[编程题] 字符串编码

时间限制：1秒

空间限制：32768K

给定一个字符串，请你将字符串重新编码，将连续的字符替换成“连续出现的个数+字符”。比如字符串AAAABCCDAA会被编码成4A1B2C1D2A。

**输入描述:**

每个测试输入包含1个测试用例

每个测试用例输入只有一行字符串，字符串只包括大写英文字母，长度不超过10000。

**输出描述:**

输出编码后的字符串

**输入例子1:**

AAAABCCDAA

**输出例子1:**

4A1B2C1D2A

[编程题] 最大和

时间限制：1秒

空间限制：32768K

在一个N\*N的数组中寻找所有横，竖，左上到右下，右上到左下，四种方向的直线连续D个数字的和里面最大的值

**输入描述:**

每个测试输入包含1个测试用例，第一行包括两个整数 N 和 D :

3 <= N <= 100

1 <= D <= N

接下来有N行，每行N个数字d:

0 <= d <= 100

**输出描述:**

输出一个整数，表示找到的和的最大值

**输入例子1:**

4 2

87 98 79 61

10 27 95 70

20 64 73 29

71 65 15 0

**输出例子1:**

193

[编程题] 推箱子

时间限制：1秒

空间限制：32768K

大家一定玩过“推箱子”这个经典的游戏。具体规则就是在一个N\*M的地图上，有1个玩家、1个箱子、1个目的地以及若干障碍，其余是空地。玩家可以往上下左右4个方向移动，但是不能移动出地图或者移动到障碍里去。如果往这个方向移动推到了箱子，箱子也会按这个方向移动一格，当然，箱子也不能被推出地图或推到障碍里。当箱子被推到目的地以后，游戏目标达成。现在告诉你游戏开始是初始的地图布局，请你求出玩家最少需要移动多少步才能够将游戏目标达成。

**输入描述:**

每个测试输入包含1个测试用例

第一行输入两个数字N，M表示地图的大小。其中0<N，M<=8。

接下来有N行，每行包含M个字符表示该行地图。其中 . 表示空地、X表示玩家、\*表示箱子、#表示障碍、@表示目的地。

每个地图必定包含1个玩家、1个箱子、1个目的地。

**输出描述:**

输出一个数字表示玩家最少需要移动多少步才能将游戏目标达成。当无论如何达成不了的时候，输出-1。

**输入例子1:**

4 4

....

..\*@

....

.X..

6 6

...#..

......

#\*##..

..##.#

..X...

.@#...

**输出例子1:**

3

11

[编程题] 赛马

时间限制：1秒

空间限制：32768K

在一条无限长的跑道上，有N匹马在不同的位置上出发开始赛马。当开始赛马比赛后，所有的马开始以自己的速度一直匀速前进。每匹马的速度都不一样，且全部是同样的均匀随机分布。在比赛中当某匹马追上了前面的某匹马时，被追上的马就出局。 请问按以上的规则比赛无限长的时间后，赛道上剩余的马匹数量的数学期望是多少

**输入描述:**

每个测试输入包含1个测试用例

输入只有一行，一个正整数N

1 <= N <= 1000

**输出描述:**

输出一个浮点数，精确到小数点后四位数字，表示剩余马匹数量的数学期望

**输入例子1:**

1

2

**输出例子1:**

1.0000

1.5000

1. //看看我的多简洁

#include <iostream>

#include <string>

using namespace std;

int main() {

    int i = 0, n = 1;

    string a;

    string b;

    cin >> a;

    while (a[i]) {

        if (a[i + 1] == a[i]) {

            n++;

        }

        else {

            cout << n << a[i];

            n = 1;

        }

        i++;

    }

    return 0;

}

2. #include<iostream>

#include<cmath>

using namespace std;

struct sum\_struct{

    int sum\_left;

    int sum\_left\_top;

    int sum\_top;

    int sum\_right\_top;

    sum\_struct(){

        sum\_left=0;

        sum\_left\_top=0;

        sum\_top=0;

        sum\_right\_top=0;

    }

};

int max\_func(int a, int b){

    return a>b?a:b;

}

int main(){

    int m,n;

    while(cin>>m>>n){

        // zero padding to deal with edge situations

        sum\_struct st[102\*102];

        int max\_sum=0, num;

        for(int i=0;i<m\*m;i++){

            cin>>num;

            int row = floor(i/m);

            int col = i%m;

            int offset\_i = (m+2)\*(row+1)+(col+1);

            // update sum

            st[offset\_i].sum\_left = num + st[offset\_i-1].sum\_left;

            st[offset\_i].sum\_top = num + st[offset\_i-(m+2)].sum\_top;

            st[offset\_i].sum\_left\_top = num + st[offset\_i-(m+2)-1].sum\_left\_top;

            st[offset\_i].sum\_right\_top = num + st[offset\_i-(m+2)+1].sum\_right\_top;

            // update max

            if(col>=n-1)

                max\_sum = max\_func(max\_sum, st[offset\_i].sum\_left - st[offset\_i-n].sum\_left);

            if(row>=n-1)

                max\_sum = max\_func(max\_sum, st[offset\_i].sum\_top - st[offset\_i-(m+2)\*n].sum\_top);

            if(col>=n-1 && row>=n-1)

                max\_sum = max\_func(max\_sum, st[offset\_i].sum\_left\_top - st[offset\_i-(m+2)\*n-n].sum\_left\_top);

            if(col<=m-n && row>=n-1)

                max\_sum = max\_func(max\_sum, st[offset\_i].sum\_right\_top - st[offset\_i-(m+2)\*n+n].sum\_right\_top);

        }

        cout<<max\_sum<<endl;

    }

    return 0;

}

3. #include <iostream>

#include <algorithm>

#include <vector>

#include <cstring>

#include <queue>

using namespace std;

typedef long long ll;

char mp[10][10];

int vis[10][10][10][10];

int data[4][2]={0,1,0,-1,1,0,-1,0};

    int n,m;

struct node{

    int x,y,step;

    int xx,yy;

    node(int \_x,int \_y,int \_step,int \_xx,int \_yy){x=\_x,y=\_y,step=\_step,xx=\_xx,yy=\_yy;}

};

int bfs(int sxx,int syy,int sx,int sy){

    queue<node> q;

    q.push(node(sx,sy,0,sxx,syy));

    while(!q.empty())

    {

        node s=q.front();

        q.pop();

        vis[s.x][s.y][s.xx][s.yy]=1;

        //cout<<s.x <<" "<<s.y<<" "<<s.xx<<" "<<s.yy<<" "<<s.step<<endl;

        if(mp[s.xx][s.yy]=='@') return s.step;

        for(int i=0;i<4;i++){

            node tmp=s;

            tmp.step++;

            tmp.x+=data[i][0];

            tmp.y+=data[i][1];

            if(tmp.x>n||tmp.y>m||tmp.x<1||tmp.y<1) continue;

            if(mp[tmp.x][tmp.y]=='#') continue;

            if(tmp.x==tmp.xx&&tmp.y==tmp.yy)

                tmp.xx+=data[i][0],tmp.yy+=data[i][1];

            if(tmp.xx>n||tmp.yy>m||tmp.xx<1||tmp.yy<1) continue;

            if(mp[tmp.xx][tmp.yy]=='#') continue;

            if(vis[tmp.x][tmp.y][tmp.xx][tmp.yy]!=1)

                q.push(tmp);

        }

    }

    return -1;

}

int main(){

    cin>>n>>m;

    for(int i=1;i<=n;i++)

        cin>>&mp[i][1];

    int sx,sy,xx,yy;

    for(int i=1;i<=n;i++)

    {

        for(int j=1;j<=m;j++)

        {

            if(mp[i][j]=='X')

            sx=i,sy=j;

            if(mp[i][j]=='\*')

            xx=i,yy=j;

        }

    }

    cout<<bfs(xx,yy,sx,sy)<<endl;

}

/\*

3 3

#.X

#\*.

<a href="/profile/523199" data-card-uid="523199" class="js-nc-card" target="\_blank" style="color: #25bb9b">@..

\*/

</a>