

试题讲解

试题类型

	第1题	第2题	第3题	第4题
第1套	数组	模拟	穷举	搜索
第2套	模拟	字符串	贪心	搜索
第3套	模拟	字符串	贪心	搜索
第4套	贪心	字符串	搜索	动规LIS

第一套试题

1.国王的米粒（gwdml.pas/c/cpp）

这是一个有名的古代故事。有一个数学家发明了一种棋盘献给了印度国王，数学家看国王非常欢喜，就向国王提出了奖赏的要求：在棋盘的第一格放一粒米，第二格放二粒米，第三格放四粒米，第四格放八粒米，. . . . 也就是说每一格都放进了比前一格多一倍的米。国王认为这简直不值一提，就毫不犹豫的答应了。谁知结果却让国王大吃一惊，当放到第**64**格时，就已经一共用了**18446744073709551615**粒米。这在当时要几百年才能种出来。

现假定该棋盘共有**30**格，请你编程计算从第**m**格至第**n**格共有多少粒米。

第一套试题

1.国王的米粒（gwdml.pas/c/cpp）

【输入格式】

输入文件gwdml.in包含一行，包含两个正整数m,n（ $1 \leq m, n \leq 30$ ）。

【输出格式】

输出文件 gwdml.out包含一行，包含一个整数，表示从第m格至第n 格共有多少米粒。

【输入样例】

1 10

【输出样例】

1023

【数据范围】

对于 100%的数据， $1 \leq m, n \leq 30$ 。

第一套试题

1.国王的米粒（gwdml.pas/c/cpp）

思路： 1.求出第m格所对应的米粒数；

```
a[1]=1;  
for (int i=1;i<m;i++)  
    a[i+1]=a[i]*2;  
//或者a[m]=pow(2,m-1)
```

2.求出第m格到第n格米粒数之和；

```
for (int i=m;i<=n;i++)  
{  
    sum+=a[i];  
    a[i+1]=a[i]*2;  
}
```

第一套试题

2.bill (bill.pas/c/cpp)

大家都知道，高三的同学很辛苦，需要补充很多营养。但是由于 **CZY** 高三教室在 4 楼 和 5 楼，而高一教室在 1 楼和 2 楼，所以导致高三同学总是很难到想吃的东西。又由于 **D**， **G**， **Z** 的动作总是很慢，所以想吃到不错的午餐更是艰难。 所以自从不用上课，**X** 就承担下了代购午餐的任务。

但是由于 **X** 不上晚自习，不能代 购晚餐，所以大家商议决定每周轮流留一张饭卡给 **X**。 可是计划赶不上变化，期间出现了忘换饭卡之类的一系列问题，导致了账目的混乱， 于是 **X** 又承担下了记录账目的任务。 每天被账目搞得一团糟的 **X** 想请你帮她写一个程序，来清楚地记账。

第一套试题

2.bill (bill.pas/c/cpp)

【输入格式】

第 1 行一个整数 n ($0 < n \leq 10000$)，表示需要记录账目天数。

第 2 至 $n + 1$ 行每行两个字母（只含 D, G, Z），表示前一个同学欠后一个同学一顿饭，字母之间有一个空格。

【输出格式】

第 1 行 D x1

第 2 行 G x2

第 3 行 Z x3

x_i 为正数是表示该同学欠了别人 x_i 顿；0 表示该同学既不欠人，也不被欠；负数表示该同学借给别人 x_i 顿。

第一套试题

2.bill (bill.pas/c/cpp)

【输入样例】

5

D G

G D

G Z

D G

D Z

【输出样例】

D 2

G 0

Z -2

【数据范围】

对于 100% 的数据， $1 \leq m, n \leq 30$ 。

第一套试题

2.bill (bill.pas/c/cpp)

思路：根据题目要求，输入每行两个字母（只含 D，G，Z），表示前一个同学欠后一个同学一顿饭。输出为每个同学欠或者被欠数量（正负号）。那么可以创建一个数组 **a**，来表示对应同学欠饭的数量，输入的前一个字母，则数组 **a** 对应下标的值加1，反之则减1。

```
for (int i=1;i<=n;i++)  
{  
    cin>>x>>y;  
    a[x]+=1;  
    a[y]-=1;  
  
}
```

第一套试题

3.迟到的生日(shlqsh.pas/c/cpp)

今天是 `wkn` 同学的生日啊，于是呢，很多人都去找机会去祝他生日快乐了。

但是呢，现在有 1000 万的无知的小盆友，他们居然忘记了 `wkn` 的生日，这个显然是不可饶恕的。当然他们也意识到了这个事情的严重性，于是他们开始想办法弥补了，然后他们被编号了，从 1 到 1000 万。然后编号为 k 的小盆友，需要到 `wkn` 家去面壁思过 p 天， p 是 k 的约数个数。现在呢，`wkn` 家同时只能容纳一个小盆友面壁，所以呢，你需要计算编号 $t1$ 到 $t2$ 这么多个小盆友至少需要面壁多少天。问题是给出数据 $t1, t2$ 后，求这么多小盆友需要面壁多少天呢？

第一套试题

3.迟到的生日(shlqsh.pas/c/cpp)

【输入样例】

2 6

【输出样例】

13

【样例说明】

2 的约数有 1,2 (他需要面壁 2 天);

3 的约数有 1,3 (他需要面壁 2 天);

4 的约数有 1,2,4 (他面壁 3 天);

5 的约数有 1,5 (他只要 2 天);

6 的约数有 1,2,3,6 (去了 4 天); 总共面壁 13 天。

【数据范围】

对于 50%的数据, $1 \leq t_1 \leq t_2 \leq 1000$

对于 100%的数据, $1 \leq t_1 \leq t_2 \leq 10,000,000$

第一套试题

3.迟到的生日(shlqsh.pas/c/cpp)

思路：穷举 t_1 - t_2 之间所有的数，判断每个数的约数有几个，然后相加，输出最后的答案：

```
for (int i=t1;i<=t2;i++)  
{  
    int t=0;  
    for (int j=1;j*j<=i;j++)  
        if (i % j == 0)  
        {  
            t++;  
        }  
}
```



时间复杂度
?

对于 50%的数据， $1 \leq t_1 \leq t_2 \leq 1000$

对于 100%的数据， $1 \leq t_1 \leq t_2 \leq 10,000,000$

第一套试题

3.迟到的生日(shlqsh.pas/c/cpp)

思路2：穷举所有的约数 i ，找出 $t1-t2$ 之间有多少个数含有该约数 i ：

问题：对于某个数 ti 来说，如何判断 $t1-t2$ 之间有多少个数能够被 ti 整除？

当 $t1=1$ ； $t2=5$ ；

对于约数1： $t1-t2$ 之间有5个数能被1整除：1 2 3 4 5；

对于约数2： $t1-t2$ 之间有2个数能被1整除：2 4；

对于约数3： $t1-t2$ 之间有1个数能被1整除：3；

对于约数4： $t1-t2$ 之间有1个数能被1整除：4；

对于约数5： $t1-t2$ 之间有1个数能被1整除：5；

```
核心程序 for(i=1;i<=t2;i++)  
ans+=t2/i-(t1-1)/i;
```

第一套试题

4.家族(family.pas/c/cpp)

在一个与世隔绝的岛屿上，有一个有趣的现象：同一个家族的人家总是相邻的（这里的 相邻是指东南西北四个方向），不同的家族之间总会有河流或是山丘隔绝，但同一个家族的人不一定有相同姓氏。现在给你岛上的地图，求出岛上有多少个不同的家族。岛上的地图有 n 行，每行有若干列，每个格子中要么是‘空格’，表示大海，要么是‘*’，表示河流或山丘，要么是小写字母，表示一户人家的姓氏。

第一套试题

4. 家族(family.pas/c/cpp)

【输入格式】

第一行是个数字 N，表示下面信息的行数；

第二行只第n+1行，每行由小写字母和*号组成，有些行的最前面也可能包含若干连续的空格，表示这些区域是大海，每一行最多不超过 200 个字符。

【输出格式】

包含一个数字，表示家族数。

【输入样例】

```
4
*zlw**pxh
l*zlwk*hx*
w*tyy**yyy
  zzl
```

【输出样例】

```
3
```

第一套试题

4. 家族(family.pas/c/cpp)

【数据范围】

10%的数据， $n \leq 1$

30%的数据， $n \leq 10$

100% 的数据， $n \leq 100$ 每一行最多不超过 200 个字符

第一套试题

4.家族(family.pas/c/cpp)

思路：根据题目要求，本题的算法模型是搜索的连通块问题，给定的地图中求出所有连通块的数量。

本题需解决的数据处理：

1.姓氏是各种小写字母，可以在输入过程中统一赋值1或者0；大海、河流和山丘则反之赋值；

2.处理输入的字符串；

数据处理好后，则按照求连通块的方法来求解答案。

第一套试题

4. 家族(family.pas/c/cpp)

程序实现:

1. 数据处理:

字符读入并赋值:

```
for (int i=1;i<=n;i++)  
{  
    c=getchar();  
    while (c!='\n')  
    {  
        s[i]++;  
        if (c>='a' && c<='z')  
            mapz[i][s[i]]=true;  
        c=getchar();  
    }  
}
```

第一套试题

4. 家族(family.pas/c/cpp)

程序实现:

1. 数据处理:

字符串读入并赋值:

```
for (int i=1;i<=n;i++)
```

```
{
```

```
    getline(cin,c);
```

```
    s[i]=c.length();
```

```
    for (int j=0;j<c.length();j++)
```

```
        if (c[j]>='a' && c[j]<='z')
```

```
            mapz[i][j]=true;
```

```
}
```

第一套试题

4.家族(family.pas/c/cpp)

程序实现:

2.运用搜索思想处理连通块:

(1)将坐标作为搜索的状态;

(2)找到一个姓氏 (即`mapz[i][j]==1`)就开始搜索和它相连接的姓氏, 并将家族总数加1。

```
for (int i=1;i<=n;i++)  
    for (int j=1;j<=s[i];j++)  
    {  
        if (mapz[i][j])  
        {  
            ans++;  
            dfs(i,j);  
        }  
    }  
}
```

第一套试题

4.家族(family.pas/c/cpp)

程序实现:

```
void dfs(int x,int y)
{
    int x1,y1;
    mapz[x][y]=false;
    for (int i=0;i<4;i++)
    {
        x1=x+fx[i];
        y1=y+fy[i];
        if (x1>0 && y1>0 && x1<=n && y1<=s[x1]
&& mapz[x1][y1])
            dfs(x1,y1);
    }
}
```