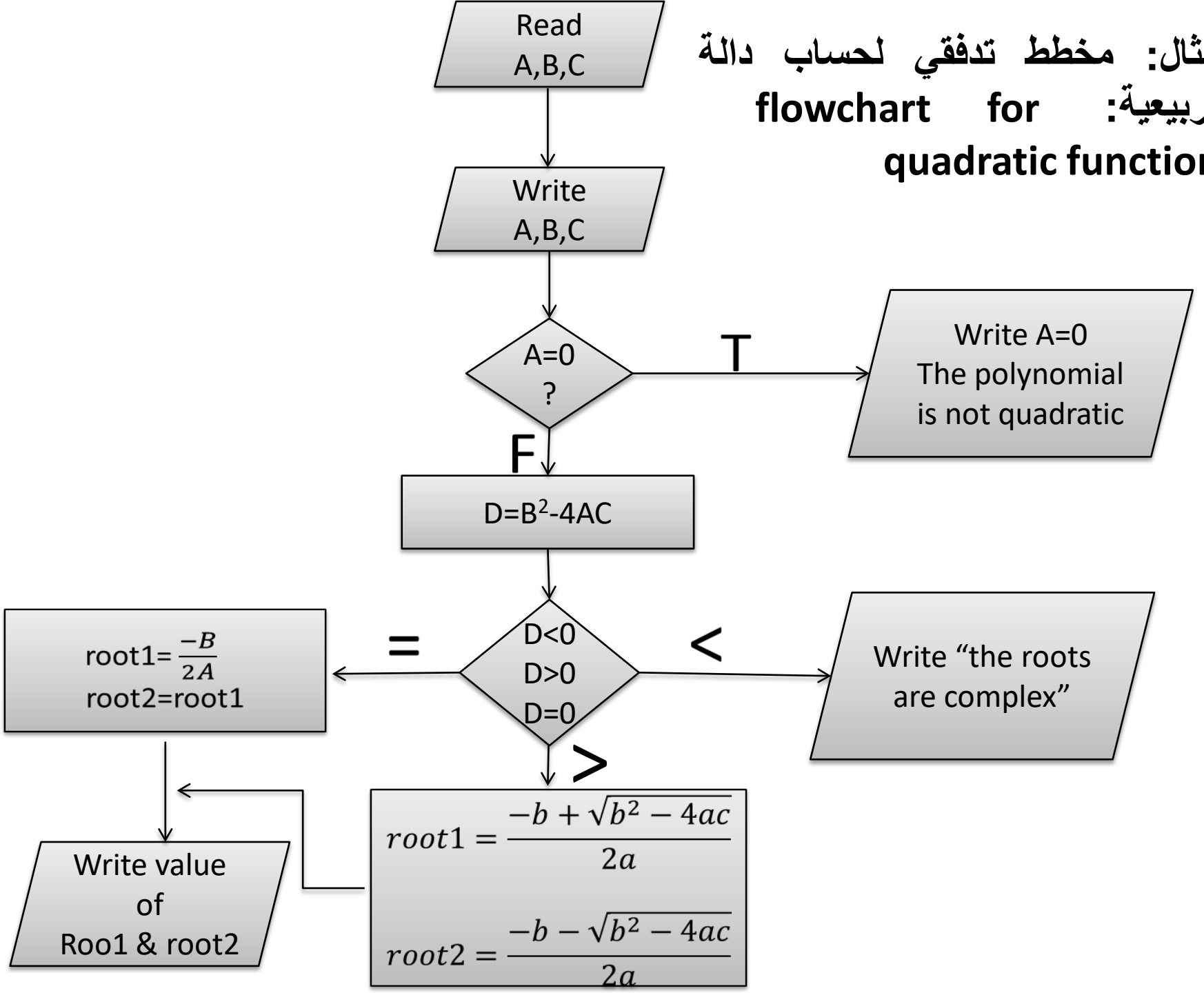
مراجعة لعملية البرمجة: Programming Overview

تعتبر حواسب اليوم آلات سريعة الأداء وممتازة التخزين وصغيرة الحجم لكنها لا تفكر ولا تفهم بل يجب تلقينها بدقة كيفية حل مسألة ما وكيفية اتخاذ القرار وهو ما يعرف بعملية البرمجة.

خطوات البرمجة:

أولا- تخطيط البرنامج: وفيه يتم تحديد تتابع العمليات المطلوبة لحل مسألة ما وعلى الأغلب تستخدم المخططات التدفقية flowchart تشترك كل اللغات بهذه الخطوة

مثال: مخطط تدفقي لحساب دالة تربيعية: flowchart for quadratic function



ثانيا- تشفير المخطط: وتتضمن كتابة الخطوات الموجزة السابقة في المخطط وفق لغة خاصة يستطيع الحاسب ترجمتها وتفسيرها، وهنا :

- يجب كتابة البرنامج بلغة يمكن للحاسب تحويلها الى لغة الالة machine languages.
- لغة الالة صعبة على المبرمج لذا استخدمنا لغات ذات عدد محدود من الصيغ والعبارات، ولها صنفان :

الاول : اللغات التجميعية (assembly languages):

- لغات خاصة ولكل معالج لغته، ولا تعتبر قابلة للحمل not portable.
- تحتاج لمجمع assembler لتحويل تعليمات اللغة الى لغة الآلة (لغة المكونات المادية وهي الأصفار والواحدات).
- تعليماتها ذات شيفرات مختصرة mnemonic codes وتتكون من العملية المراد تنفيذها (+، -، *، /، AND، OR،...) والعنوان الواجب استخدامه .

مثالها: ADD 535

بعض التعليمات الشائعة:

Operation code (Opcode)	Address part	Instruction function
CLA	430	مسح محتويات المراكم ووضع محتويات العنوان 430 فيه. CLA = clear and add
ADD	530	جمع ما بداخل العنوان 530 الى محتوى المراكم وتخزين النتيجة في المراكم
SUB	235	طرح ما بداخل العنوان 235 من محتوى المراكم ووضع النتيجة في المراكم A-235 -← A
STO	433	تخزين محتوى المراكم تحت العنوان 433. تبقى محتويات المراكم نفسها وتتلف محتويات الموقع STO = store
HLT	000	طلب توقف للمعالج HALT
MUL	400	ضرب محتويات الموقع 400 بما في المراكم ووضع الناتج في المراكم ، وهي تساعد في تنفيذ برامج أكثر

برنامج جمع ثلاثة أعداد مخزنة في مواقع الذاكرة المتتالية 6، 7، 8.

Address in memory	Instruction opcode	Word address	Content of acc. After instruction is performed
1	CLA	6	200
2	ADD	7	500
3	ADD	8	900
4	STO	9	900
5	HLT	0	900
6	contains	200	
7	contains	300	
8	contains	400	

اكتب برنامج يحسب ax^3+bx^2+cx+d حيث o,x,d,c,b,a مخزنة في المواقع 27,26,25,24,23,22

Address in memory	Instruction opcode	Word address	Content of acc. After execution
1	CLA	22	a
2	MUL	26	ax
3	ADD	23	ax+b
4	MUL	26	$X(ax+b)=ax^2+bx$
5	ADD	24	ax^2+bx+c
6	MUL	26	$X(ax^2+bx+c)=$ ax^3+bx^2+cx
7	ADD	25	ax^3+bx^2+cx+d
8	STO	27	ax^3+bx^2+cx+d
9	HLT	000	ax^3+bx^2+cx+d
22	contains	a	
23	contains	b	
24	contains	c	
25	contains	d	
26	contains	x	
27	contains	o	

تعليمات التفرع والقفز branch , skip or jump instructions

- غير شرطية: مثالها BRA 420 وتعني اقفز لتنفيذ العملية في العنوان 420 وتابع في العنوان 421 دون مراعاة أي شرط.
- شرطية: مثالها BRM 420 وتعني اقفز اذا كانت القيمة في ACC سالبة فقط . BRM = branch on minus

ميزات لغة الاسمبلي:

١. شيفرات تشغيل مختصرة (مجموعة تعليمات محدودة).
٢. تمثيل مناسب للمعطيات (عشري، ست عشري، أحرف، رموز ..).
٣. يمكن طباعة قائمة تضم تعليمات البرنامج بالأسمبلي والبرنامج الغرضي object program والذي هو بلغة الالة مما يساعد على تتبع الاخطاء من قبل المبرمج.
٤. كشف الاخطاء في البرنامج: يستطيع برنامج الاسمبلر التعرف على الاخطاء وتبيانها (مثل استخدام قيمتين لـ x أو شيفرة تعليمة غير صحيحة).

الثاني: اللغات عالية المستوى (high level languages):

- ابسط ومستقلة عن نوع المعالج .
- من ميزاتها: قابلية الحمل والتوثيق وسهولة التعلم .....
- من امثلتها : C , C++ , BASIC, Fortran, Java, Python...

مثال: جمع عددين صحيحين بلغة C++ :

```
//Addition Program
#include<iostream>
int Main()
{
int x, y;
int sum;
cout <<  "Enter first integer\n";
cin >> x;
cout << "Enter second integer\n";
cin >> y;
sum = x + y;
cout << "Sum is: " << sum<<endl; //print sum
Return 0;
}     //end function main.
```