CompPhys Assignment Week 01

李尚坤 20307130215

1. **求解实系数方程的解**
   1. **题目描述**

求解实系数方程的解。要求：系数为程序执行时键盘输入，为任意实数，所有的解均输出到屏幕。

* 1. **解决方案描述**

根据输入的系数情况，共分为4种情况讨论：

* 所有系数全为零，方程恒成立，任意复数均为方程的根；
* 二次项和一次项系数为零，常数项系数不为零，方程无根；
* 二次项系数为零，一次项系数不为零，方程为一元一次方程，有一个根；
* 二次项系数不为零，方程为二元一次方程，根据判别式求解。若，方程有两个相异实根；若，方程有两个相等实根；若，方程有两个相异复根。
  1. **伪代码**

|  |  |
| --- | --- |
| Algorithm 1 求解实系数方程 | |
| ***Input***: three real numbers a, b, c | |
| ***Output***: solutions of equation , (real or complex numbers) | |
| 1: | **If** and and |
| 2: | **Then** any complex number |
| 3: | **Else** **If** and and |
| 4: | **Then** no root |
| 5: | **Else If** **and** |
| 6: | **Then** |
| 7: | **Return** |
| 8: | **Else** //when |
| 9: | delta //判别式 |
| 10: | **If** delta |
| 11: | **Then** , |
| 12: | **Else If** delta |
| 13: | **Then** |
| 14: | **Else** |
| 15: | **Then** , |
| 16: | **End** |
| 17: | **Return** |
| 18: | **End** |

* 1. **输入/输出示例**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs** | | | **Outputs** | | **Note** |
| a | b | c | x1 | x2 |
| 0 | 0 | 0 | - | - | any complex number |
| 0 | 0 | 1 | - | - | no root |
| 0 | 2 | 5 | -2.50000 | - | 一元一次方程 |
| 1 | 4 | 2 | -0.58579 | -3.41421 | 判别式大于0 |
| 1 | 2 | 1 | -1.00000 | -1.00000 | 判别式等于0 |
| 1 | 2 | 3 | 1.00000+1.41421i | 1.00000-1.41421i | 判别式小于0 |

* 1. **用户手册**
* 本程序用于求解实系数方程；
* 当运行程序时，在交互界面会弹出提示语：“Please input the coefficient:”。请用键盘依次输入3个实数，中间用逗号或者空格隔开，例如输入“2,3 5”，即代表a=2,b=3,c=5；
* 输入完毕后请按回车键运行程序，随后对应的提示语及结果将会输出到屏幕上。

1. **求解24点**

本题源文件为Week01\_q2.f90

执行文件为Week01\_q2.exe

注：为了防止直接运行.exe文件时cmd在运行结束后直接关闭，在源文件的末尾增加了一行命令pause

* 1. **题目描述**

24点游戏是儿时玩的主要益智类游戏之一，玩法为：从一副扑克中抽取4张牌，对4张牌使用加减乘除中的任何方法，使计算结果为24。例如，2,3,4,6，通过( ( ( 4 + 6 ) - 2 ) \* 3 ) = 24，最快算出24者胜。请采用Fortran90编程求解24点游戏的解。

* 1. **解决方案描述**

本题目采用穷举法进行求解：

1 先从四个数中任取两个数（记为a和b），对这两个数进行四则运算，得到的数与剩下的两个数记为x, y, z；

2 再从x, y, z这三个数中任取两个数，对这两个数进行四则运算，得到的数，与剩下的一个数，一共剩两个数

3 将这两个数进行四则运算，若结果为24，则输出

* 1. **伪代码**

|  |  |
| --- | --- |
| Algorithm 2 求解24点 | |
| ***Input***: four integer numbers | |
| ***Output***: the answer of point 24 in type character | |
| 1: | **For** every combination of and in **Do** |
| 2: | //存储第一次计算的值 |
| 3: |  |
| 4: | **For** each in **Do** |
| 5: |  |
| 6: | **For** every combination of and **Do** |
| 7: |  |
| 8: |  |
| 9: | **For** each in **Do** |
| 10: | //z是剩下未计算的最后一个数 |
| 11: |  |
| 12: |  |
| 13: | **If** element in |
| 14: | **Then** print the relevant string in |
| 15: | **End** |
| 16: | **End** |
| 17: | **End** |
| 18: | **End** |
| 19: | **End** |

* 1. **输入输出示例**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inputs** | | | | **Outputs** | **Note** |
| 1 | 6 | 6 | 4 | ((( 1+ 4)\* 6)- 6) ((( 1+ 4)\* 6)- 6)  ( 6-(( 1- 4)\* 6)) ( 6-(( 1- 4)\* 6))  ((( 4- 1)\* 6)+ 6) ((( 4- 1)\* 6)+ 6) | 该程序中部分答案存在重合，这是我们采取的遍历方式导致的。程序中暂时还未去除这些重复的答案 |
| 2 | 5 | 3 | 7 | ( 3-(( 2- 5)\* 7)) (( 3\* 7)-( 2- 5))  ((( 5- 2)\* 7)+ 3) (( 3\* 7)+( 5- 2))  (( 5\* 3)+( 2+ 7)) ((( 5\* 3)+ 2)+ 7)  ((( 5\* 3)+ 7)+ 2) (( 2+ 7)+( 5\* 3))  ((( 3\* 7)- 2)+ 5) ( 5-( 2-( 3\* 7)))  ((( 3\* 7)+ 5)- 2) (( 3\* 7)-( 2- 5))  (( 5- 2)+( 3\* 7)) | 同上 |

* 1. **用户手册**
* 本程序用于计算24点；
* 当运行程序时，在交互界面会弹出提示语：“Please input four numbers(range 1~13):”。请用键盘依次输入4个数（允许重复），中间用逗号或者空格隔开，例如输入“2,3 5,8”，则代表输入到数组中的数分别为2，3，5，8
* 输入完毕后请按回车键运行程序，随后所有满足情况式子将会输出到屏幕上；
* 部分答案会存在重复的情况，希望这不会给您造成困扰。