خرائط المسار هنا بمثابة المخطط الكامل الذي يبين ترتيب وعلاقة ووظيفة كل مرحلة بما قبلها وبما بعدها داخل إطار النظام المتكامل ويمكن تلخيص الدور الذي تقدمه هذه الخرائط بما يأتي:

١. تبين موقع كل خطوة من الخطوات الأخرى المكونة للنظام بحيث يسهل اكتشاف أي خلل

يحدث في النظام كله بمجرد النظر ، مما ييسر عمليات صيانة الأجهزة وبأقل التكاليف.

٢. تسهل إجراء التعديلات التي قد تطرأ مستقبلاً على برنامج النظام في أي جزء منه.

٣. بيان التفصيلات عن المعطيات المطلوب إدخالها إلى النظام.

٤. بيان التفصيلات عن أنواع النتائج المتوقعة أو المطلوبة من البرنامج المعد للنظام.

٥. بيان طرق ربط النظام ببيئة الأنظمة الموجودة في المؤسسة المعنية.

٢. خرائط سير البرنامج Programs Flowchart

ويستعمل هذا النوع من الخرائط لبيان الخطوات الرئيسية التي توضع لحل مسألة ما وذلك بشكل رسوم اصطلاحية تبين العلاقات المنطقية بين سائر خطوات الحل وموقع ووظيفة كل منها في إطار الحل الشامل للمسألة، ويكن تصنيف خرائط سير البرنامج إلى أربعة أنواع رئيسية:

أ- خرائط التتابع البسيط (Simple sequential Flowchart).

ب- الخرائط ذات الفروع (Branched Flowchart).

ج- خرائط الدوران الواحد (simple-Loop Flowchart).

د- خرائط الدورانات المتعددة (Multi-Loop Flowchart).

ويمكن للبرنامج الواحد أن يشمل أكثر من نوع واحد من هذه الأنواع .

أ- خرائط التتابع البسيط (Simple sequential Flowchart):

يتم ترتيب خطوات الحل لهذا النوع من الخرائط بشكل سلسلة مستقيمة من بداية البرنامج حتى نهايته ولا تحتوي على أية تفرعات أو دورانات .

مثال: اكتب الخوارزم للآلزام لآساب مساحة ومحيط الدائرة بمعلومية نصف قطرها ، ثم ارسمي خريطة سيرالبرنامج Flowchart المناظرة له؟

Write an Algorithm and Draw a Flowchart to calculate circumference and area of circle and display result?

الحل:

مساحة الدائرة πR^2

محيط الدائرة $2\pi R$

حيث π = النسبة التقريبية وقيمتها العددية ثابتة وتساوي 3.14 ، R نصف القطر وهو متغير (معطيات)

خريطة سير البرنامج Flowchart	الخوارزم Algorithm	
<pre> graph TD Start([Start]) --> ReadR[/Read R/] ReadR --> PI[PI=3.14] PI --> A["A= PI*R*R"] A --> C["C=2*PI*R"] C --> PrintAC[/Print A, C/] PrintAC --> Stop([Stop]) </pre>	1- Start 2- Read R 3- put $\pi=3.14$ 4- Calculate circle's area from the formula: $A= \pi*R*R$ 5- Calculate circle's circumference from the formula: $C=2*\pi*R$ 6- Print A, C,R 7- Stop	١- ابدأ ٢- اقرأ قيمة R ٣- ضع قيمة $\pi= 3.14$ ٤- احسب مساحة الدائرة A من المعادلة : $A= \pi*R*R$ ٥- احسب محيط الدائرة C من المعادلة: $C=2*\pi*R$ ٦- اطبع قيم كل من A,C,R ٧- توقف

ملاحظات:

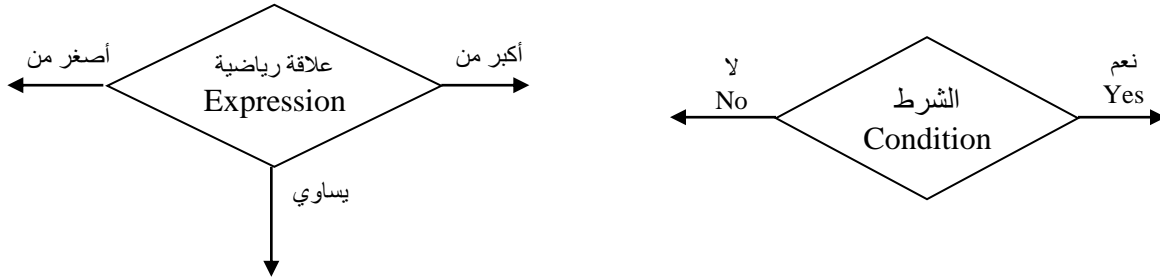
❖ تحديد رمز المساحة A والمحيط C اختياري ولكن من الأفضل أن تكون الرموز معبرة لمجتمويات المتغير فالرمز A يعبر عن المساحة Area ، والرمز C يعبر عن المحيط Circumference.

❖ عملية الضرب داخل خرائط سيرالعمليات يستخدم لها الرمز * أما القسمة فيستخدم لها الرمز /

ب- الخرائط ذات الفروع (Branched Flowchart):

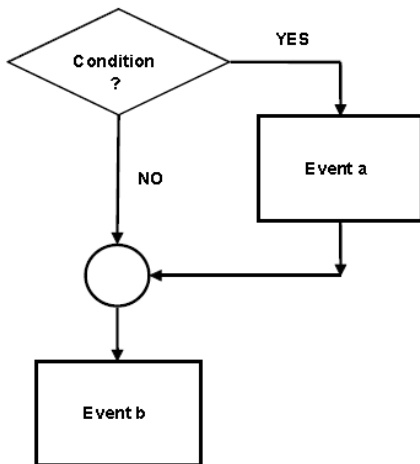
يحدث التفرع داخل البرنامج عند الحاجة لاتخاذ قرار أو مفاضلة بين اختيارين أو أكثر فيسير كل اختيار في طريق مستقل (تفرع) عن الآخر. وهناك نوعان من القرارات يمكن للمبرمج استعمال أحدها حسب المسألة المراد حلها.

أ) قرار ذو تفرعين ب) قرار ذو ثلاثة تفرعات

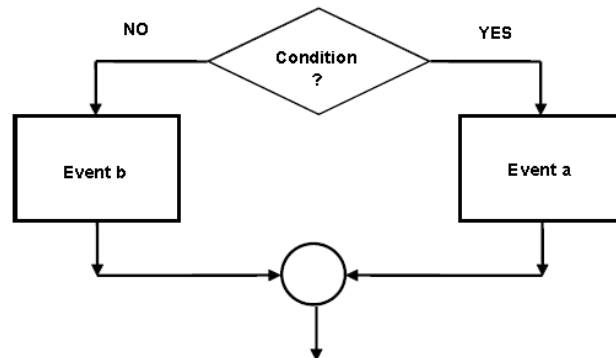


وبشكل عام فإن خرائط التفرع يمكن أن تأخذ إحدى الصورتين التاليتين:

شكل (ب)



شكل (أ)



- ❖ في شكل (أ) يبين أنه إذا كان جواب الشرط "نعم" فإن الحدث التالي في التنفيذ هو الحدث (a) ، أما إذا كان جواب الشرط "لا" فإن الحدث التالي في التنفيذ هو الحدث (b).
- ❖ في شكل (ب) يبين أنه إذا كان جواب الشرط "نعم" فإن الحدث التالي في التنفيذ هو الحدث (a) يليه الحدث (b) ، أما إذا كان جواب الشرط "لا" فإن الحدث التالي في التنفيذ هو الحدث (b) مباشرة.

مثال: يوضح القرار ذو التفرعين

اكتبي الخوارزم اللازم لحساب قيمة الدالة $F(x)$ المعرفه كمايلي

$$F(x) = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$$

، ثم ارسمي خريطة المسار المناظرة له.

الحل:

الخوارزمAlgorithm	خريطة سير البرنامجFlowchart
<p>١- ابدأ</p> <p>٢- اقرأ قيمة X</p> <p>٣- إذا كانت X أكبر من أو تساوي صفر اذهب إلى الخطوة (٤)، وإلا فاذهب إلى الخطوة (٥).</p> <p>٤- احسب قيمة الدالة $F(x)$ من $F(x) = x$ ثم اذهب إلى الخطوة (٦).</p> <p>٥- احسب قيمة الدالة $F(x)$ من $F(x) = -x$.</p> <p>٦- اطبع قيمة $x, F(x)$</p> <p>٧- توقف.</p>	<pre> graph TD Start([Start]) --> ReadX[/Read X/] ReadX --> Decision{X >= 0} Decision -- yes --> Fx_x[F(x) = x] Decision -- No --> Fx_neg[F(x) = -x] Fx_x --> Connector(()) Fx_neg --> Connector Connector --> Print[/Print F(x), x/] Print --> Stop([Stop]) </pre>

ملاحظة: عند تنفيذ الخطوة (٥) فإن الخطوة المنطقية التالية لها هي خطوة الطباعة (الخطوة (٦)) ، ولذلك لم يتم توجيه البرنامج بالعبارة اذهب إلى خطوة (٦) ، كما أشرنا في الخطوة (٤) حيث تم توجيه البرنامج إلى الخطوة (٦) وتجاوز الخطوة (٥) حيث أن الخطوتان (٤) ، (٥) لا تحققان معا.

مثال: يوضح القرار ذو ثلاثة تفرعات.

اكتب الخوارزم و ارسمي خريطة سير العمليات لحساب قيمة W المعطاة من العلاقة التالية

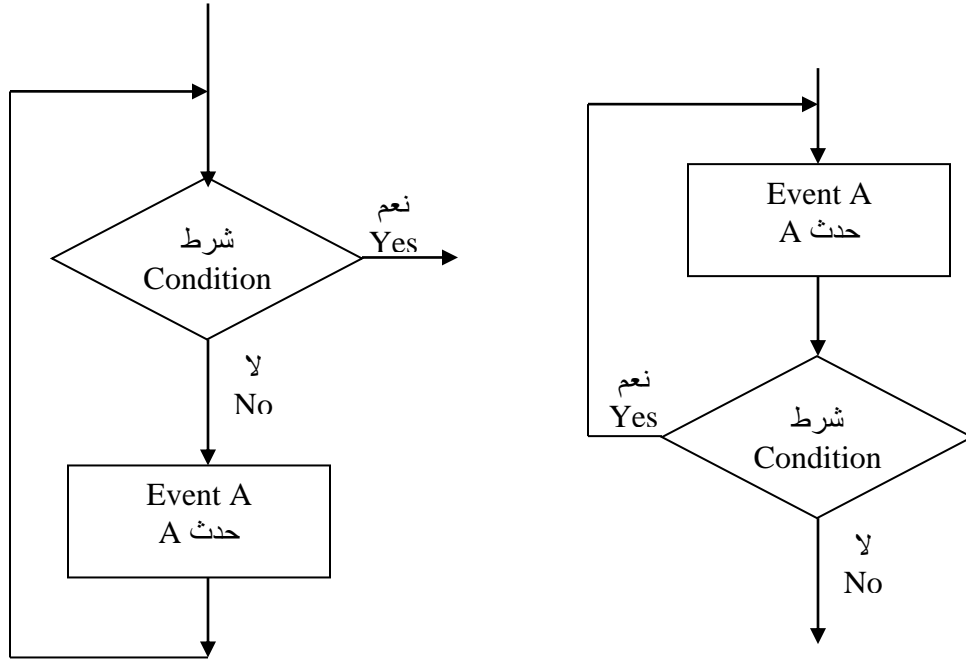
$$W = \begin{cases} X + 1, & X > 0 \\ \sin(X) + 5, & X = 0 \\ 2X - 1, & X < 0 \end{cases}$$

الحل:

Flowchart خريطة سير البرنامج	الخوارزم Algorithm	
<pre> graph TD Start([Start]) --> ReadX[/Read X/] ReadX --> X{X} X -- "< 0" --> W1[W=2*X-1] X -- "= 0" --> W2[W=sin(X)+5] X -- "> 0" --> W3[W=X+1] W1 --> Merge(()) W2 --> Merge W3 --> Merge Merge --> PrintWX[/Print W,X/] PrintWX --> Stop([Stop]) </pre>	1- Start	١ - ابدأ
	2- Read x	٢ - اقرأ قيمة X
	3- If x greater than Zero then go to step (4), If x equal to Zero then go to step (5), If x less than Zero then go to step (6).	٣ - إذا كانت X أكبر من صفر اذهب إلى الخطوة (٤). وإذا كانت X تساوي صفر فذهب إلى الخطوة (٥) أما إذا كانت X أقل من صفر اذهب إلى خطوة (٦).
	4- calculate W =X+1, and then go to step (7)	٤ - احسب قيمة W من المعادلة: W =X+1 ثم اذهب إلى خطوة (٧).
	5- calculate W =sin(X)+5, and then go to step (7)	٥ - احسب قيمة W من المعادلة: W =sin(X)+5 ، ثم اذهب إلى خطوة (٧).
	6- calculate W =2*X-1, and then go to step (7)	٦ - احسب قيمة W من المعادلة: W =2*X-1 ، ثم اذهب إلى خطوة (٧).
	7- print W , x	٧ - اطبع قيم كل من X, W
	8- Stop.	٨ - توقف.

ج- خرائط الدوران الواحد (simple-Loop Flowchart):

نحتاج هذا النوع من الخرائط لإعادة عملية أو مجموعة من العمليات في البرنامج عددا محدودا أو غير محدود من المرات ويكون الشكل العام لمثل هذه الخرائط كما يلي:



وقد سميت هذه الخرائط بخرائط الدوران الواحد لأنها تستعمل حلقة واحدة للدوران ، وتسمى أحيانا بخرائط الدوران البسيط.

مثال:

ارسمي خريطة المسار لإيجاد مجموع m من الأعداد الحقيقية (X_1, X_2, \dots, X_m)

الحل:

النتيجة المطلوبة هي مجموع الأعداد T حيث أن :

$$T = \sum_{i=1}^m (X_i)$$

خطوات الحل يمكن أن تسير حسابيا على النحو:

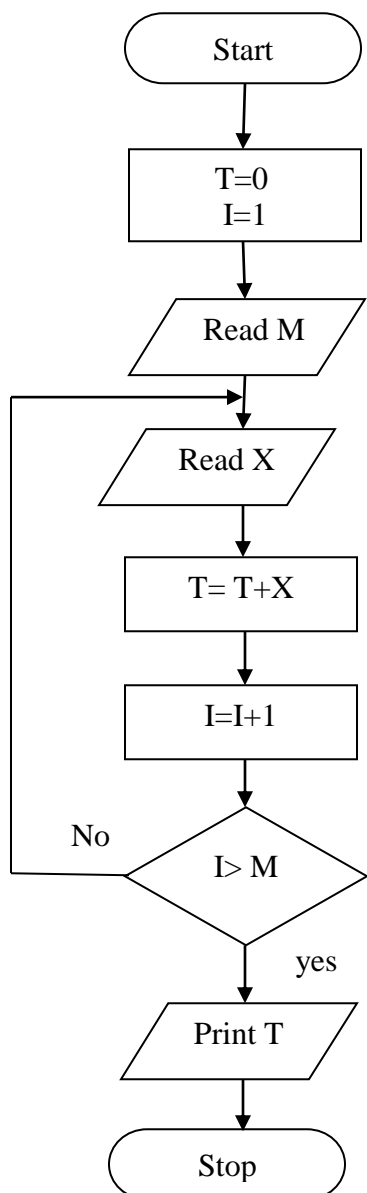
$$T_0=0, \quad T_1=T_0+X_1, \quad T_2=T_1+X_2$$

$$T_m=T_{m-1}+X_m=X_1+X_2+\dots+X_{m-1}+X_m$$

ونموذج الحل هذا يمكن أن يختصر بنموذج مكافئ هو

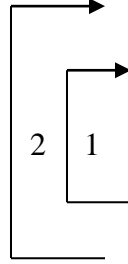
$$T = T + X_i, \quad i=1,2,\dots,m$$

على أن تكون القيمة الأولى للمجموع T هي $T=0$



د- خرائط الدورانات المتعددة (Multi-Loop Flowchart):

في هذه الحالة تكون الدورانات داخل بعضها البعض لا تتقاطع ، فإذا كان لدينا دورانان من هذا النوع كما في الشكل التالي فيسمى الدوران رقم (١) دورانا داخليا Inner Loop بينما الدوران رقم (٢) دورانا خارجيا outer Loop ، ويتم التنسيق بين مثل هذين الدورانين ، بحيث تكون أولوية التنفيذ للدوران الداخلي.



وقد سميت هذه الخرائط بخرائط الدورانات المتعددة لأنها تستعمل أكثر من حلقة دوران واحدة، وقد تسمى أحيانا بخرائط الدورانات المتداخلة أو المترابطة أو الضمنية .

مثال:

ارسمي خريطة المسار لإيجاد معدل درجات كل طالب إذا كان عدد المواد لكل طالب N ، وعدد طلاب الفرقة M ؟

ملاحظات:

سوف نستخدم المتغير S كمخزن لتجميع درجات الطالب، والمتغير G يعبر عن درجة الطالب في المادة ،

AV يمثل متوسط الدرجات، N عدد المواد المقررة لكل

طالب ، M عدد الطلاب في كل فرقة ، J عداد لحصر

عدد المواد N ، I عداد لحصر عدد الطلاب M .

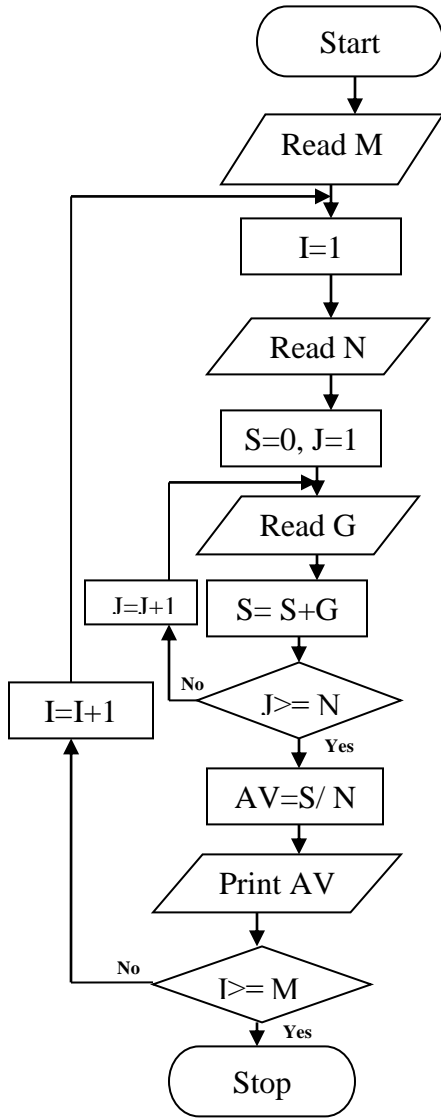
يلاحظ من الشكل أن : الدوران الداخلي يقوم بقراءة وجمع

درجات المواد لكل طالب على حده ، بينما الدوران الخارجي يقوم

بحساب متوسط درجات الطالب الكلية ثم طباعته قبل البدء في إدخال

درجات طالب آخر ، العداد J للخروج من الدوران الداخلي ، العداد I

للخروج من الدوران الخارجي .



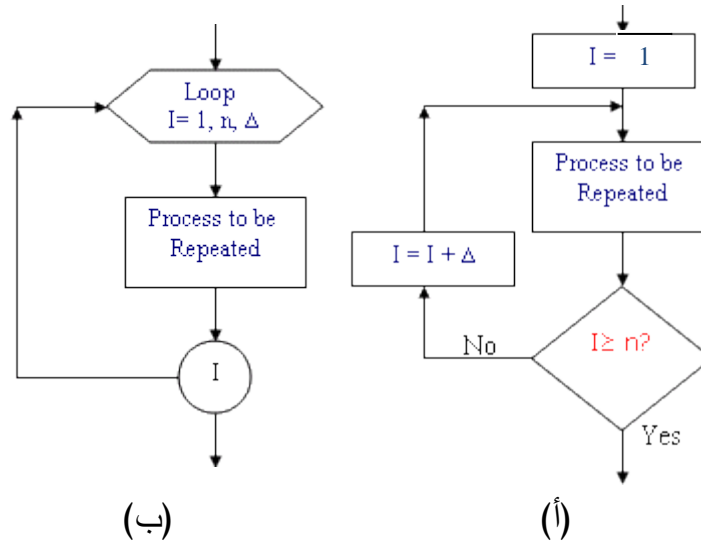
صيغة الدوران باستعمال الشكل الاصطلاحي (الدوران التكراري Loop):

في الفقرتين السابقتين تعلمنا مفهوم الدوران البسيط والدورات المتعددة (المتداخلة) ويمكننا الآن استخدام الشكل الاصطلاحي (الدوران التكراري Loop). نلاحظ من المثال السابق أننا نحتاج إلى العناصر الآتية:

❖ القيمة الأولية للعداد I (هنا $I=1$).

❖ القيمة النهائية للعداد I (هنا $I=N$).

❖ قيمة الزيادة عند نهاية كل دورة Δ .

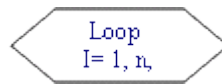


نلاحظ في الشكل السابق الجزء (أ) أن إجراءات الدوران كانت تتم طبقاً للخطوات الآتية والمفصلة من قبل المبرمج:

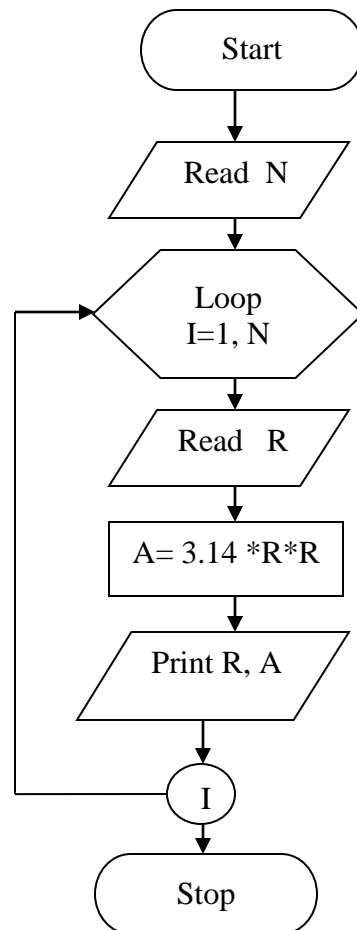
١. أعط العداد I قيمة أولية.
٢. أتم الإجراءات المطلوب إعادتها.
٣. اتخاذ قرار: إذا كانت قيمة العداد I وصلت إلى القيمة النهائية N فخرج إلى الخطوة (٦) في البرنامج وإلا فإذهب إلى الخطوة (٤).
٤. زد I بمقدار Δ .
٥. عد إلى الخطوة (٢).
٦. أكمل ما تبقى من البرنامج.

يمكننا استبدال الخطوات المفصلة (١، ٣، ٤، ٥) في الشكل (أ) بخطوة مجملة واحدة مبينة في الشكل الاصطلاحي للدوران (ب) حيث تنفذ هذه العملية بصورة أوتوماتيكية من قبل الحاسب، وهذا من شأنه تسهيل عملية البرمجة واختصار عدد التعليمات في البرنامج وتجنب بعض الأخطاء.

ملحوظة: تعتبر قيمة Δ تساوي ١ دائماً إذا لم تعط قيمة أخرى بخلاف ذلك، وفي حالة عدم ذكر قيمة Δ يصبح الشكل الاصطلاحي (الدوران المتكرر) الوارد في الشكل (ب) كما يلي حيث تكون قيمة Δ تساوي ١ وبصورة أوتوماتيكية.



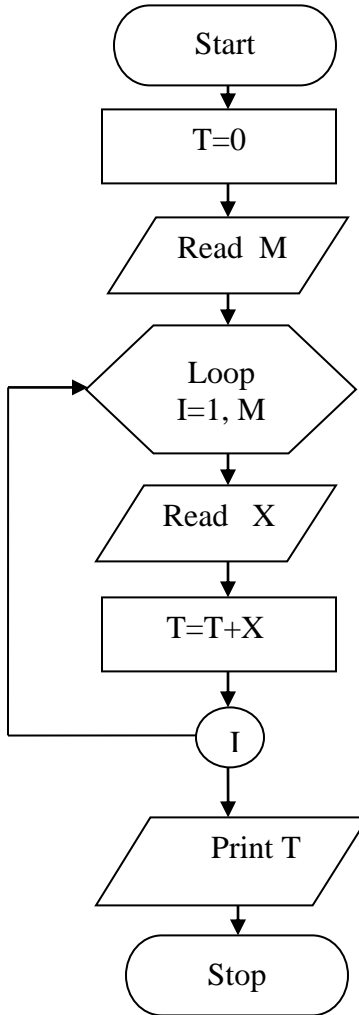
مثال: ارسمي خريطة المسار لإيجاد مساحة N من الدوائر باستخدام الشكل الاصطلاحي للدوران.



الحل:

مثال: ارسمي خريطة المسار لإيجاد مجموع M من الأعداد الحقيقية X_1, X_2, \dots, X_m

الحل:



التطور التاريخي الحاسب الالى

بدأ تطور الحاسب منذ القدم مع بداية حاجة الناس ورغبتهم للعد والإحصاء وقد كان بداية هذا التطور على يد عالمين وهما :

١- هواردايكن ١٩٤٤-١٣٧ Aiken Howard : تعاون هذا العالم من جامعة هارفارد مع شركة IBM لاختراع جهاز حاسب ميكانيكي كهربائي .

٢- جون ماشلي John Mauchly وجون إكرت 1946 – 1949 John Eckert قام هذان العالمان من جامعة بنسلفانيا بتطوير الحاسب لتزويد سعة تخزينه للمعلومات وذلك لاستخدامه في الأغراض العسكرية للجيش الأمريكي.

وقد تدرج تطوير الحاسب بتغير اسمه وشكله والمهام التي يؤديها عبر الزمن كما يوضح العرض التالي :

الأباكس : وهي آلة لميكنة العمليات الحسابية وتستخدم في تعليم الأطفال طريقة العد.

آلة نابير الخشبية: تستخدم هذه الآلة لإجراء عمليات الضرب والقسمة المطولة .

آلة الجمع لباسكال : وهي آلة جمع ميكانيكية .

المسطرة الحاسبة : هي آلة يمكن عن طريقها إجراء مختلف العمليات الحسابية المعقدة .

حاسبة لينتر : هي آلة تستطيع تنفيذ عمليات الضرب والقسمة واستخراج الجذور التربيعية.

آلة الفروق لباباخ : هي آلة تقوم باستخراج الجداول اللوغارتمية بدقة كبيرة .

آلة الجمع الطابعة : وهي آلة تقوم بطبع ناتج العملية الحسابية على شريط من الورق .

ماكينة هوليرت للتبويب : وهي أول آلة كهروميكانيكية ، وتعمل بالبطاقات المثقبة.

الجيل الاول: ١٩٥١-١٩٥٩م

جهاز UNTVAC ١٩٥١م وهو اول جهاز استخدم في الاغراض العامة وفي عام ١٩٥٣م صنعت IBM 701 وهو يتكون من الصمامات الالكترونية(الصمامات المفرغة).

خصائص الجيل الاول:

- الصمامات الالكترونية
- البرمجة بلغة الآلة.
- بطيئة
- الحجم كبير

الجيل الثاني: ١٩٥٩-١٩٦٥م

الترانزستور - IBM 1401

خصائص الجيل الثاني:

- الترانزستور
- ظهور الاجهزة صغيرة الحجم بسبب صغر الترانزستور
- السرعة
- استخدام ذاكرة القلوب الممغنطة
- استخدام أنظمة الحكم في الإدخال والإخراج
- استخدام لغات عليا (كوبول وفورتران)

الجيل الثالث: ١٩٦٥-١٩٧٢م

Integrated Circuits الدوائر المتكاملة IBM 360

خصائص الجيل الثالث:

- الدوائر المتكاملة
- السرعة العالية وإمكانية التخزين
- استخدام نظام المشاركة بالوقت (Time Sharing Systems)
- تطبيق نظم الشبكات للحاسبات الآلية Computer Network يمكن ربط الشركات بفروعها

الجيل الرابع: ١٩٧٢-١٩٨٠م

في عام ١٩٧٢م ظهر الحاسب من نوع IBM 370- Burroughs 7000

خصائص الجيل الرابع:

- زيادة طاقة وحدات الإدخال والإخراج
- الدوائر المتكاملة ذات الشرائح العجيبة وهي ذات قدرة اعلى وسرعة فائقة وتعمل لفترات طويلة.
- إمكانية تشغيل أكثر من برنامج في وقت واحد حسب ترتيبها الذي وضعت به تلك البرامج الماد تنفيذها بوحدات الدخول.

الجيل الخامس: ١٩٨٠م-

في عام ١٩٨٠م ظهر PC حاسب شخصي صغير وأيضا الحواسيب الكبيرة من نوع UNIVAC 1100 IBM 4331

خصائص الجيل الخامس:

- تطبيق ما يسمى بإدارة نظم المعلومات الالكترونية
- صغر الحجم
- انخفاض التكلفة وزيادة الكفاءة
- السرعة العالية

علاقة الحاسب الالى ببقية التخصصات

الحاسب الالى وعلم الفيزياء

قد يسأل البعض منا ما علاقة الحاسب الالى بعلم الفيزياء ؟ انها علاقة كبيرة فمواد الحاسب الالى وفكرة عملها اساسا مأخوذ من هذا العلم الكبير ففي علم الفيزياء يوجد فرع اسمه اشباه الموصلات فكما نعلم ان المواد ثلاث انواع موصلة وعازلة وشبه موصلة وهي التي تكون مقاومتها موصلة اي ليست موصلة ولا عازلة والمواد شبه الموصلة تستخدم في صناعة مكونات الحاسب الالى فنرى الميكروبروسيسور وهو عقل الحاسب الالى مصنوع من مواد شبه موصلة مثل مادة السيلكون وصناعة الترانزستورات الموجودة في الميكروبروسيسور .

الحاسب الالى وعلم الرياضيات

طبعاً كلمة حاسب مأخوذة من الحسابات او الارقام وعلم الرياضيات فنلاحظ في الحاسب الالى يوجد المنطق الرياضي والاستنتاج والمصفوفات والمتسلسلات والمحددات وهذا هو علم الجبر فرع من علم الرياضيات ، ونلاحظ ايضا تطبيقات الاحتمالات في البرامج وايضا مبادئ التحليل العددي وحلول التفاضل والتكامل والمعادلات التفاضلية كل هذا نستخدمه في البرامج في الحاسب الالى ونلاحظ ايضا نظرية الخوارزميات وكيفية تحليل المشكلة البرمجية وكثير وكثيرفالبرمجة كما ستلاحظون بها اشياء كثيرة رياضية ولكن سهلة جدا ليس كما تتصورون ...

تطبيقات الحاسب الالى موجودة في جميع العلوم

طبعا كلنا نعلم دخول الحاسب الالى كل المجالات والعلوم والابحاث منها :-

١- الطب

دخل الحاسب مجال الطب فالاجهزة التي نشاهدها سواء اجهزة العمليات او الاجهزة التلفزيونية المستخدمة في الاشعة وجميع الاجهزة الطبية مبنية علي الحاسبات وايضا كشف الDNA هناك مشروع كامل يحلل هذا Human Genome Project

٢- الهندسة

فهناك برامج كثيرة يستعملها المهندسون مثل برنامج Auto cad ويستخدم في التخطيط والرسم الهندسي.

٣- الاتصالات

الحاسب الالى ايضا يستخدم في الاتصالات واجهزة الشبكات .

٤- الفلك

اجهزة التلسكوبات والاقمار الصناعية والكشف عن الكواكب كلها دخل فيها الحاسب الالى وتطور هذا العلم بالحاسب الالى

٥- الانترنت

الشبكة العنكبوتية اصبحت ملايين الاجهزة يمكنها الاتصال معا في شبكة واحدة وتحت بروتوكولات معينة واصبح بإمكان اي شخص ان يتعرف علي كل العالم وهو موجود في البيت فاصبح العالم عبارة عن قرية صغيرة.

٦- مجالات اخرى

كما نعلم جهاز ATM وهو الجهاز الذي نراه في البنوك لنضع فيه Master Card الذي نستعملها لسحب النقود هذه الالة ما هي الا جهاز به برنامج مكتوب بلغة معينة وينفذ اوامر معينة .