实验一、分支和循环结构的简单程序设计

**1、实验目的（1分钟）**

（1）熟悉可视化计算工具Raptor的运行环境。

（2）掌握Raptor中赋值、输入、输出、过程调用、选择、循环6种符号的使用方法。

（3）能够设计顺序、选择、循环结构的简单程序。

1. **实验内容（2分钟）**

（1）顺序结构程序设计：设计一个Raptor程序，计算并输出两个正整数a和b的和，a和b的值由用户输入。

（2）选择结构程序设计：设计一个Raptor程序，计算两个整数a和b的最大值并输出，其中a和b的值由用户输入。

1. 循环结构程序设计：使用循环结构，设计一个Raptor程序，计算并输出1+2+3+.....+100的结果。
2. 质数（Prime Number）又称素数，它指的是对于一个大于1的自然数，除了1和它本身外，不能被其它自然数整除，换句话说就是该数除了1和它本身以外不再有其它的因数，否则称为合数。请编写Raptor程序，求解出100以内的所有质数。

**3、实验设计思路（12分钟）**

（1）选择结构程序设计举例：设计一个Raptor程序，计算两个整数a和b的最大值并输出，其中a和b的值由用户输入子过程（计算最大值要求用子函数）。

**实验步骤：**打开Raptor软件，单击左半部分的input符号框，将其拖动到start与end之间。双击输入符号框(Input)，在“Enter Prompt here”部分输入提示文本" Please enter a value for variable a:"，在“Enter Variable here”部分输入变量名a，该变量用于存储输入的变量值。对变量b进行相同的操作，并点击done按钮。插入输入符号框的结果如图1.6。

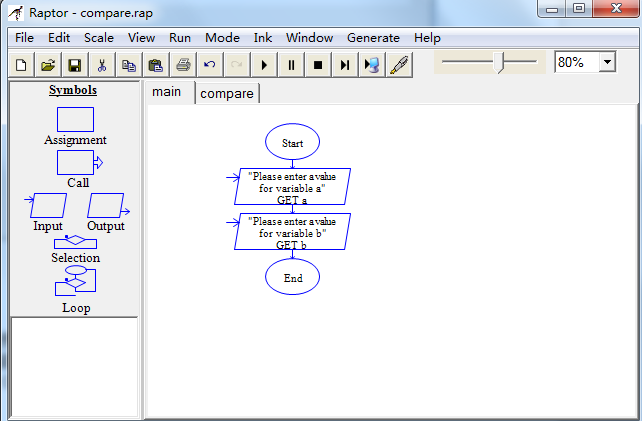


图1.6 插入a和b的输入符号框

接下来使用一种新的符号框—调用(Call),通过该符号框可以调用一个能够完成特定功能的子过程，该子过程也被称为子函数。加入调用符号框之后，结果如图1.7所示。

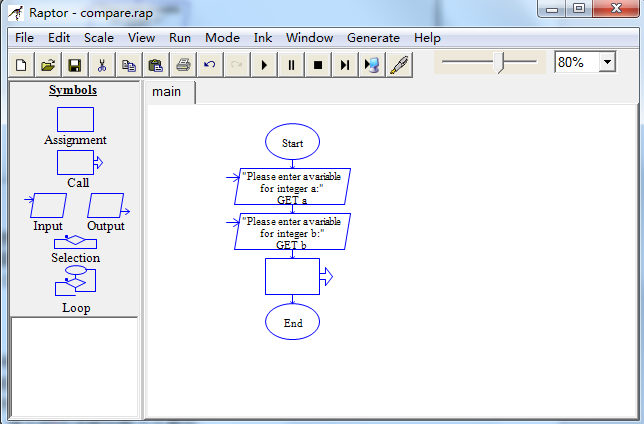


图1.7 插入调用符号框

插入调用符号框是为了调用一个子过程，在调用该子过程之前需要先定义它，要求该子过程能够完成两个整数的比较，并返回比较的结果。定义子过程的方法如下，首先右击main(如图1.7的红色箭头所示），在弹出的选项中选择Add Procedure，弹出的对话框如图1.8所示。

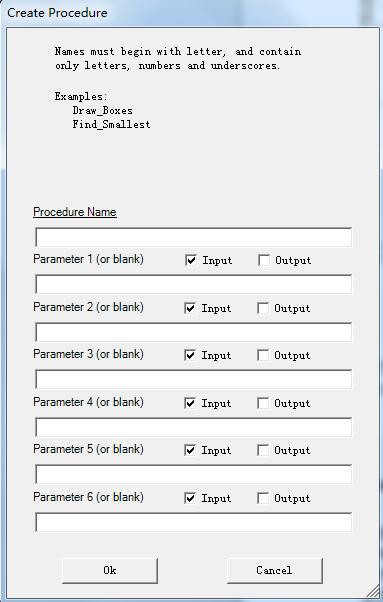


图1.8 插入子过程

在图1.8中，Parameter1........Parameter6代表参数，参数分为两种类型：输入类型(input）和输出类型（output），通过参数后边的框中进行选择相应的类型。在compare子过程中，可以设置三个参数r，a和b（此处参数名称可以选择用其他的合法字符表示，不仅限于r，a，b)。将第一个参数设置为输出（output）类型，用于保存比较后所得到的结果，剩下的两个参数为输入（input）类型，用于保存将要进行比较的数值。结果如图1.9所示。

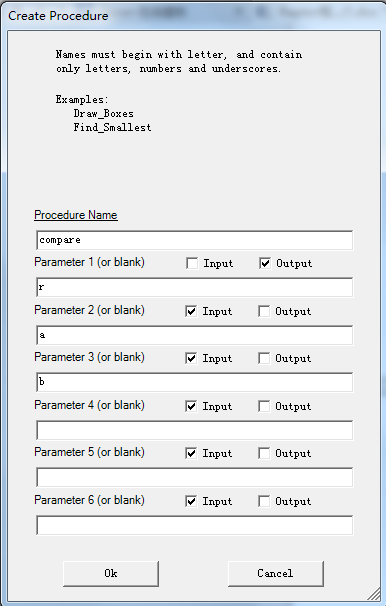


图1.9 定义子过程的名称和参数

点击ok按钮，可以得到如图1.10所示的子过程。

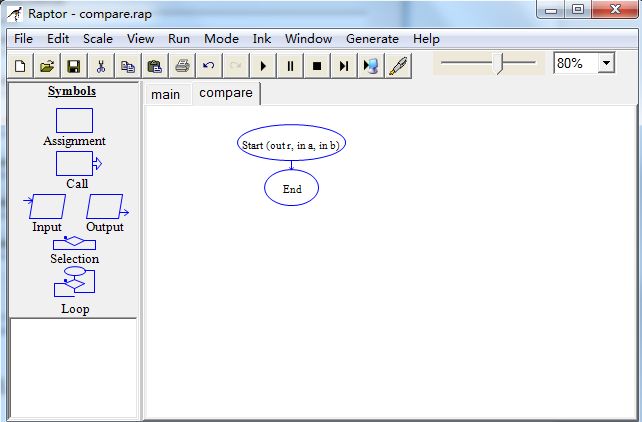


图1.10 子过程

接下来需要对该子过程进行定义，此处需要单击选择(selection)符号框，将其拖动到start之后，双击selection中的菱形，弹出如图1.11所示的输入选择条件对话框。

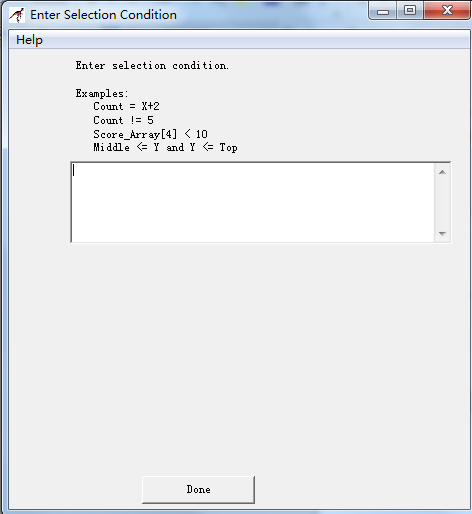


图1.12 输入选择条件对话框

在该对话框中，需要输入分支条件，在此输入a>b(也可以输入a<=b，不同的选择对应的分支不同)，点击“done”按钮，结果如图1.12所示。

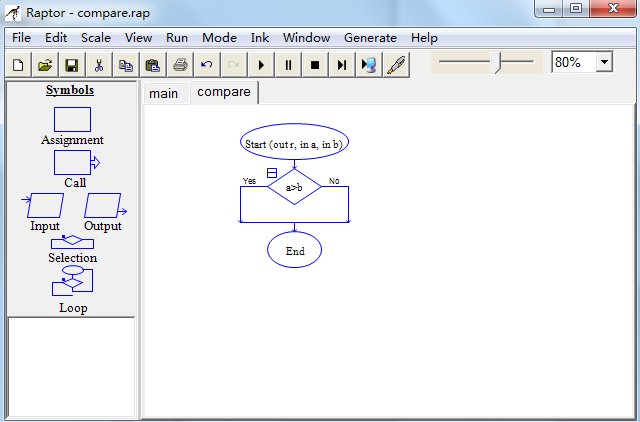


图1.12 输入分支条件对话框

在图1.12中，当a>b成立时，也就是对应左分支(Yes），此时较大值为a，用r保存结果，插入赋值符号框并将a赋值给r；当a>b不成立时，也就是右分支(No),此时插入赋值符号框将b赋值给r。单击赋值（Assignment)符号框，将其拖动到Yes和No对应箭头的下方，并进行赋值，结果如图1.13所示。

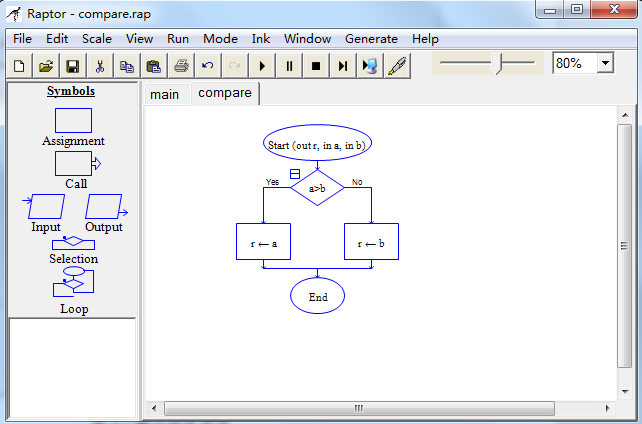


图1.13 分支选择后的结果

定义好“子过程”，接下来通过调用(call)符号框来调用该子过程，单击图1.7中的调用符号框，会弹出如图1.14所示的对话框。

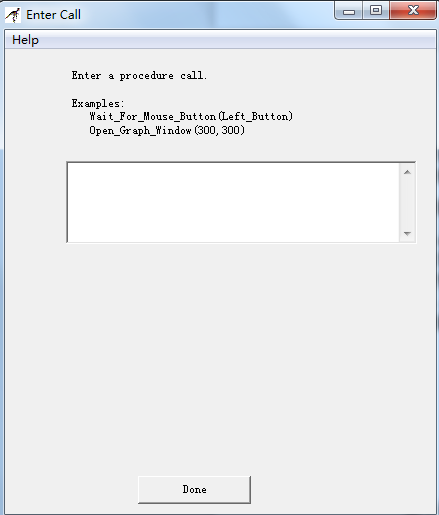


图1.14 输入调用的子过程对话框

在图1.14中要求输入需要调用的子过程，要调用的是已经定义过的compare子过程，可以在图中看到提示，最开始的部分是子过程的名称（如Open\_Graph\_Window)，括号里边的内容是子过程的参数，要注意的一点是参数要匹配，这里所说的匹配包括参数个数、参数顺序和参数类型。输入compare(m,a,b),点击done按钮得到的结果如图1.15所示。

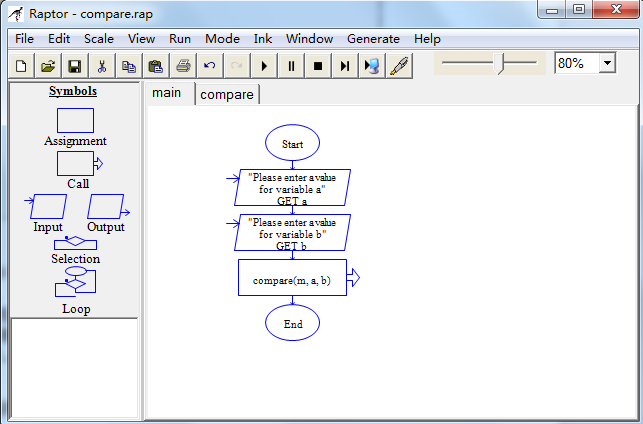
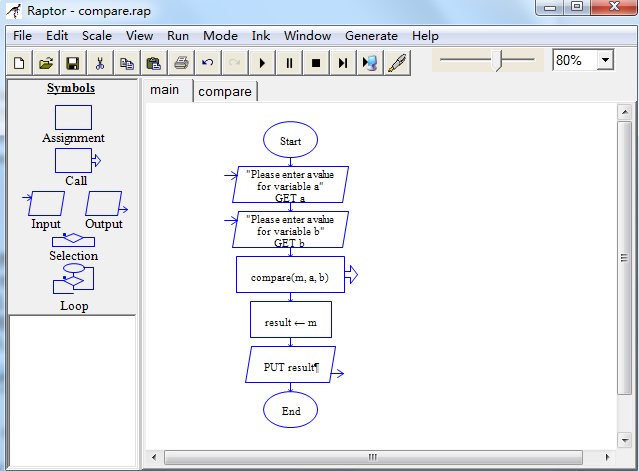


图1.15调用子过程

在compare(m,a,b)中，m对应的是子过程中的r，而r是用来存储输出的结果，在调用子过程结束时，会将r的值传递给m。最后定义一个名为result的变量来存储最终的结果并将其输出，因此需要插入赋值符号，并将m的值赋给result并输出result，完整的Raptor程序，结果如图1.16所示。



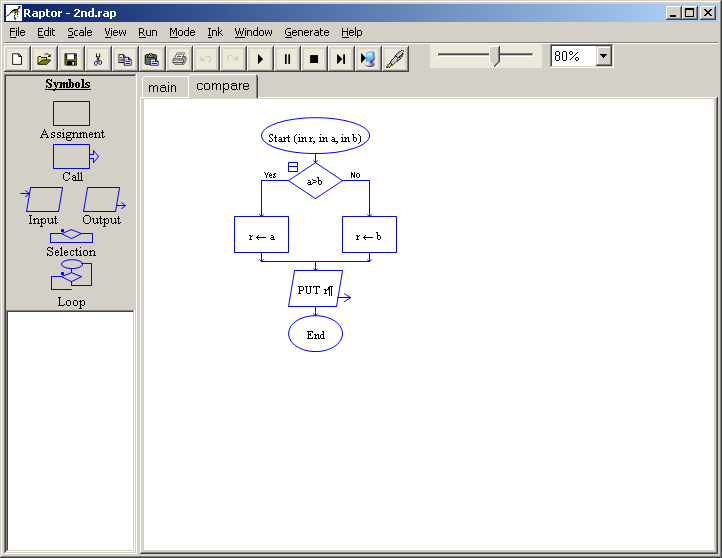
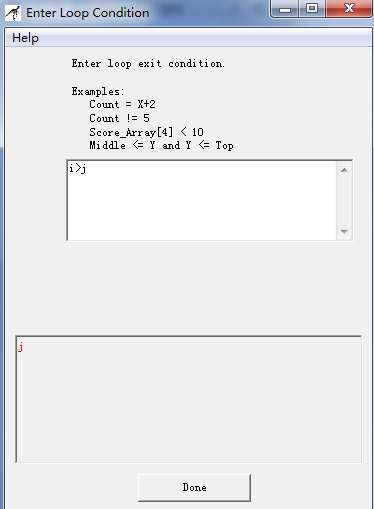


图1.16 完整的程序

（2）循环结构程序设计举例：使用循环结构，设计一个Raptor程序，计算并输出1+2+3+.....+100的结果。

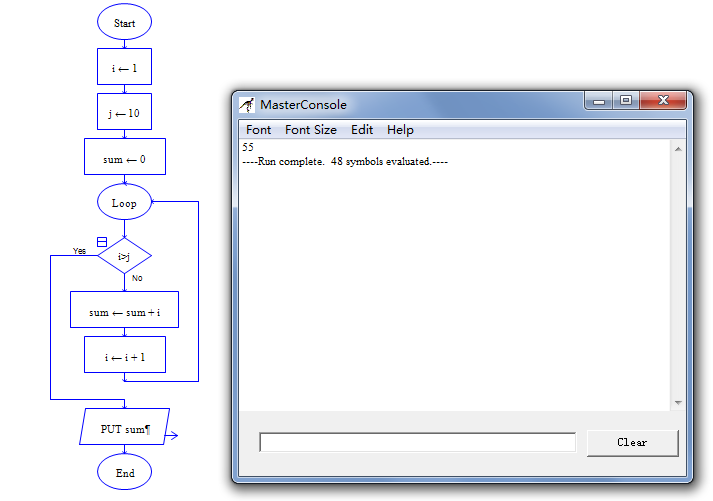
**实验步骤：**打开Raptor软件，首先在程序中加入三个变量i,j和sum，用sum表示最终求得的结果；i即为当前进行累加的值，又是当前统计的累加的变量个数，因此i的初始值为1；j为变量的总数，因为本例中一共有100个变量，因此j的值为100。接下来是程序中最重要的部分，也就是引入循环符号。循环符号需要一个判断条件，利用该条件是否成立来判断要循环是否继续。在该例中要计算100个数的累加，因此当i的值大于100时循环即结束。否则继续执行累加。

单击循环(Loop)符号将其拖入sum赋值符号的下方，双击循环符号的菱形部分，会弹出如附图A-28的对话框，该对话框要求输入循环是否继续进行的条件，由于此处做的是对10个数进行累加，因此可以输入i>j(i的初始值为1，j的初始值为10)。当i>j,即i>10时，说明已经累加了十个数，循环结束。

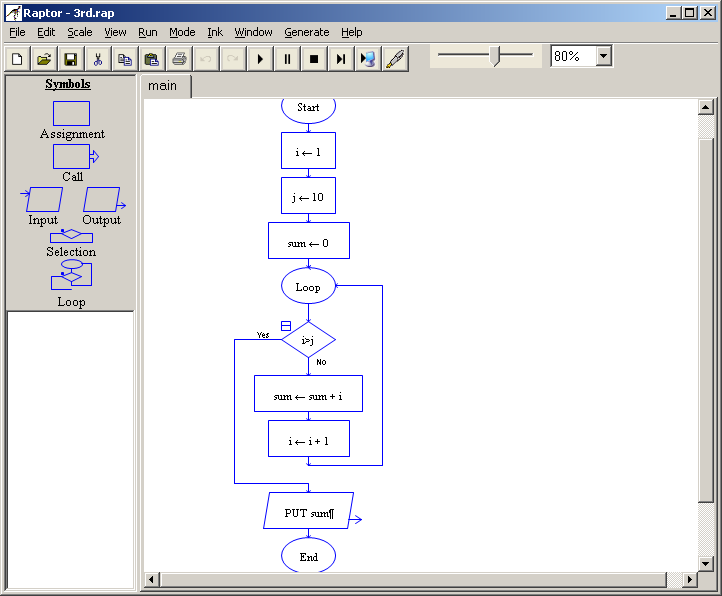


附图A-28 输入循环结束条件

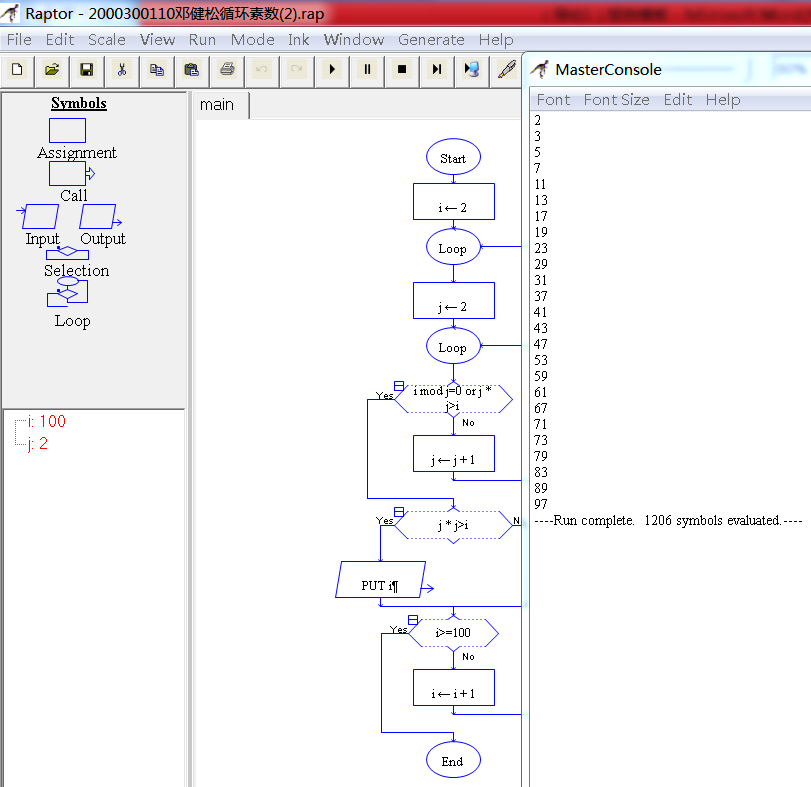
当i>j时，说明已经进行了10次累加，循环结束，并输出计算结果；如果i<=j，那么累加还未结束，接下来要继续进行累加。累加的实现方式如下：首先，sum=sum+i，这是进行第i次累加，在进行第i次累加之前，sum存储的是前i-1次累加的结果(i从1递增到i-1),然后加上i，即可得到前i次累加的结果；其次，因为下一次要进行的是第i+1次累加，还要判断i的值是否大于j以确定循环是否继续进行，因此还要执行i=i+1操作。加入这两步操作并输出sum即可得到完整的Raptor程序。完整的Raptor程序如附图A-29所示。程序的运行结果如附图A-30所示。



附图A-29 完整的raptor程序 附图A-30 程序的运行结果



（3）循环结构程序设计举例：质数定义：只能被1或者自身整除的自然数（不包括1），称为质数。  
　　**算法:**利用定义可以循环判断该数除以比它小的每个自然数（大于1），如果有能被它整除的，则它就不是质数。



**4、实验重点（1分钟）**

掌握基于可视化计算工具Raptor的选择、循环语句的简单程序设计的使用方法。

**5、实验难点（1分钟）**

可视化计算工具Raptor的“子过程（子函数）”的使用方法。

**6、实验注意事项（1分钟）**

（1）输入的字符串需要加双引号

（2）编写子过程（子函数）时mode的设置