

青少年软件编程（C 语言）等级考试试卷（七级）

分数：100 题数：4

一、编程题(共 4 题，共 100 分)

1. 走出迷宫

当你站在一个迷宫里的时候，往往会被错综复杂的道路弄得失去方向感，如果你能得到迷宫地图，事情就会变得非常简单。

假设你已经得到了一个 $n*m$ 的迷宫的图纸，请你找出从起点到出口的最短路。

时间限制：1000

内存限制：65536

输入

第一行是两个整数 n 和 m ($1 \leq n, m \leq 100$)，表示迷宫的行数和列数。接下来 n 行，每行一个长为 m 的字符串，表示整个迷宫的布局。字符 '.' 表示空地，'#' 表示墙，'S' 表示起点，'T' 表示出口。

输出

输出从起点到出口最少需要走的步数。

样例输入

```
3 3
S#T
. #.
...
```

样例输出

```
6
```

试题编号：20230318-7-01

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：点击浏览

考生答案：（此题已作答）

考生得分：25

是否评分：已评分

评价描述：

考生答案：

```
#include <bits/stdc++.h>
```

```

using namespace std;
int n,m,p[10008],q[10008],ans[108][108],qx,qy,zx,zy;
char a[108][108];
int xx[5]= {0,1,-1,0,0};
int yy[5]= {0,0,0,1,-1};
void bfs() {
    a[qx][qy]='#';
    ans[qx][qy]=0;
    int head=1;
    int tail=1;
    p[1]=qx;
    q[1]=qy;
    while(head<=tail) {
        for(int i=1; i<=4; i++) {
            int h=p[head]+xx[i];
            int l=q[head]+yy[i];
            if(a[h][l]!='#'&&h>=1&&l>=1&&h<=n&&l<=m) {
                a[h][l]='#';
                tail++;
                p[tail]=h;
                q[tail]=l;
                ans[h][l]=ans[p[head]][q[head]]+1;
                if(h==zx&&l==zy) {
                    cout<<ans[h][l]<<endl;
                    return ;
                }
            }
        }
        head++;
    }
}
int main() {
    cin>>n>>m;
    for(int i=1; i<=n; i++) {
        for(int j=1; j<=m; j++) {
            cin>>a[i][j];
            if(a[i][j]=='S')
            {
                qx=i;
                qy=j;
            }
            if(a[i][j]=='T')
            {
                zx=i;

```

```

        zy=j;
    }
}
}
bfs();

return 0;
}

```

2. 拯救公主

多灾多难的公主又被大魔王抓走啦！国王派遣了第一勇士阿福去拯救她。

身为超级厉害的术士，同时也是阿福的好伙伴，你决定祝他一臂之力。你为阿福提供了一张大魔王根据地的地图，上面标记了阿福和公主所在的位置，以及一些不能够踏入的禁区。你还贴心地为阿福制造了一些传送门，通过一个传送门可以瞬间转移到任意一个传送门，当然阿福也可以选择不通过传送门瞬移。传送门的位置也被标记在了地图上。此外，你还查探到公主所在的地方被设下了结界，需要集齐 K 种宝石才能打开。当然，你在地图上也标记出了不同宝石所在的位置。

你希望阿福能够带着公主早日凯旋。于是在阿福出发之前，你还需要为阿福计算出他最快救出公主的时间。

地图用一个 $R \times C$ 的字符矩阵来表示。字符 S 表示阿福所在的位置，字符 E 表示公主所在的位置，字符 $\#$ 表示不能踏入的禁区，字符 $\$$ 表示传送门，字符 $.$ 表示该位置安全，数字字符 0 至 4 表示了宝石的类型。阿福每次可以从当前的位置走到他上下左右四个方向上的任意一个位置，但不能走出地图边界。阿福每走一步需要花费 1 个单位时间，从一个传送门到达另一个传送门不需

要花费时间。当阿福走到宝石所在的位置时，就视为得到了该宝石，不需要花费额外时间。

时间限制：1000
内存限制：65536

输入

第一行是一个正整数 T ($1 \leq T \leq 10$)，表示一共有 T 组数据。每一组数据的第一行包含了三个用空格分开的正整数 R 、 C ($2 \leq R, C \leq 200$) 和 K ，表示地图是一个 $R \times C$ 的矩阵，而阿福需要集齐 K 种宝石才能够打开拘禁公主的结界。接下来的 R 行描述了地图的具体内容，每一行包含了 C 个字符。字符含义如题目描述中所述。保证有且仅有一个 S 和 E 。 $\$$ 的数量不超过 10 个。宝石的类型在数字 0 至 4 范围内，即不会超过 5 种宝石。

输出

对于每一组数据，输出阿福救出公主所花费的最少单位时间。若阿福无法救出公主，则输出“oop!”（只输出引号里面的内容，不输出引号）。每组数据的输出结果占一行。

样例输入

```
1
7 8 2
.....
..S..#0.
.##..1..
.0#.....
...1#...
...##E..
...1....
```

样例输出

```
11
```

试题编号：20230318-7-02
试题类型：编程题
标准答案：
试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：25

是否评分：已评分

评价描述：

考生答案：

```
#include<cstdio>
#include<cstring>
#include<queue>
#include<cmath>
using namespace std;
struct POINT
{
    int x,y,tot;
    bool get_jewel[7],enchantment;
} point,door;
int main()
{
    int k,F[4][2]={-1,0,1,0,0,-1,0,1};
    scanf("%d",&k);
    for(int kk=1;kk<=k;kk++)
    {

        queue<POINT> que;
        int r,c,needkind,door_coord[15][2]={},sdoor_coord=0;
        char maze[205][205]={};
        bool walked[205][205][35]={},finish=false;
        scanf("%d%d%d",&r,&c,&needkind);
        for(int i=0;i<r;i++)
        {
            scanf("%s",maze[i]);
            for(int j=0;j<c;j++)
            {
                if(maze[i][j]=='S')
                {
                    memset(point.get_jewel,false,sizeof(point.get_jewel));
                    point.x=i;point.y=j;point.tot=0;point.enchantment=true;
                }
                if(maze[i][j]=='$')
                {
                    door_coord[sdoor_coord][0]=i;
                    door_coord[sdoor_coord++][1]=j;
                }
            }
        }
    }
}
```

```

    }
}
que.push(point);
while(!que.empty())
{
    for(int i=0;i<4;i++)
    {
        point=que.front();
        point.tot++;point.x+=F[i][0];point.y+=F[i][1];
        if(point.x<0 || point.x>=r || point.y<0 || point.y>=c || maze[point.x]
[point.y]=='#') continue;
        if('0'<=maze[point.x][point.y] && maze[point.x][point.y]<='4')
            point.get_jewel[maze[point.x][point.y]-'0']=true;
        int sget_jewel=0,ss=0;
        for(int i=0;i<5;i++)
            if(point.get_jewel[i])
                sget_jewel+=pow(2,i),ss++;
        if(walked[point.x][point.y][sget_jewel]) continue;
        walked[point.x][point.y][sget_jewel]=true;
        if(ss>=needkind) point.enchantment=false;
        if(maze[point.x][point.y]=='$')
        {
            door=point;
            for(int i=0;i<sdoor_coord;i++)
                if(door_coord[i][0]!=point.x || door_coord[i][1]!=point.y)
                {
                    door.x=door_coord[i][0];door.y=door_coord[i][1];
                    que.push(door);
                }
        }
        if(maze[point.x][point.y]=='E' && !point.enchantment)
        {
            printf("%d\n",point.tot);
            finish=true;
        }
        if(finish) break;
        que.push(point);
    }
    if(finish) break;
    que.pop();
}
if(!finish) printf("oop!\n");
}
return 0;

```

}

3. 有多少种二叉树

输入 $n(1 < n < 13)$ ，求 n 个结点的二叉树有多少种形态

时间限制：1000

内存限制：65536

输入

整数 n

输出

答案

样例输入

3

样例输出

5

试题编号：20230318-7-03

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：25

是否评分：已评分

评价描述：

考生答案：

```
#include <iostream>
```

```
using namespace std;
```

```
long long cat(int n)
```

```
{
```

```
    if (n == 0)
```

```
        return 1;
```

```
    long long sum(0);
```

```
    for (int i = 0; i < n; i++)
```

```
        sum += cat(i) * cat(n-1-i);
```

```
    return sum;
```

```
}  
int main(){  
    int n;  
    cin>>n;  
    cout << cat(n) ;  
    return 0;  
}
```

4. Sequence

给定 m 个数字序列，每个序列包含 n 个非负整数。我们从每一个序列中选取一个数字组成一个新的序列，显然一共可以构造出 n^m 个新序列。接下来我们对每一个新的序列中的数字进行求和，一共会得到 n^m 个和，请找出最小的 n 个和

时间限制：3000

内存限制：65536

输入

输入的第一行是一个整数 T ，表示测试用例的数量，接下来是 T 个测试用例的输入 每个测试用例输入的第一行是两个正整数 m ($0 < m \leq 100$) 和 n ($0 < n \leq 2000$)，然后有 m 行，每行有 n 个数，数字之间用空格分开，表示这 m 个序列 序列中的数字不会大于 10000

输出

对每组测试用例，输出一行用空格隔开的数，表示最小的 n 个和

样例输入

```
1  
2 3  
1 2 3  
2 2 3
```

样例输出

```
3 3 4
```

试题编号：20230318-7-04

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：25

是否评分：已评分

评价描述：

考生答案：

```
#include <cstdio>
#include <iostream>
#include <cstring>
#include <queue>
#include <algorithm>
using namespace std;
const int maxn = 2005;
int t,m,n,a[maxn],b[maxn];
priority_queue<int> q;
int main()
{
    scanf("%d",&t);
    while(t--) {
        scanf("%d%d",&m,&n);
        for (int i = 0; i < n; ++i) scanf("%d",&a[i]);sort(a,a+n);
        m--;
        while(m--) {
            for (int i = 0; i < n; ++i) scanf("%d",&b[i]);sort(b,b+n);
            for (int i = 0; i < n; i++) q.push(a[i]+b[0]);
            for (int i = 1; i < n; ++i) {
                int j;
                for (j = 0; j < n; ++j)
                    if(b[i]+a[j]<q.top()) q.pop(),q.push(b[i]+a[j]);
                else{break;}
            }
            if(j == 0){break;}
        }
        for (int i = n-1; i >= 0; i--) a[i] = q.top(),q.pop();
    }
    for (int i = 0; i < n; i++) printf("%d%c",a[i],i == n-1?'\n':' ');
}
```