

Casio计算器的“编程”

灰天飞雁 <http://www.cnblogs.com/htfy>

March 28, 2014

1 介绍一下计算器吧



正如你所见，这是一个非常普通的CASIO fx-350ES PLUS型计算器，整数存储支持10位，浮点数支持 $10^{-99} \sim 10^{99}$ 。

除此以外，其还有9个变量的存储功能以及赋值语句 \rightarrow ，同时还拥有:的语法使得每按一次=就会顺次执行各个:隔开的语句，这使得编程变为了可能。

2 基础

一切编程语言最基本的要具有逻辑判断功能，这一章就说说如何构建逻辑判断。

- 一个BUG 计算器对于产生的中间结果小于 1×10^{-99} 就会被作为0处理

- 等于

$$\text{BOOL } equal(a, b) = \left(\frac{1}{1 + |a - b|} \cdot 10^{-99} \right) \cdot 10^{99}$$

- 小于

$$\text{BOOL } less(a, b) = \left(\frac{1}{1 + (1.000000001^{a-b} \times 10^{-99}) \times 10^{99}} \times 10^{-99} \right) \times 10^{99}$$

- $\text{BOOL?}a\#b$

$$(a - b) \cdot \text{bool} + b$$

- 判断整数

$$\frac{1}{1 + |\sin \pi x|} \cdot 10^{-99} \cdot 10^{99}$$

3 那就开始吧

3.1 计算 $2^1 + \dots + 2^{15}$

首先变量清0

$$0 \rightarrow E : 0 \rightarrow A : 0 \rightarrow X$$

E: 是否循环结束。A: 循环次数。X: 和。

$$\text{less}(A, 15)?1\#0 \rightarrow E : A + 1 \rightarrow A : E?X + 2^A\#X \rightarrow X$$

打开的话，就是

$$\frac{1}{1 + (1.000000001^{A-15} \times 10^{-99}) \times 10^{99}} \times 10^{-99} \times 10^{99} \rightarrow E : A+1 \rightarrow A : E \times 2^A + X \rightarrow X$$

3.2 二重循环！

现在我们想计算

$$\sum_{A=1}^7 \sum_{B=0}^A C_A^B$$

这次我们需要用到5个变量。F记录是否里层循环结束，E标记程序的结束，A、B为循环变量，X为所求的和。

首先变量清0

$$0 \rightarrow E : 0 \rightarrow F : 0 \rightarrow A : 0 \rightarrow B : 0 \rightarrow X$$

然后计算

$$\text{equal}(A, B) \rightarrow F : F?A+1\#A \rightarrow A : F?0\#B+1 \rightarrow B : \text{less}(A, 8) \rightarrow E : E?X + C_A^B\#X \rightarrow X$$

4 系统化

4.1 step思想

计算器只会循环计算，我们可以用 $Step$ 这个变量记录当前所需要执行的语句¹

4.2 基本语句

1. 基本的赋值

- 书写形式 $A \rightarrow B$

- 实际形式²

$$: equal(i, step)? A \# B \rightarrow B$$

$$: equal(i, step)? step + 1 \# step \rightarrow step$$

2. 无条件跳转

- 书写形式 $jmp T$

- 实际形式

$$: equal(i, step)? T \# step \rightarrow step$$

3. 有条件跳转

- 书写形式 $jmpx x, T$

- 实际形式

$$: equal(i, step)? x? T \# step + 1 \# step \rightarrow step$$

4. 无条件中断

- 书写形式 $int T$

- 实际形式

$$: equal(i, step)? i + 1 \# R \rightarrow R$$

$$: equal(i, step)? T \# step \rightarrow step$$

5. 有条件中断

¹不一定是实际的::之间的“语句”，指的是下列的“基本语句”

²i为行号，下同

- 书写形式 $\text{int } x, T$

- 实际形式

$$: \text{equal}(i, \text{step}) \cdot x ? i + 1 \# R \rightarrow R$$

$$: \text{equal}(i, \text{step}) ? x ? T \# \text{step} + 1 \# \text{step} \rightarrow \text{step}$$

4.3 程序结构

讲了这么多，可能大家都不知道 R 是什么，下面我们就来看一个程序

```

1  jmp 6
2  x+1 -> F
3  jmp R
4  x*x-x+1 -> G
5  jmp R
6  A -> X
7  int 4
8  G -> M
9  (A+B)/2 -> X
10 int 4
11 G -> N
12 M*N<0 ? A # (A+B)/2 -> A
13 M*N<0 ? B # (A+B)/2 -> B
14 jmpx |A-B|<=10^(-7) , 14

```

从第1行到第5行，都是我们所说的函数区，函数区开头有一个无条件跳转指令跳转到主程序部分，防止不需要时计算函数。

可以看到，这里定义了两个函数 F 和 G ，每个函数的最后的 R 指的是跳转回的地址，由于函数调用必须是通过 int 体现，最后必须要跳转回 int 的下一句继续整个程序。

最后一句是在二分法寻找零点满足条件时结束程序（自循环用意）。