

3024讲解

八进制小数:

写一个程序将0与1（不包括0和1）之间的八进制数转化为等价的十进制数。例如，八进制数0.75就是十进制数0.953125($7*8^{-1}+5*8^{-2}$)。小数点右边有n位的所有八进制数都可以转化为小数点右边至多有 $3*n$ 位的十进制数。

3024讲解

- 思路

十进制转八进制相当于将十进制小数点后第一位数除以8一次的商，加上第二位除以8两次后的商，再加上第三位除以8三次后的商。。。

例如，八进制数0.75转十进制数相当于

$$7/8 + 5/8^2 = (5/8 + 7)/8 = 0.953125$$

因此，在实现过程中可以通过 k 次除以8的递归运算求得，

$$\begin{array}{ll} \text{第1次} & d_k / 8 \\ \text{第2次} & (d_k / 8 + d_{k-1}) / 8 \\ \vdots & \vdots \\ \text{第}k\text{次} & (((d_k / 8 + d_{k-1}) / 8 + d_{k-2}) / 8 + \dots + d_1) / 8 \end{array}$$

3024讲解

- 输入和输出

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

```
int main()
```

```
{
```

```
    char d10[300],d8[100];
```

```
//存放十进制和八进制的字符数组
```

```
    int cas,i;
```

```
    scanf("%d",&cas);
```

```
//输入问题数
```

```
    for(i=0;i<cas;i++)
```

```
    {
```

```
        scanf("%s",d8);
```

```
//输入八进制小数
```

```
        printf("case #%d:\n",i);
```

```
//输出问题号
```

```
        printf("0.%.s\n",d10);
```

```
//输出十进制小数
```

```
    }
```

```
    return 0;
```

```
}
```

3024讲解

- 八进制数和十进制数的长度

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

```
int main()
{
```

```
    char d10[300],d8[100];
```

```
//存放十进制和八进制的字符数组
```

```
    int cas,l, len, index;
```

```
    scanf("%d",&cas);
```

```
//输入问题数
```

```
    for(i=0;i<cas;i++)
```

```
    {
```

```
        scanf("%s",d8);
```

```
//输入八进制小数
```

```
        len=strlen(d8)-1;
```

```
//八进制小数最右数位的索引
```

```
        index=0;
```

```
        /*八进制数转十进制数*/
```

```
        d10[index]='\0';
```

```
        printf("case #%d:\n",i);
```

```
//输出问题号
```

```
        printf("0.%s\n",d10);
```

```
//输出十进制小数
```

```
    }
```

```
    return 0;
```

```
}
```

3024讲解

- 八进制数和十进制数的长度

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

```
int main()
```

```
{
```

```
    char d10[300],d8[100];
```

```
//存放十进制和八进制的字符数组
```

```
    int cas,l, len, index;
```

```
    scanf("%d",&cas);
```

```
//输入问题数
```

```
    for(i=0;i<cas;i++)
```

```
    {
```

```
        scanf("%s",d8);
```

```
//输入八进制小数
```

```
        len=strlen(d8)-1;
```

```
//八进制小数最右数位的索引
```

```
        index=0;
```

```
        /*八进制数转十进制数*/
```

```
        d10[index]='\0';
```

```
        printf("case #%d:\n",i);
```

```
//输出问题号
```

```
        printf("0.%s\n",d10);
```

```
//输出十进制小数
```

```
    }
```

```
    return 0;
```

```
}
```

3024讲解

/*八进制数转十进制数*/

```
for(;len!=1;--len)           //遍历八进制小数点右侧从右到左每一位数字
{
    num=d8[len]-'0';         //计算当前位数值
    j=0;
    for(;j<index||num;++j) //未遍历完现有十进制小数数值或未完成整除时循环
    {
        //除以八的递归运算
        temp=num*10+(j<index?d10[j]-'0':0); //当第j位之前有值时相加
        d10[j]=temp/8+'0';
        num=temp%8;
    }
    index=j;                  //存储当前十进制小数点右边位数
}
```

3024讲解

- 最终代码

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

```
int main()
{
    char d10[300],d8[100];           //存放十进制和八进制的字符数组
    int cas,l,j, len, index, temp,num;
    scanf("%d",&cas);                //输入问题数
    for(i=0;i<cas;i++)
    {
        scanf("%s",d8);              //输入八进制小数
        len=strlen(d8)-1;            //八进制小数最右数位的索引
        index=0;
        /*八进制数转十进制数*/
        d10[index]='\0';
        printf("case #%d:\n",i);     //输出问题号
        printf("0.%.s\n",d10);       //输出十进制小数
    }
    return 0;
}
```