

### 【2.4.1.1 The Decoder】

#### 参考程序

```
#include <stdio.h>
#include <string.h>
char string[10005];
int main()
{
    while ( scanf("%s", &string) != EOF ) {    //输入字符串，直到文件结束
        int len = strlen(string);            //字符串长度为 len
        for ( int i = 0 ; i < len ; ++ i )    //根据编码方案，输入字符转化为输出字符
            printf("%c", string[i]+'*'- '1');
        printf("\n");        //字符串结束，换行
    }
    return 0;
}
```

### 【2.4.1.2 Above Average】

#### 参考程序

```
#include <iostream>
using namespace std;
int main()
{
    int C, N, tot_gra, abo_c, grade[1000];
    float average, perc;
    int i;
    cin >> C;          //C: 测试用例数
    while(C--)
    {
        cin >> N;        // N: 班级人数
        tot_gra = 0;      // 班级总分 tot_gra, 初始化为 0
        for(i = 0; i < N; i++)    //输入每个学生成绩, 累计总分
        {
            cin >> grade[i];
            tot_gra += grade[i];
        }
        average = tot_gra / N;    //计算平均成绩 average
        abo_c = 0;                //高于平均成绩的人数 abo_c, 初始化为 0
        for(i=0; i < N; i++)        //计算高于平均分的人数
            if(grade[i] > average)
                abo_c ++;
        perc = abo_c * 100.0 / N;    //计算高于平均成绩人数的百分比 perc
        printf("%.3f%%\n", perc);
    }
    return 0;
}
```

### 【2.4.2.1 Square Numbers】

#### 参考程序

```
#include <iostream>
using namespace std;
const int N = 1e5; //本题的数据范围
int prefixsum[N + 1] = {0}; //前缀和数组元素 prefixsum[i] 定义为 0-i 区间平方数的个数
int main()
{
    for(int i = 1, j = 1; i <= N; i++) //离线计算前缀和数组元素 prefixsum[i]
        if(i == j * j) {
            prefixsum[i] = prefixsum[i - 1] + 1;
            j++;
        } else
            prefixsum[i] = prefixsum[i - 1];
    int a, b; //a 和 b 如本题描述
    while(~scanf("%d%d", &a, &b) && (a || b)) //输入 a 和 b, 每次循环处理一个测试用例
        printf("%d\n", prefixsum[b] - prefixsum[a - 1]); //a 和 b 之间有多少个平方数
    return 0;
}
```

#### 【2.4.2.2 Ugly Numbers】

---

##### 参考程序

```
#include<iostream>
using namespace std;
int main(){
    long long a[1501] = {0,1}; //丑陋数数组  $a$  赋初值
    int p2 = 1, p3 = 1, p5 = 1; //指针  $p2$ ,  $p3$  和  $p5$  赋初值
    for (int i =2; i<=1500; i++){
        a[i] = min(a[p2]*2, min(a[p3]*3, a[p5]*5)); //下一个丑陋数
        if(a[i]==2*a[p2])p2++; //相应的指针加 1
        if(a[i]==3*a[p3])p3++;
        if(a[i]==5*a[p5])p5++;
    }
    int n;
    while(cin>>n, n) cout<<a[n]<<endl; //根据测试数据  $n$ ，输出  $a[n]$ 
    return 0;
}
```

### 【2.4.3.1 B2-Sequence】

---

#### 参考程序

```
#include <iostream>
using namespace std;
int main()
{
    int data[101]={ 0 }, n, T = 1;
    while (~scanf("%d",&n)) {
        int b2=0;
        for (int i = 1 ; i <= n ; ++ i) {
            scanf("%d", &data[i]);
            if (data[i]<=data[i-1]) b2=1;
        }
        int sum[20001]={ 0 };
        if (b2==0)
            for (int i = 1 ; i <= n ; ++ i) //枚举所有的两两之和  $b_i+b_j$ 
                for (int j = i ; j <= n ; ++ j) {
                    if (sum[data[i]+data[j]] !=0) b2=1;
                    sum[data[i]+data[j]]=1;
                }
        if (!b2)
            printf("Case #%d: It is a B2-Sequence.\n\n",T ++);
        else
            printf("Case #%d: It is not a B2-Sequence.\n\n",T ++);
    }
    return 0;
}
```