2895讲解

循环小数：

给定一个分数，判断其是否是一个无限循环小数，并输出它的第一个循环节。例如：分数1/3是一个无限循环小数，第一个循环节为3；而1/2不是一个无限循环小数。

- 输入

不多于100行，每行一个m/n形式的分数（0＜ m＜ n＜ 100000）

- 输出

对于每一个分数，当其是一个无限循环小数时，输出它的第一个循环节；否则输出0。每行的最后有一个换行符。

2895讲解

- **思路**

  以4/7的计算过程为例

  4/7=0余4，即结果为0余数为4；

  4*10=40、40/7=5余5，即结果为0.5余数为5；

  5*10=50、50/7=7余1，即结果为0.57余数为1；

  1*10=10、10/7=1余3，即结果为0.571余数为3；

  3*10=30、30/7=4余2，即结果为0.5714余数为2；

  2*10=20、20/7=2余6，即结果为0.57142余数为6；

  6*10=60、60/7=8余4，即结果为0.571428余数为4；

  由于在计算过程中余数4出现过，后面的计算过程即将重复，即找到了循环节571428。

2895讲解

- 定义存储数组

```
int  main()
{
    int *remainder,*quotient;   //定义指向存储余数和商的数组的指针
}
```

2895讲解

- 输入

```
#include<stdio.h>
int  main()
{
    int m,n;
    int *remainder,*quotient;                  //定义存储余数和商的数组指针
    while(scanf("%d/%d",&m,&n)==2)  //当输入两个整数开始循环
    {
        if(m<=0 || n<=0)                       //如果输入为负数结束循环
            break;
    }
}
```

2895讲解

- 动态分配存储空间

```
#include<stdio.h>
int  main()
{
    int m,n;
    int *remainder,*quotient;              //定义存储余数和商的数组指针
    while(scanf("%d/%d",&m,&n)==2) //当输入两个整数开始循环
    {
        if(m<=0 || n<=0)                   //如果输入为负数结束循环
          break;
        remainder = new int[n];
        quotient = new int[n];
        delete remainder;
        delete quotient;
    }
}
```

2895讲解

- 初始化数组

```
#include<stdio.h>
int  main()
{
        int m,n;
        int *remainder,*quotient;            //定义存储余数和商的数组指针
        while(scanf("%d/%d",&m,&n)==2) //当输入两个整数开始循环
        {
              if(m<=0 || n<=0)                  //如果输入为负数结束循环
                    break;
              remainder = new int[n];
              quotient = new int[n];

              for(i=0;i<n;i++)
              {
                    remainder[i] = quotient[i] = -1;
               }

             // 计算

              delete remainder;
              delete quotient;
        }
}
```

2895讲解

- 计算思路与早前4/7的例子一致，以quotient[i]存储小数点后第i位的商，并通过remainder[i]来判断在第i位上是否是另一个循环节的开始。首先遍历整个数组，并将当前索引i存于remainder[m]

```
//计算
for(i=0;i<=n;i++)
{
        remainder[m]=i;
}
```

2895讲解

- **开始计算**

```
//计算
for(i=0;i<=n;i++)
{
        remainder[m]=i;
        m*=10;                  //移位
        quotient[i]=m/n;        //求商并存储结果
        m=m%n;                  //求余并赋值给m
}
```

2895讲解

- **当非循环小数输出0**

```
//计算
for(i=0;i<=n;i++)
{
        remainder[m]=i;
        m*=10;                          //移位
        quotient[i]=m/n;                //求商并存储结果
        m=m%n;                          //求余并赋值给m
        if(m==0){ printf("0\n");break; } //非循环小数
}
```

2895讲解

- **当余数重复输出循环节**

```
//计算
for(i=0;i<=n;i++)
{
        remainder[m]=i;
        m*=10;                          //移位
        quotient[i]=m/n;                //求商并存储结果
        m=m%n;                          //求余并赋值给m
        if(m==0){ printf("0\n");break; } //非循环小数
        if(remainder[m]!=-1)            //当余数重复输出循环节
        {
                for(j=remainder[m];j<=i;j++)
                        printf("%d",quotient[j]);
                printf("\n");
                break;
        }
}
```