



文 / emi

称球问题是一个非常经典的逻辑算法题，如果读者能充分掌握其基本思路，完全可在此基础上编制一些逻辑判别程序。本文笔者将结合程序来介绍该算法的 C 语言实现。

一：问题内容

现有一个无砝码的天平，要求称 3 次从 12 个球中寻找一个与其它球轻重不同的坏球，并确定该球与其它球相比的轻重情况。该问题是很经典的一种称球问题，对逻辑判断能力有一定的要求。

二：解题思路和流程图

本题的具体解决思路如下所述：

- 1、将球依次编号为 1—12
- 2、将 1—4 号球和 5—8 号球分别放在天平上称，如果平衡，则坏球在 9—12 中；如果不平衡，假设左轻右重，则坏球在 1—4 中（轻）或 5—8 中（重）。
- 3、假设坏球在 9—12 中：将 9—11 和 3 个好球分别放在天平上称，如果平衡，则坏球为 12，再称一次即可判断其轻重；如果不平衡，将 9，10 分别放在天平两端称即可。
- 4、假设坏球在 1-8 中：天平左边放三个好球和 5，右边放 4 和 6，7，8，如果平衡，则坏球在 1，2，3 中；如果仍保持左轻右重，则坏球在 6，7，8 中；如果天平反转，左重右轻，则坏球在 4（轻）或 5（重）中。
- 5、根据以上的操作，最后一步均可轻松称出坏球和轻重情况。

总求解过程流程图，如图 1 所示：

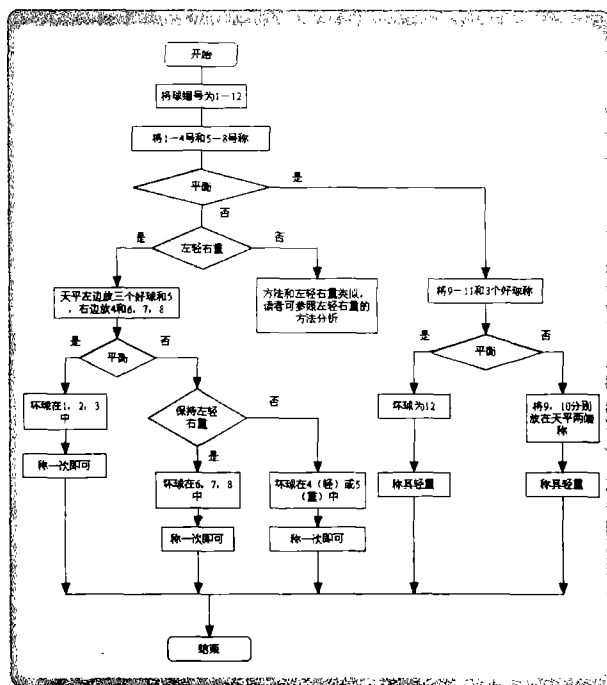


图 1

三：程序源代码分析

本程序使用 Turbo C 2.0 编制。

首先声明一些必要的头文件，代码如下所示：

```
#include<stdio.h>
```

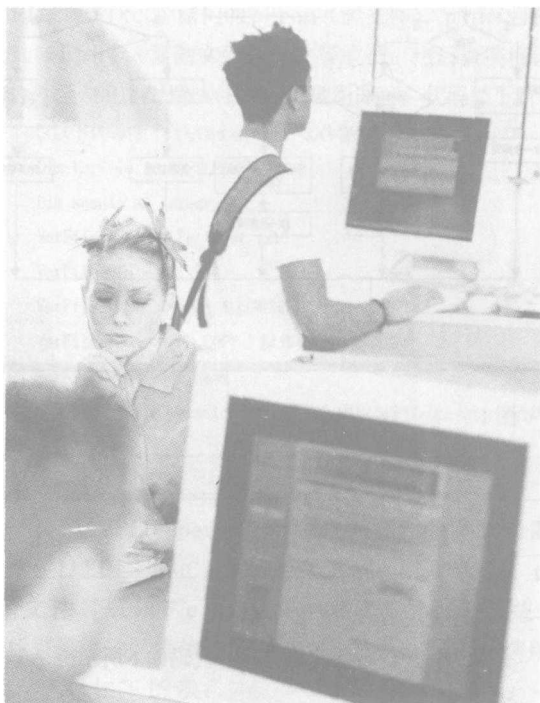
声明一些功能函数，代码如下所示：

```
int weight(int left[],int right[],int n);
void bad(int x,int b);
void onew(int a);
void twow(int a1,int a2,int b1,int b2);
void threew(int a1,int a2,int a3,int b);
void fourw(int a1,int a2,int a3,int a4);
void twelvew(int a1,int a2,int a3,int a4,int b1,int b2,int b3,
int b4,int c1,int c2,int c3,int c4);
int num=0;
```

然后按流程图编制主函数和各功能函数，读者可结合注释分析，代码如下所示：

```
void main() {
    printf("现有编号为1-12的球，已知有一个坏球和其它11个球的重量不同。按以下步骤，用天平称3次即可找出坏球：\n");
    twelvew(1,2,3,4,5,6,7,8,9,10,11,12);
}

int weight(int left[],int right[],int n)
{ int t;//输入的数值
  int i;//天平左右两边的球数
  printf("第%d次：\n",num);
  printf("将编号为");
  for(i=0;i<n;i++)
    printf("%d",left[i]);
  printf("的球放于天平左边，将编号为");
  for(i=0;i<n;i++)
    printf("%d",right[i]);
  printf("的球放于天平右边。");
  printf("请选择天平状态，0：相等，1：左重右轻，-1：左轻右重。");
```



```
scanf("%d",&t);
while(t<-1||t>1)
{ printf("\n请输入0,1,-1三个数中的一个！0：相等，1：左重右轻，-1：左轻右重。请选择：");
  scanf("%d",&t);
}
printf("\n");
return t;
}

//判断坏球及该球的轻重
void bad(int x,int b)
{ bool h=true;//判断是否有解
  char *p;//判断坏球的轻重
  switch(b)
  {case 0:
    p="不可能相等！";
    h=false;
    break;
  case 1:
    p="重";
    break;
  case -1:
    p="轻";
    break;
  default:
    break;
  }
  if(h)
    printf("坏球是第%d个球，且其比其它球%s\n",x,p);
}

//已知一个球是坏球，但不知轻重
void onew(int a)
{ int l[]={a};
  int r[]={0};
  int t=weight(l,r,1);
  bad(a,t);
}

//已知坏球在2个球之中，一轻一重
void twow(int a1,int a2,int t1,int t2)
{ int l[]={a1};
  int r[]={0};
  int t=weight(l,r,1);
  switch(t+t1)
  {
  case 1:case -1:
    bad(a2,t2);
    break;
  case 2:case -2:
    bad(a1,t1);
    break;
  }
```

```

case 0:
printf("不可能出现此情况！");
break;
default:
break;
}
}

// 已知坏球在三个球之中，且知轻重
void threew(int a1,int a2,int a3,int t1)
{ int t;
  int l[]={a1};
  int r[]={a2};
  t=weight(l,r,1);
  switch(t+t1)
  { case 1:case -1:
    bad(a3,t1);
    break;
    case 2:case -2:
    bad(a1,t1);
    break;
    case 0:
    bad(a2,t1);
    break;
    default:
    break;
  }
}

// 已知坏球在4个球之中，但不知轻重
void fourw(int a1,int a2,int a3,int a4)
{ int t;
  int l[]={a1,a2,a3};
  int r[]={0,0,0};
  t=weight(l,r,3);
  switch(t)
  {case 0:
    onew(a4);
    break;
    case 1:
    threew(a1,a2,a3,1);
    break;
    case -1:
    threew(a1,a2,a3,-1);
    break;
    default:
    break;
  }
}

//12 球中有一个坏球，且不知轻重
void twelvew(int a1,int a2,int a3,int a4,int b1,int b2,int b3,int
b4,int c1,int c2,int c3,int c4)

```

```

{ int good1[]={c1,c2,c3,c4};
  int good2[]={a1,a2,a3,a4,b1,b2,b3,b4};
  int i,t,t2;
  int l[]={a1,a2,a3,a4};
  int r[]={b1,b2,b3,b4};
  int l1[]={0,0,0,0,b1};
  int r1[]={a4,b2,b3,b4};
  t=weight(l,r,4);
  printf("好球为：");
  switch(t)
  { //天平平衡则称后四个球
    case 0:
    for(i=0;i<8;i++)
    {printf(good2[i]<<"，");
    printf("用0号球代表好球。\\n");
    fourw(c1,c2,c3,c4);
    break;
    //天平不平衡则称前8个球
    case 1:case -1:
    for(i=0;i<4;i++)
    {printf("%d，",good1[i]);
    printf("用0号球代表好球。\\n");
    t2=weight(l1,r1,4);
    switch(t+t2){
    //t2=0 即天平平衡
    case 1:case -1:
    threew(a1,a2,a3,t);
    break;
    //t2=t 即天平保持状态不变
    case 2:case -2:
    threew(b2,b3,b4,-t);
    break;
    //t2=-t 即天平反转
    case 0:
    twow(a4,b1,t,-t);
    break;
    default:
    break;
    }
    break;
  }
}
}

```

结束语

怎么样？一个经典的称球问题解决了。有兴趣的读者可在本程序的基础上添加一些模块，即可解决37个球称4次的问题。笔者希望本文介绍的程序能为广大读者编写复杂的逻辑程序打下坚实的基础。