بسم الله الرحمن الرحيم

المرشد في البرمجة

طلال حسن أمين حسين

قسم علوم الحاسوب

جامعة أم درمان الاسلامية - السودان

7/29/2012

الإهداء

اهدي هذا الكتاب الى والديّ واتمنى لها دوام الصحة والعافية وأسأل الله ان يحفظها ويغفر لها ويرحمها ويدخلها فسيح جناته

والى والدي العزيز واتمنى له دوام الصحة والعافية واسأل اللهم ان يغفر له ويرحمه ويدخله فسيح جناته

كما أهدي هذا الكتاب الى أخويّ وأصدقائي وكل طالب علم وأتمنى ان يستفيدو منه

المقدمة:

من الملاحظ للعيان التطور المتسارع في تقنيات الحاسب اللآلي والبرمجيات في جميع مجالات الحياة المختلفة فإن مجاراة هذا الواقع اصبح من أولى أولوياتنا ونجد الاستفادة من تلك التقنيات وتطويعها في حل المشاكل التي تواجهنا والاستفادة القصوى من كل الأدوات المتاحة يجعلنا نقترب من ذلك التطور.

عملية حل المشاكل تتمثل في برمجة برمجيات لإنجاز حلولاً لتلك المشاكل وهذه البرمجة لها قواعد لو اتبعت نحصل على منتج برمجي متكامل من دون اخطاء ولها ايضاً وسائل وأدوات متعددة وفي هذا الكتاب سوف نتطرق الى تلك الادوات والوسائل المهمة حتى يتسنى لنا حل اي مشكلة برمجية بصورة قياسية وصورة مبسطة وسهلة.

دورة حياة البرنامج:

العتاديات هي الاشياء المحسوسة في نظام الحاسوب مثل الماوس و لوحة المفاتيح وغيرها اما البرمجيات هي مجموعة من التعليمات او الأوامر التي تدعى بالبرامج أو المشاريع والتي تقوم بقيادة العتاديات.البرامج تكتب لحل مشاكل او إنجاز مهام على الحاسوب. المبرمجين يقومون بترجمة الحلول للمشاكل او المهام المطلوب إنجازها بلغة يفهمها الحاسوب. لذلك عند كتابة البرامج يجب ان نضع في الاعتبار ان الحاسوب سيقوم فقط بما طلب منه من تعليمات. لذلك يجب علينا ان نكون حزرين و شاملين للتعليمات التي نريد تنفيذها.

البرمجة (مهام على الحاسوب):

البرمجة هي مهام على الحاسوب تمر بخطوات متعددة منها ما يأتي

- أولاً : تحديد ما هي المجرجات Outputs التي يتوقع خروجها من البرنامج و ما المهمة التي يجب ان تنجز.
- ثانياً : تعريف البيانات Identifiers و المدخلات لاهميتها في تحديد المخرجات.

❖ ثالثاً : تحديد كيف تتم معالجة المدخلات Processing للحصول على المخرجات المطلوبة ذلك لتحديد ما هي الصيغة او الطريقة التي يمكن ان تستخدم للحصول على المخرجات المطلوبة.

منهج حل المشكلة البرمجية هو نفس المنهج الذي يستخدم في حل المسائل الجبرية على سبيل المثال : إذا اردنا معرفة سرعة سيارة تقطع مسافة 50 ميل في ساعتين.

الحل :

أولاً : نحدد نوع الإجابة المطلوبة وهي يجب ان تكون رقم يعطي معدل السرعة بالميل لكل ساعة.

ثانياً : البيانات المطلوبة للحصول على الإجابة وهي المسافة والزمن

ثالثاً : الطريقة او المعالجة التي تتم على البيانات للحصول على مخرجات وهي

السرعة =المسافة ÷ الزمن

رابعاً : بعد إجراء المعالجة تكون الاجابة

معدل السرعة = 50 ميل÷2 ساعة

∴ معدل سرعة السيارة = 25 /الساعةميل

التمثيل التصوري لحل اي مشكلة بصورة عامة يقوم على الأتي:

اولاً نحدد ما الشكل الذي يجب ان تكون المخرجات عليه ثم المدخلات المطلوبة من ثم معالجة المدخلات للحصول على المخرجات المطلوبة.

تخطيط البرنامج :

النخطيط لتصميم برنامج يعتبر بمثابة الوصفة التي بإتباعها يمكننا بناء برنامج خالي من الاخطاء ومحتويات هذا البرنامج يتم تحديدها بماهية مخرجات البرنامج اي ما الذي يتوقع ان يقوم البرنامج بإخراجه وعن طريق هذه المخرجات نقوم بتحديد المدخلات المناسبة وإجراء المعالجة المناسبة عليها للحصول على المخرجات المطلوبة. عملية التخطيط للبرنامج قبل الشروع في كتابته تعمل على تقليل عدد الاخطاء التي يمكن ان تحدث في البرنامج فإنه إذا قمنا ببناء او تتطوير برنامج من دون تخطيط مسبق فكاننا نقوم ببناء جسر او مصنع دون خطة مفصلة.

معظم المبرمجين خصوصاً الطلاب في بدايات لغات البرمجة يحاولون كتابة البرامج من دون تخطيط مسبق ولكن تظهر اهمية التخطيط في حل المشاكل (البرامج) المعقدة ففيها يجب التخطيط للبرنامج. اذا قمنا بالتخطيط للبرنامج فإننا نقضي وقتا قليلاً في كتابته.

المبرمجين المهرة يقومون بوضع خطط لبرامجهم بإستخدام خطوات متسلسلة تدعى بدورة حياة البرنامج Program Development Cycle حيث يتم توضيح كل عملية في البرنامج خطوة خطوة وذلك للإستغلال الامثل للوقت والمساعدة في تصميم برامج خالية من الاخطاء تعطى المخرجات المطلوبة وفي ما يأتي توضيح للخطوات التي يمر بها البرنامج:

1. **التحليل**: تعريف المشكلة.

   التأكد التام من فهم البرنامج والمخرجات المطلوبة وأن تكون هنالك رؤية واضحة بما هية البيانات (المدخلات) المعطاه والعلاقة بين المدخلات والمخرجات المطلوبة.

2. **التصميم**: خطة حل المشكلة.

   وجود تسلسل منطقي بخطوات واضحة لحل المشكلة وذلك يدعى بالخوارزمية ,كل التفاصيل والخطوات يجب ان توضح في الخوارزمية. في القسم التالي سوف نناقش ثلاث طرق معروفة لتطوير تخطيط منطقي : flowcharts المخططات الانسيابية, pseudo-code الكود التقريبي (الزائف) و top-down charts التخطيط من اعلى لأسفل.

هذه الادوات تساعد المبرمج في تقسيم المشكلة الى سلاسل من المهام الصغيرة التي يمكن ان ينجزها الحسوب لحل المشكلة.

التخطيط أيضا يتضمن استخدام بيانات تمثيلية لاختبار الخوارزمية منطقياَ وضمان ان الخوارزمية صحيحة.

3. **إختيار واجهة المستخدم:** إختيار الكائِنات (text boxes صناديق نصوص, command buttons أزرار الأوامر ..الخ).

ونحدد ايضا كيف يتم تضمين المدخلات وكيف يتم عرض المخرجات ثم بعد ذلك ننشئ كإنات تستقبل المدخلات وتعرض المخرجات وايضاَ ننشئ ازرار command buttons للسماح للمستخدم بالتحكم في البرنامج.

4. **كتابة الكود:** ترجمة الخوارزمية الى لغة برمجة.

خلال هذه المرحلة البرنامج يكتب بواسطة لغة الفيجوال بيزك والمبرمج يقوم با إستخدام الخوارزمية في الخطوة او المرحلة 2 (مرحلة التصميم) .

5. **الإختبار والتصحيح:** تحديد وإزالة اي اخطاء في البرنامج.

الإختبار: هوعملية البحث عن الاخطاء في البرنامج **والتصحيح:** هو عملية تصحيح الاخطاء التي وجدت (الخطاء في البرنامج يدعى bug) عند كتابة البرنامج ال Visual Basic تعطي أخطاء محددة من الأخطاء وعند تنفيذ البرنامج ايضا هنالك اخطاء محددة مع ذلك فإن معظم الأخطاء بسبب أخطاء في كتابة البرنامج او خطاء في الخوارزمية او نتيجة لعدم الاستخدام الصحيح لقواعد Visual Basic والتي يتم تصحيحها بعناية بواسطة Detective Work. ومن أمثلة الأخطاء إستخدام عملية الاضافة عندما تكون عملية الضرب مسبقاً.

6. **التوثيق الكامل:** تنظيم جميع المكونات التي تصف البرنامج

التوثيف الغرض منه السماع لشخص اخر او مبرمج اخر في الزمن اللاحق مستقبلاً لفهم البرنامج. التوثيق الداخلي Internal Documentation عبارة عن جمل او عبارات لا يتم تنفيذها في البرنامج ولكنها تشير الى الغرض او توضيح او شرح لعمل جزء من البرنامج. في البرامج التجارية التوثيق يحتو ايضا على دليل الاستخدام. ومن الانواع الخرى للتوثيق ايضاً flowchart المخطط الانسيابي, Pseudocode الكود التقريبي للبرنامج و top-down charts. المخططات من اعلى لأسفل التي استخدمت في بناء البرنامج مسبقاَ. بالرقم من ان

التوثيق يعتبر اخر مرحلة في دورة حياة البرنامج إلا انه يجب ان يتم في مرحلة كتابة كود البرنامج.

أدوات البرمجة:

في هذا القسم سنناقش بعض الخوارزميات المحددة وتطوير ثلاث ادوات تستخدم لتحويل الخوارزميات الى برامج في الحاسوب وهي flowcharts, pseudocode, hierarchy charts.

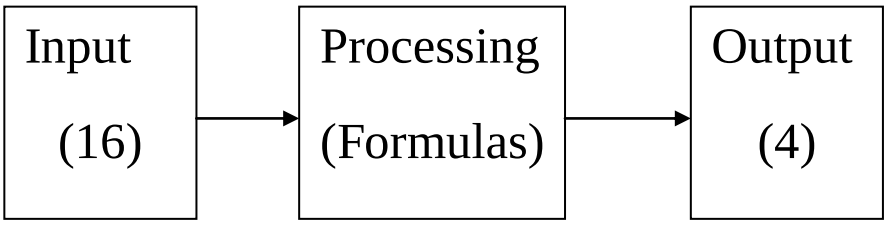
استخدام الخوارزميات يواجهنا كل يوم في اتخاز القرارات وانجاز المهام. وعلى سبيل المثال عندما نريد ارسال رسالة يجب ان نحدد تكلفة أجرة الرسالة على البريد التي سوف توضع في الظرف وهنالك قانون في البريد يقول ان الابهام يستخدم لختم خمس أو اقل ورقات في الظرف الخوارزمية التالية توضح انجاز هذه المهمة.

1. أطلب عدد الاوراق (Input)
2. قسمت عدد الاوراق على خمسة (Processing)
3. قم بتقريب باقي القسمة الى الرقم التالي ان وجد (Processing)
4. رد بعدد الاختام (Output)

الخوارزمية السابقة تأخذ عدد الاوراق كمدخلات وتعالج البيانات ثم تنتج عدد الاختام المطلوبة كمخرجات ويمكننا إختبار الخوارزمية برسالة بها 16 ورقة كما يلي:

1. ادخل عدد الاوراق =16
2. قسمت 16 على 5 = 3.2
3. قرب 3.2 الى 4
4. رد بأن عدد الاختام هو 4

وهذه الخوارزمية تعتبر مثال لحل مشكلة ويمكن تصورها كالأتي:



وكما ذكرنا سابقاً فان هنالك ثلاث ادوات معروفة تستخدم في مرحلة تصميم البرنامج وفي ما يلى سنوضح كل اداة:

- المخططات الإنسيابية Flowcharts: هي عبارة عن تصور تخطيطي بالخطوات المنطقية لتنفيذ مهمة ما ويتم توضويح كل خطوة وعلاقتها مع الاخرى.
- الكود التقريبي Pseudocode: في هذه الاداة تتم كتابة البرنامج بالانجليزية مع بعض مصطلحات لغة البرمجة لإختصار المشكلة او المهمة.
- المخططات المتدرجة(الهرمية) Hierarchy charts: هذه الأداة بها يتم توضيح علاقة الاجزاء المختلفة من البرنامج مع بعضها البعض.

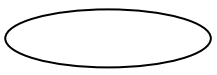
مخططات التدفق:

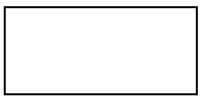
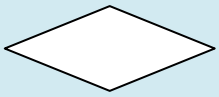
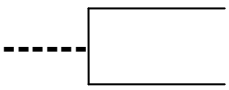
هي مخططات تحتوي على رموز هندسية خاصة متصلة بأسهم ومع كل رمز جملة توضح النشاط الذي سينجز في الخطوة. وكل رمز يوضح نوع العملية التي سوف تحدث وعلى سبيل المثال فإن متوازي الاضلاع يشير الى عملية إدخال او إخراج. الأسهم تقوم بربط الرموز وتدعى بخطوط التدفق flowline وهي توضح الانسياب والتعاقب الذي يتم في تنفيذ الخطوات.

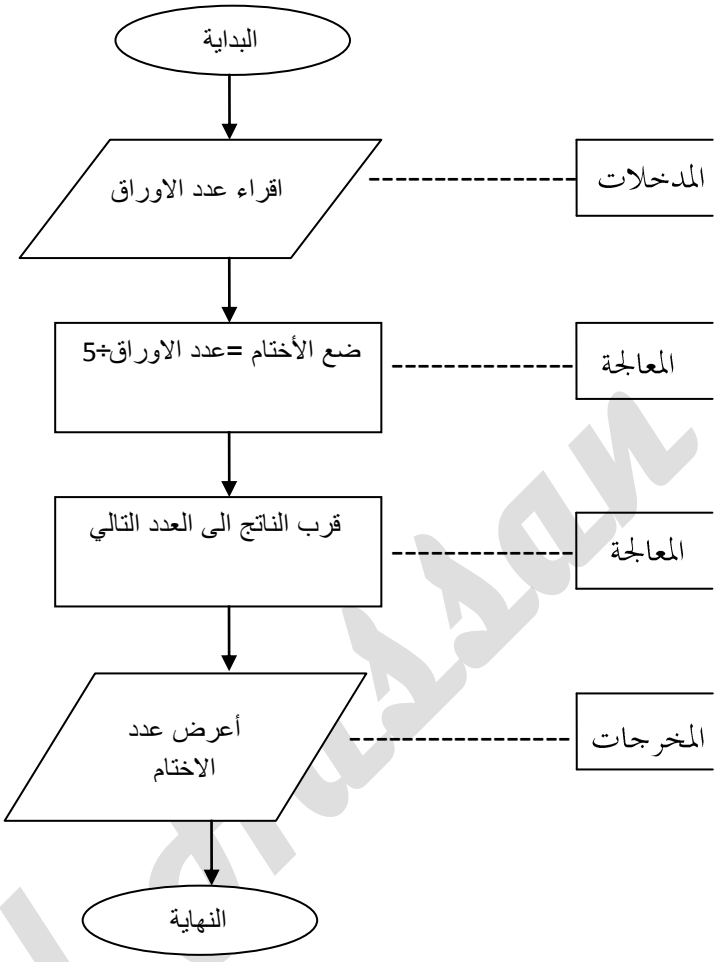
مخططات التدفق يجب ان تتدفق من اعلى الصفحة الى اسفل الصفحة . وبالرقم من ان الرموز المستخدمة في مخططات التدفق تعتبر قياسية الا انه لا توجد قياسية في تحديد مقدار التفاصيل المطلوبة مع كل رمز.

جدول رموز المخططات الانسيابية متفق عليه من قبل النعهد الامريكي القومي للقياسات ANSI والصورة 1-1 مخطط التدفق لمشكلة الختم في البريد والتي سبق ان قمنا بكتابة خوارزميتها.

من محاسن إستخدام مخططات التدفق للتخطيط للبرنامج هي تقديم تمثيل تصوري للمهمة او البرنامج وبذلك تساعد في سهولة فهم المهمة او ابرنامج منطقياً حيث يمكننا مشاهدة كل خطوة بوضوح وكيف تتصل كل خطوة مع التي تليها. ومن مساوئ مخططات التدفق عندما يكون البرنامج كبير جدا فان مخططات التدفق يمكن ان تستمر لاكثر من صفحة مما يجعل تتبع المخطط او تعديله صعباً.

الرمز Symbol	الإسم Name	المعنى Meaning
	سهم التدفق Flowline	يستخدم لربط الرموز ويشير الى التدفق المنطقي للبرنامج
	الطرفي Terminal	يستخدم لتمثيل البداية او النهاية في البرنامج
	الإدخال/الإخراج Input/output	تستخدم في عمليات الادخال والاخراج مثل القراءة والطباعة ويتم توضيح البيانات التي سيتم

		قرائتها او كتابتها في داخل الرمز.
	المعالجة Processing	تستخدم في العمليات الحسابية وعمليات المعالجة المختلفة. والتعليمات توضح داخل الرمز.
	القرار Decision	يستخدم لتمثيل اي عملية منطقية او عملية مقارنة. لا يشبه رموز الادخال والاخراج والمعالجة والتي لها مدخل واحد ومخرج احد اما رمز القرار Decision له مدخل واحد ومسارين لمخرجين وإختيار المسار يعتمد على الإجابة لسؤال نعم ام لا
	الموصلة Connector	تستخدم لربط السهم المختلفة
	موصل الصفحة Offpage Connector	يستخدم للإشارة بان المخطط النسيابي مستمر الى الصفحة التالية
	التزييل Annotation	تستخدم لإعطاء معلومات إضافية عن رمز مخطط تدفق اخر



الشكل (1-1) مخطط التدفق لمشكلة ختم البريد

الكود التقريبي (الزائف) Pseudocode:

هو عبارة عن نسخة مختصرة من الكود البرمجي الفعلى لذلك سمي بالكود التقريبي او الزائف. ونجد ان الرموز الهندسية المستخدمة في مخططات التدفق هنا تستبدل بجمل باللغة الانجليذية والتي توضح بإيجاز العمليات المختلفة في البرنامج. ويعتبر الكود الزائف شبه بالكود البرمجي الفعلي من مخططات التدفق.

الكود الزائف يسمح للمبرمج بالتركيز على خطوات حل المشكلة بدلا من استخدام لغات البرمجة مباشرةً. والمبرمج يمكن ان يصف اي الخوارزمية بالكود الزائف من دون التقيد بقواعد لغات البرمجة وعندما تكتمل كتابة الكود الزائف تصبح من السهولة ترجمته الى اي لغة برمجة اخرى.

الكود الزائف التالي يوضح حل مشكلة ختم البريد :

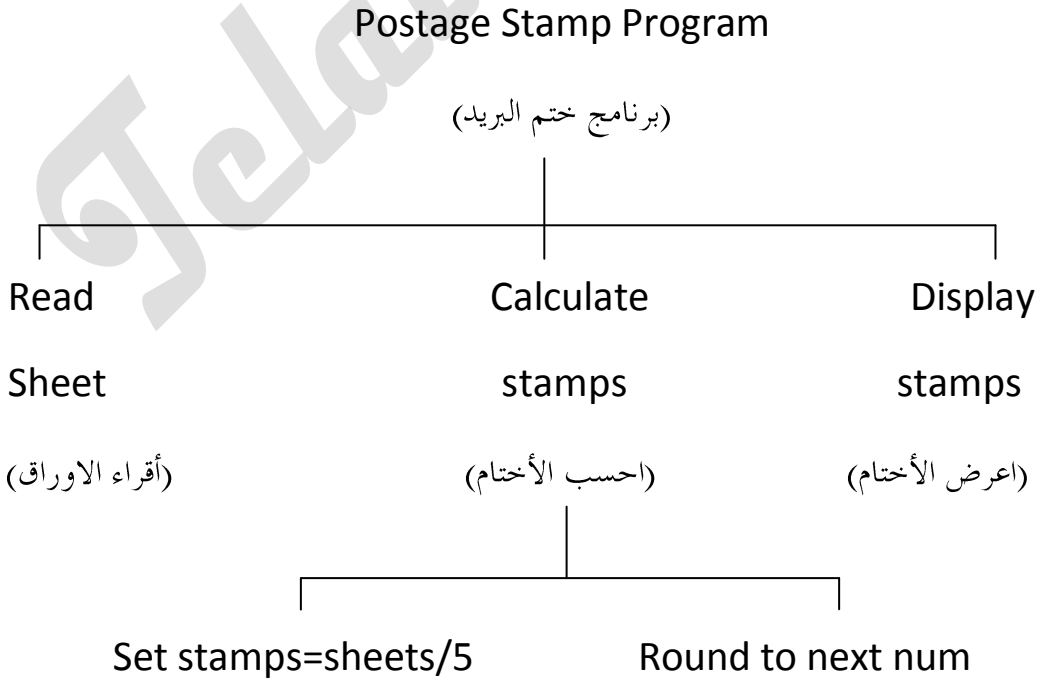
- Read sheets (input).
- Set the number of stamp to sheet/5 (processing).
- Round the number of stamp s up to the next whole number (processing).
- Display the number of stamps.

للكود التقريبي (الزائف) محاسن عديدة ومنها انه يعتبر مختصراً ولا يمتد الى صفحات متعددة مثل مخططات التدفق وأيضاً يعتبر شبيه بالكود الذي سوف يكتب لذلك فإن معظم المبرمجين يفضلونه.

مخططات التدرج Hierarchy Charts

اخر اداة برمجة سنتناقش عنها هي مخططات التدرج التي توضح التركيب العام للبرنامج وهي ايضاً تعرف بي مخططات التراكيب Structure Charts ولها صور نتعددة مثل HIPO وهي إختصار لي Hierarchy plus Input, Process and Output ويعني مخطط تدرج للمدخلات والمعالجة والمخرجات, Top-down-charts يعني المخطط من اعلى لأسفل, VTOC وهي إختصار Visual Table of Contents جدول المكونات المرئي. جميع هذه المسميات تشير الى تصميم مخططات شبيهة بمخططات تنظيم الشركات.

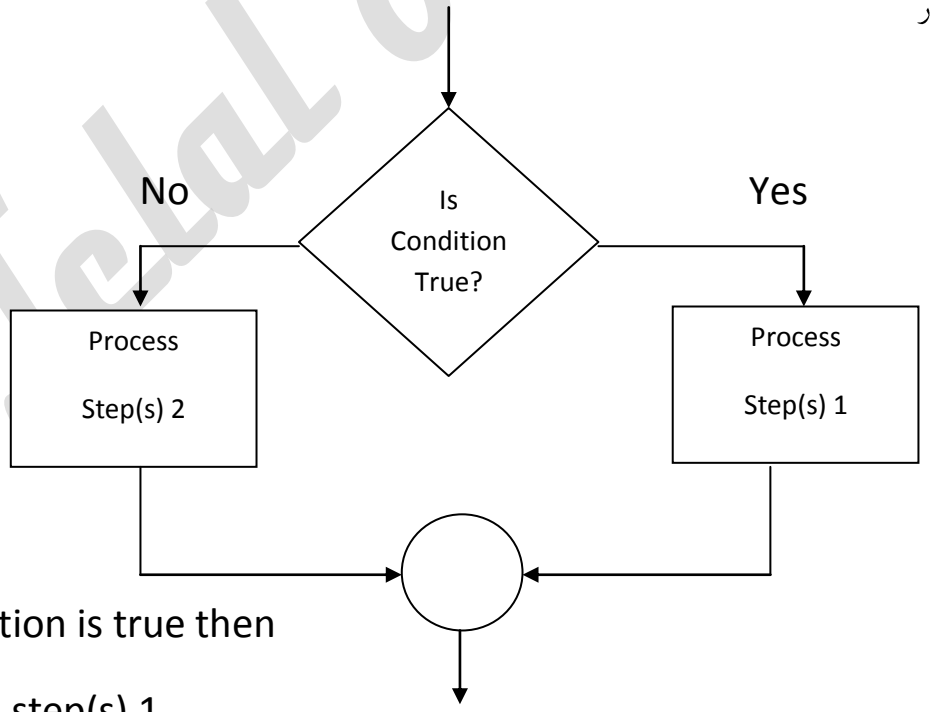
مخططات التدرج توضح تنظيم البرنامج ولكن يتم حذف عمليات منطقية محددة ومخططات التدرج تصف ماهية عمل كل جزء او وحدة module في البرنامج وما هي علاقة كل وحدة بالاخرى اما تفاصيل طريقة عمل كل وحدة فهي تحذف والمخططات تقراء من أعلى الى اسفل ومن اليسار الى اليمين وكل وحدة يمكن ان تقسم الى سلسلة من الوحدات الفرعية فإن جميع الوحدات تحت الوحدة الاصلية تمثل لمحة عامة لمخطط التدرج البرنامج وتوضح كل مهمة انجزت في البرنامج واين انجزت والشكل 1-2 يوضح مخطط التدرج لمشكلة ختم البريد.



شكل رقم 1-2 يوضح مخطط التدرج لمشكلة ختم البريد

الفائدة الاساسية من مخططات التدرج هي التخطيط المبدئي للبرنامج فإننا نقوم بتقسيم الاجزاء الرئيسية للبرنامج فبذلك يمكننا مشاهدة ما الذي يجب فعله على العموم ومن هذه النقطة يمكننا تحليل وحدة الى تفاصيل اكثر بإستخدام مخططات التدفق او الكود الزائف وهذه العملية تدعى بطريقة divide-and-conquer.

مشكلة ختم البريد تم حلها بسلسلة من الخطوات المتمثلة في قراءة البيانات, إجراء الحسابات وعرض النتيجة وكل خطوة سلسلة تابعة للتي تليها فإننا ننتقل من خطوة الى خطوة من دون تخطي اي خطوة او تعليمة في البرنامج وهذا النوع من التراكيب يعرف بالتراكيب المتسلسلة Sequence Structure. معظم المشاكل في بعض الاحيان تتطلب القرار لتحديد شرط لسلسلة من التعليمات التي يجب ان تنفذ اذا كانت الاجابة على السؤال بنعم فإن مجموعة من التعليمات يجب ان تنفذ واما اذا كانت الاجابة بلا فإن تعليمات اخري يجب ان تنفذ وهذا النوع من التراكيب يدعى بتراكيب القرار والشكل 1-3 يحتوي على مخططات التدفق والكود الزائف لتراكيب القرار



If condition is true then

Process step(s) 1

Else

Process step(s) 2

End if

سنستخدم كل من التراكيب المتسلسلة Sequence Structure وتراكيب القرار Decision Structure في حل المشكلة التالية:

خوارزمية الشوارع المرقمة في مدينة نيويورك:

المشكلة: تحديد إتجاه الشارع زو الاتجاه الواحد بإعطاء رقم الشارع اما اتجاه شرقاً او اتجاه غرباً

تحليل المشكلة: هنالك قاعدة بسيطة تخبرنا بإتجاه الشارع في نيو يورك وهي ان الشوارع زو الارقام الزوجية تتجه شرقاً.

المدخلات: رقم الشارع.

المعالجة: قرر مإذا كان الشارع يقبل القسمة على 2 ام لا.

المخرجات :إتجاه شرقاً او إتجاه غرباً.

الشكل 1-4 الى 1-6 يوضح الكود الزائف و مخططات التدفق ومخططات التدرج لمشكلة شوارع نيويورك المرقمة:

Program: Determine the direction of numbered NYC streets

Get street

If street is even Then

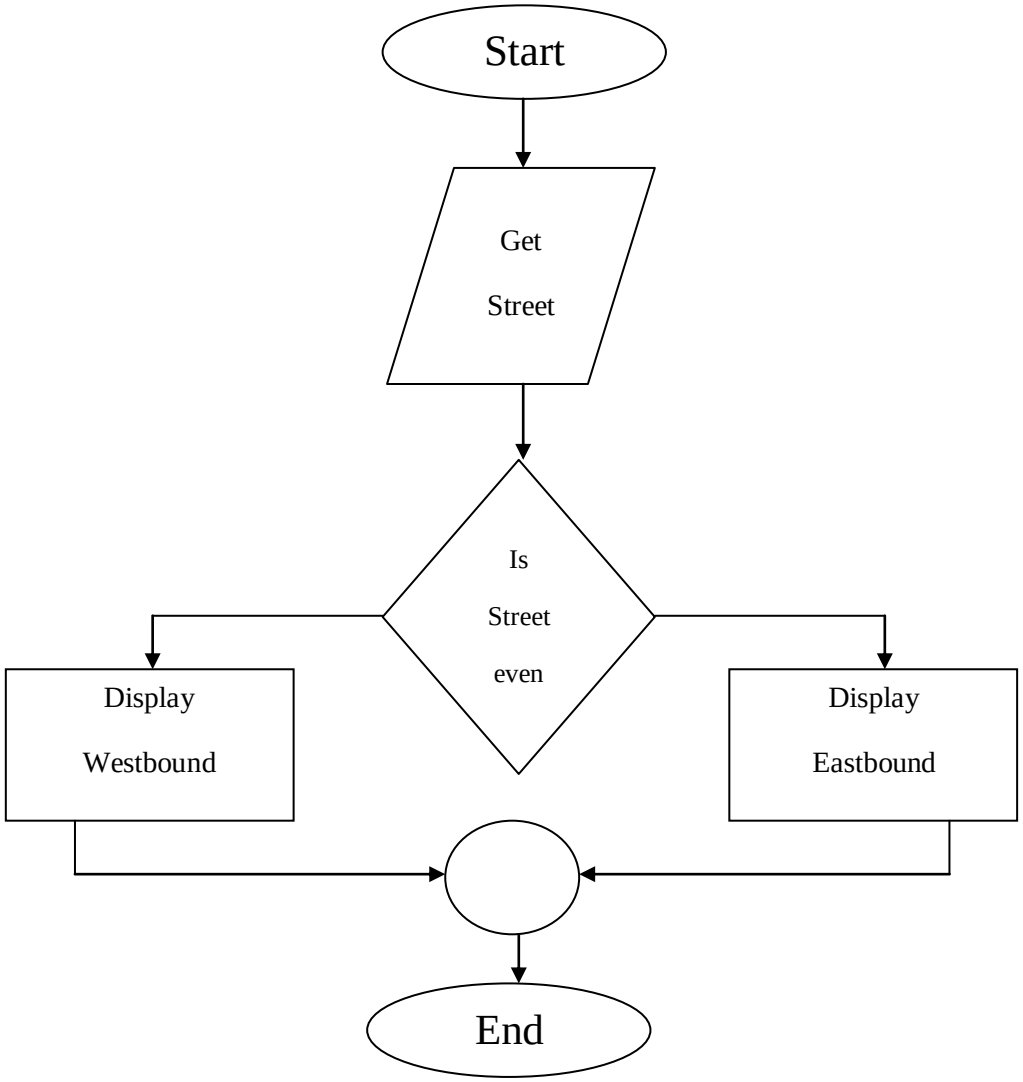
Display eastbound

Else

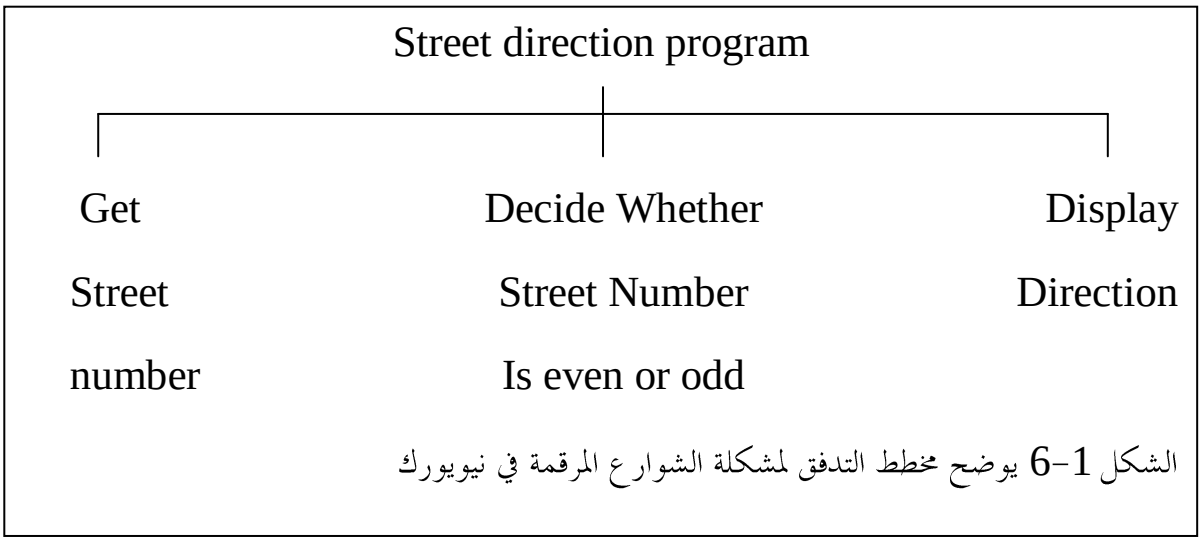
Display westbound

End If

الشكل 1-4 يوضح الكود الزائف لمشكلة الشوارع المرقمة في نيويورك



الشكل 1-5 يوضح مخطط التدرج لمشكلة الشوارع المرقمة في نيويورك



الشكل 1-6 يوضح مخطط التدفق لمشكلة الشوارع المرقمة في نيويورك

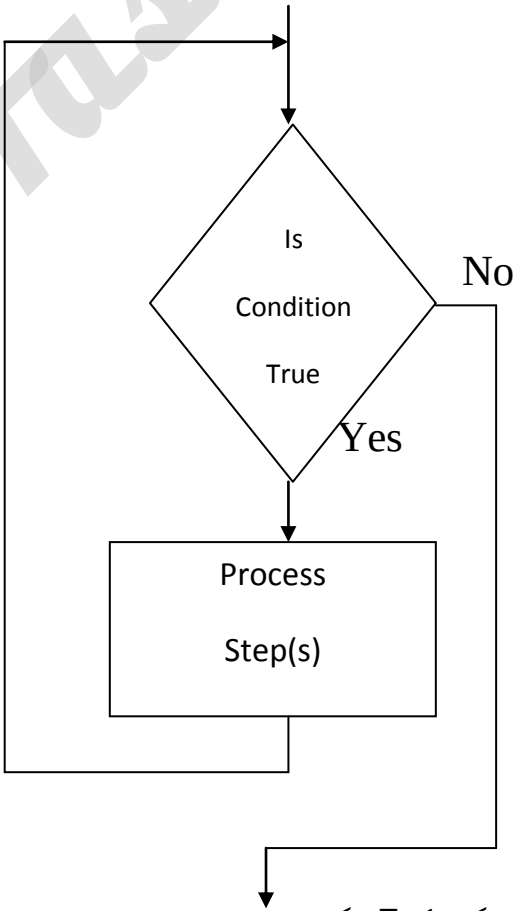
حل المشكلة التالية يتطلب تكرار سلسلة من التعليمات. التراكيب البرمجية التي يتم فيها تنفيذ عمليات عدد من المرات تركيب التكرار Loop Structure.

في تراكيب التكرار نحتاج الى شرط يوضح متي يجب ان ينتهي التكرار, ومن غير شرط خروج من التكرار فأن الحلقة سوف تتكرر بصورة غير منتهية وهنالك طريقة واحدة للتحكم في عدد مرات حلقات التكرار (ونادراً ما تشير الى عدد ثابت من الدورات او التكرارات) وهي إختيار شرط قبل كل مرة في كل حلقة والاستمرار في تنفيذ الحلقة طالما ان الشرط صحيح والشكل 1-7 يوضح.

Do while condition is true

Process Step(s)

Loop



الشكل 1-7 الكود الزائف ومخطط التدفق لتراكيب التكرار

خوارزمية معدل نتيجة الفصل الدراسي:

المشكلة: حساب معدل متوسط التقدير العام للطلاب

تحليل المشكلة: متوسط التقدير يساوي مجموع جميع التقديرات مقسوم على عدد الطلاب. ونحن نحتاج الى حلقة تكرارية لقراءة التقدير ثم إضافته الى مجموع تقديرات الطلاب, وفي الحلقة نحتاج الى عداد counter لحساب عدد الطلاب في الفصل انظر الشكل 1-8 الى 1-10.

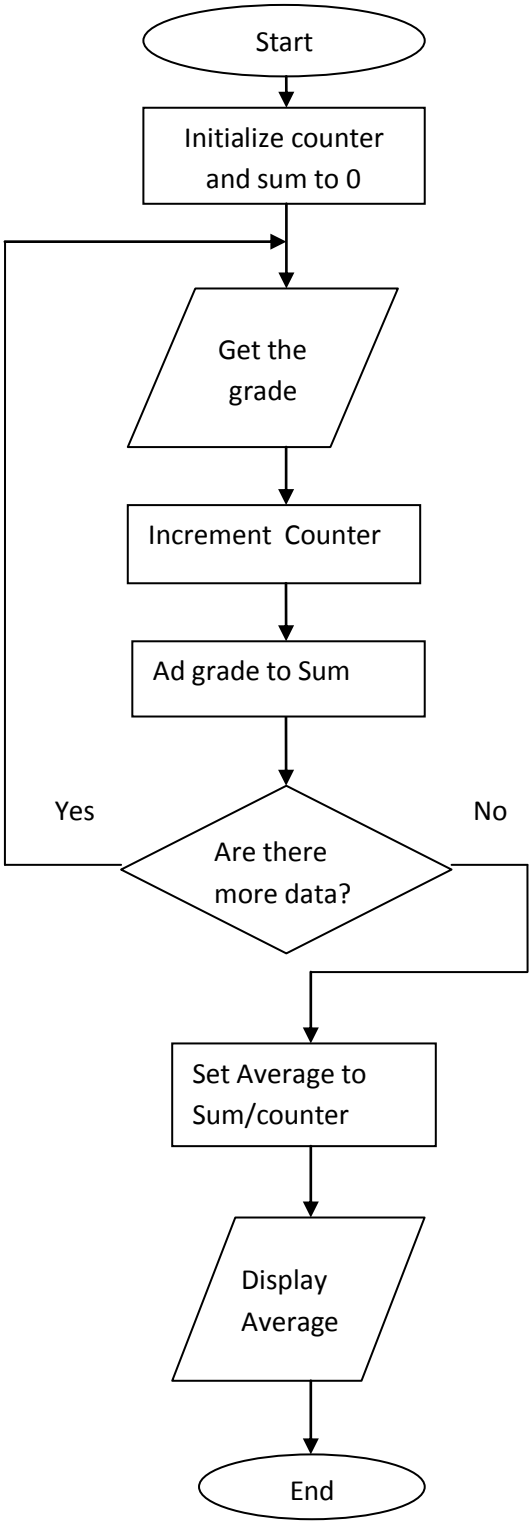
المدخلات: التقدير (الدرجة او النسبة بالارقام).

المعالجة: إيجاد مجموع التقديرات, حساب عدد الطلاب وحساب متوسط التقديرات.

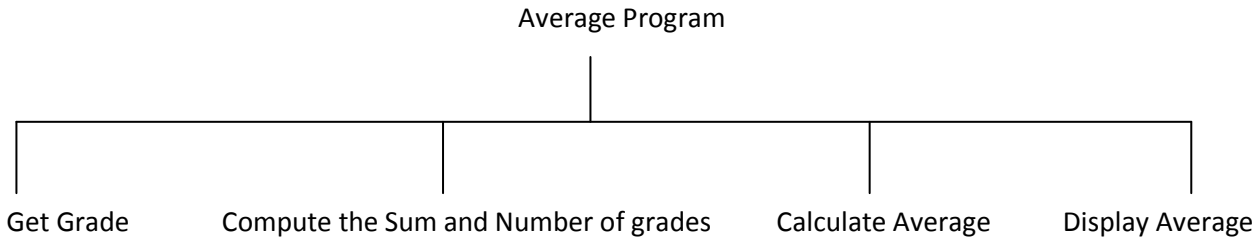
المخرجات: متوسط التقديرات.

```
Program : Determine the average grade og class
Initialize Counter and Sum to 0
Do while there are more data
Get the grade
Add the grade to sum
Increment the Counter Loop
Compute the Average=Sum/Conuter
Display Average
```

الشكل 1-8 يوضح الكود الزائف لمشكلة متوسط تقدير الفصل



الشكل 1-9 يوضح مخطط التدفق لمشكلة متوسط تقدير الفصل



الشكل 1-10 يوضح مخطط التدرج لمشكلة متوسط تقدير الفصل

نقاط هامـــــــــــــة

- الخوارزمية يجب ان يتم اختبارها في مرحلة مخططات التدفق قبل عملية الكود وتحويله الى برنامج. والبيانات المختلفة التي تستخدم في المدخلات او المخرجات يجب ان تُختبر وهذا الإجراء يسمى بال desk checking .
- مخططات التدفق والكود الزائف ومخططات التدرج تعتبر من ادوات حل المشاكل عالمياً فيمكن استخدامها لبناء البرامج في اي لغة برمجة.
- هنالك اربع تراكيب برمجية منطقية اساسية : التسلسل Sequence, القرار Decision, التكرار Loop والتفرع الغير مشروط Unconditional Branch وهو يظهر في بعض لغات البرمجة مثل جملة Goto وهي تقوم بالقفز من مكان الى اخر. البرمجة الهيكلية Structured Programming تستخدم اول ثلاثة تراكيب ولكنها لا تستخدم الرابعة. ونجد ان من محاسن الكود الزائف على مخططات التدفق هي ان الكود الزائف لاتوجد به فقرة او بند يتضمن التفرع الغير مشروط لذلك فإنه يجبر المبرمجين على كتابة برامج هيكلية Structured Programs.
- مخططات التدفق تستهلك الوقت في الكتابة وصعبة التعديل لهذا السبب المبرمجين المحترفين يفضلون على الارجح الكود الزائف ومخططات التدرج لان مخططات التدفق تقوم بتوضيح وتعريف التدفق البرمجي لتقنيات البرمجة ولذلك فانها من الادوات المتاحة في تعليم البرمجة.
- عملية تتبع مخططات التدفق تشبه لعبة الطاولة فنحن نبداء من رمز البداية ثم ننتقل بعد ذلك من رمز الى اخر حتى نصل الى زمز النهاية.
- هنالك اساليب متعدد للكود الزائف. بعض المبرمجين يستخدمون نومذج العناوين بينما الاخرين يستخدمون نموذج يشبه في معظمه لغة البرمجة والافضل ان يركز الكود الزائف

على المهام الاساسية التي يجب ان تنجز ويتم تجنب التفاصيل الصغيرة والمتكررة تم اكمال ما تبقى في مرحلة البرمجة.