

第三套试题

1.排队买票(ticket.pas/c/cpp)

暑假来临，fm和他的朋友们外出游玩，当他们到达某一景点时，发现该景点的售票点已经排了长队，他们不想错过这个景点，无奈只能默默排队。幸运的是，他们凭借敏锐的观察力发现队伍虽然看起来长，但是有许多人是一起的，一个紧挨着一个，就像fm他们一样。

当最前面的人买完票之后，他们就会一起离开队伍。fm和他的朋友们很好奇如果把队伍看成若干的组，如果把那些一起的人看成一组的话，那么fm他们那组在队伍里排第几组呢。很巧合的是你就在他们后面，所以fm很自然的就找上了你，请你帮忙计算一下。

第三套试题

1.排队买票(ticket.pas/c/cpp)

对于组的定义这里详细说明下：

1.如果队伍里的某个人不认识他相邻的人那么他就可以看成一个组；

2.如果队伍里某个人的前或后有人认识他，那么所有这些相邻的人都可以看成一个组。怎么样才能知道哪些人是互相认识，能够成为一个“组”呢，细心的fm发现虽然不是穿同样颜色的人都认识，但是所有认识的人都会同样颜色的衣服，并且两个相邻的“组”的人是不会穿同样颜色的衣服，这点请你放心。

第三套试题

1.排队买票(ticket.pas/c/cpp)

【输入格式】

第一行包含一个整数N ($1 \leq N \leq 25$)，表示在fm前面有多少人。

接下来的N行，每行一个大写字母，表示该人穿的衣服的颜色。

【输出格式】

一个整数，表示fm他们组排在第几组。

【输入样例】

3
C
Z
P

【输出样例】

4

【数据范围】

对于 100% 的数据，n不超过 25。

第三套试题

1.排队买票(ticket.pas/c/cpp)

思路：模拟查找前面有多少相邻且不一样的块数，那么只需判断当前位置字符和下一位置字符是否相等，如果相等则序号向后移，直到找到不相等的位置，然后组数加1，接着位置也加1，再往复循环查找判断。

```
for (int i=0;i<n;)
{
    while (a[i]==a[i+1]) i++;
    if (i<=n) sum++;
    i++;
}
```

第三套试题

2.手机 (mobile.pas/c/cpp)

一般的老式手机的键盘是这样的：

1	2 abc	3 def
4 ghi	5 jkl	6 mno
7 pqrs	8 tuv	9 wxyz
*	0	#

要按出英文字母就必须按数字键多下。例如要按出 **x** 就得按 **9** 两下，第一下会出 **w**，而第二下会把 **w** 变成 **x**。**0** 键按一下会出一个空格。你的任务是读取若干句只包含英文小写字母和空格的句子，求出要在手机上打出这个句子至少需要按多少下键盘。

第三套试题

2.手机 (mobile.pas/c/cpp)

【输入格式】

包含一行，表示一个句子，句子中只包含英文小写字母和空格，且不超过 200 个字符。

【输出格式】

包含一行一个整数，表示按键盘的总次数。。

【输入样例】

i have a dream

【输出样例】

23

【数据范围】

对于 100% 的数据，不超过 200 个字符。

第三套试题

2.手机 (mobile.pas/c/cpp)

思路：模拟手机按键过程，对于输入的字符串s，从左至右进行判断该字母需要按几次，再进行累加得到最终答案。

tips: 对于每个字母按几次可以用**case**语句或者直接设置一个数组将字母与按键次数相对应。

```
for (int i=0;i<s.length();i++)  
{  
    if (s[i]==' ')  
        ans++;  
    else  
        ans+=num[s[i]-'a'];  
}
```

第三套试题

3.排队出发 (queue.pas/c/cpp)

神牛岛是传说中的—个岛屿，凡是成功到那里游历，完成探险并返回的人，都会成为神牛。但是，现实中却没有人知道如何到达神牛岛。

这天夜里，笃志者睡着之后，不久就进入了梦乡。他突然看到有人在问，“有人想去神牛岛的吗？”神牛岛之旅的牌子前，就开始有不少勇士报名要去冒险探索。

第三套试题

3.排队出发 (queue.pas/c/cpp)

“我们会把勇士安排在前，带领大家一起去神牛岛。下面开始点名！”管理队伍的 LXY 神牛说。其实说实话，给学生排队这种工作是最让神牛头疼的了。因为同学们都有自尊心，都不愿意排后面。共有 n 个同学要排成一列，每个同学有两个属性：影响力和承受能力。给一个同学造成的心理创伤指数等于所有在他前面同学的影响力之和减去他的承受能力。现在请你帮忙安排一下点名顺序，尽量使受到心理创伤最大的同学少受创伤。

第三套试题

3.排队出发 (queue.pas/c/cpp)

【输入格式】

第 1 行是整数 n ，表示同学的个数。

第 2~ $n+1$ 行每行两个自然数，分别是该同学的影响力和承受能力。

【输出格式】

包含 1 行，为你安排的顺序中受到心理创伤最大的同学受到的创 伤。

【输入样例】

```
3
10 3
2 5
3 3
```

【输出样例】

```
2
```

【数据范围】

对于 100% 的数据， $1 \leq n \leq 50000$ ， $1 \leq \text{影响力} \leq 10000$ ， $1 \leq \text{承受能力} \leq 1,000,000,000$ 。

第三套试题

3.排队出发 (queue.pas/c/cpp)

思路：题目要求：每个人的所受伤害定义为排在他前面的人的影响力之和减去自己的承受力，设每个人的影响力和承受力为 $s[i].x$ 和 $s[i].y$ 。

经典的贪心问题：使得受到伤害最大的人伤害最小。

给 n 个人排序，那么按照什么规则进行排序？



影响力？



承受力？

第三套试题

3. 排队出发 (queue.pas/c/cpp)

思路：定义i号人的影响力为 a_i ，承受力为 b_i

假定之前所有人都已用最优策略确定了顺序，考虑最后两个人i, j的顺序。

若为i、j，那么i所受伤害为 $\text{Sum}-b_i$ ，j所受伤害为 $\text{Sum}+a_i-b_j$

若为j、i，那么i所受伤害为 $\text{Sum}-b_j$ ，j所受伤害为 $\text{Sum}+a_j-b_i$ （Sum为之前所有人的影响力之和）

比较 $\max(\text{Sum}-b_i, \text{Sum}+a_i-b_j)$ 和 $\max(\text{Sum}-b_j, \text{Sum}+a_j-b_i)$ 大小，很明显 $\text{Sum}-b_i < \text{Sum}+a_j-b_i$ ， $\text{Sum}-b_j < \text{Sum}+a_i-b_j$ ，所以我们只需要比较 $\text{Sum}+a_i-b_j$ 和 $\text{Sum}+a_j-b_i$ 的大小。

如果我们假定i放在前面更优，那么 $\text{Sum}+a_i-b_j < \text{Sum}+a_j-b_i$ 必定成立，可以得到 $a_i+b_i < a_j+b_j$ 。

第三套试题

3.排队出发 (queue.pas/c/cpp)

思路：那么我们只需要按 $a+b$ 为关键字升序排序即可得到最优排列，计算出最大的伤害值即为答案。

```
sort(s+1,s+n+1,cmp);
int tot=0,ans=-2100000000;
for (int i=1;i<=n;i++) {
    ans=max(ans,tot-s[i].y);
    tot+=s[i].x;
}
```

```
bool cmp(demo x,demo y) {
    if (x.x+x.y<y.x+y.y) return 1;
    if (x.x+x.y>y.x+y.y) return 0;
    if (x.x<y.x) return 1;
    return 0;
}
```

第三套试题

4. 迷宫 (maze.pas/c/c++)

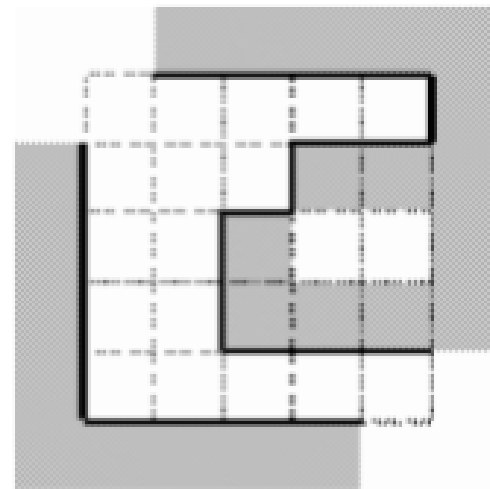
迷宫的管理员们决定在新开始的季节里使用新的墙纸。出于这个目的他们需要一个程序 来计算迷宫内墙壁的面积。这就是你即将要做的工作。 我们把这个迷宫用一个 $N \times N$ ($3 \leq N \leq 33$) 的矩阵表示。一些矩阵单元包含一个“.”（这代表一个空的方块），另一些矩阵单元包含一个“#”（这代表一个用巨石砌成的石墙 占据的方块）。全部方块的大小都为 3×3 平方米。

第三套试题

4. 迷宫 (maze.pas/c/c++)

墙壁由迷宫的四周（除了作为迷宫出入口的左上角和右下角以外）以及那些标记为“#”的矩阵单元构成，除此之外没有其他的墙。在输入的矩阵里左上角和右下角永远是一个“.”。你的任务是计算迷宫里可见部分的墙壁的面积。

换句话说，就是对迷宫的游客来说墙壁表面可见的部分。注意在两块相邻的石块之间没有空隙，即使两块石块在转角处相接触，我们都认为它们是相邻的。看看图示的例子：迷宫中可见的墙壁都用加粗的线条来描画。所有墙壁的高度都是三米。



第三套试题

4. 迷宫 (maze.pas/c/c++)

【输入格式】

第一行包含一个数字 N 。

接下来的 N 行每行都包含有 N 个字符。每行描述了迷宫 矩阵的一行。每行都只有“.”、“#”这两个字符并都以一个换行符结束。输入里没有任何的空格。

【输出格式】

包含一个整数，即所需要的壁纸的准确面积。

第三套试题

4. 迷宫 (maze.pas/c/c++)

【输入样例】

5

.....

...##

..#..

..###

.....

【输出样例】

198

【数据范围】

对于 100% 的数据, $3 \leq N \leq 33$

第三套试题

4.迷宫 (maze.pas/c/c++)

思路：典型的搜索问题，以入口和出口为起点遍历整张图，对于每个可到达的点，如果其上下左右有一堵墙则答案加1。因为是求面积，答案再乘上9。

tips: 为了便于处理，在整张图外围加一圈墙。最后记得减去4（出入口处没有墙，各减2）

第三套试题

4.迷宫 (maze.pas/c/c++)

程序实现：1.设置标记数组标记墙（包括四周）和走过的路

```
void dfs(int x,int y)
{
    if (B[x][y]) return;
    B[x][y]=1;
    for (int i=0;i<4;i++)
    {
        if (!M[x+dx[i]][y+dy[i]])
            cnt++;
        else
            if (!B[x+dx[i]][y+dy[i]])
                dfs(x+dx[i],y+dy[i]);
    }
}
```