

A - Substrings

题解：详细看 ppt

代码：

```
#include<bits/stdc++.h>
using namespace std;

char p[105][105];
int t,n;

int main(){
    int mi=9999;int ma=-1;
    int f;
    int len1,len;
    int flag;
    scanf("%d",&t);
    while(t--){
        scanf("%d",&n);
        ma=9999;
        for(int i=0;i<n;i++){
            scanf("%s",p[i]);
            len1=strlen(p[i]);
            if(len1<mi){
                mi = len1;
                f = i;
            }
        }
        char str1[105],str2[105];
        len=strlen(p[f]);flag=1;
        ma=0;
        for(int i=0;i<len;i++){
            for(int j=i;j<len;j++){
                for(int k=i;k<=j;k++){
                    str1[k-i]=p[f][k];
                    str2[j-k]=p[f][k];
                }
                str1[j-i+1] = str2[j-i+1] = '\0';
                int l=strlen(str1);
                for(int k=0;k<n;k++){
                    if(!strstr(p[k],str1)&&!strstr(p[k],str2)){
                        flag=0;
                        break;
                    }
                }
            }
        }
    }
}
```

```

    }
}
if(l>ma&&flag){
    ma=l;
}
flag=1;
}
}
printf("%d\n",ma);
}
}

```

B - Calling Extraterrestrial Intelligence Again

题解：详细看 ppt

代码：

```

#include<bits/stdc++.h>
using namespace std;

int prime[2000],k;
bool visit[10005];
int num;

void getprime()
{
    memset(visit, false, sizeof(visit));
    num = 0;
    for (int i = 2; i <= 10000; ++i)
    {
        if ( !visit[i] ) prime[++num] = i;
        for (int j = 1; j <= num && i * prime[j] <= 10000 ; j++)
        {
            visit[ i * prime[j] ] = true;
            if (i % prime[j] == 0) break;
        }
    }
}

int main(){
    getprime();
}

```

```

int m,a,b;
int p,q;
while(~scanf("%d%d%d",&m,&a,&b)&&m&&a&&b){
    double as=1.0*a/b;
    int res;
    int ma=0;
    for(int i=num;i>0;i--){
        for(int j=num;j>=i;j--){
            res=prime[i]*prime[j];
            if(res<=m&&1.0*prime[i]/prime[j]>=as){
                if(res>ma){
                    ma=res;
                    p=prime[i];
                    q=prime[j];
                }
                break;
            }
        }
    }
    printf("%d %d\n",p,q);
}
}

```

C - Tempter of the Bone

题解：详细看 ppt

代码：

```

#include<bits/stdc++.h>
using namespace std;

int n,m,t;
char mp[10][10];
int starti,startj,endi,endj;
int cnt;
int flag=0;
int dir[4][2]= {{0,1},{0,-1},{1,0},{-1,0}};

void dfs(int px,int py,int time){
    if(flag==1) return;
    if(mp[px][py]=='X') return;

```

```

    if(time>t) return;
    if(px<1||px>n||py<1||py>m) return;
    if(px==endi&&py==endj&&time==t){
        flag=1;
        return;
    }
    for(int i=0;i<4;i++){
        mp[px][py]='X';
        dfs(px+dir[i][0],py+dir[i][1],time+1);
        mp[px][py]='.';
        if(flag==1) return;
    }
}

int main(){
    while(scanf("%d%d%d",&n,&m,&t)&&n&&m&&t){
        cnt=0;
        for(int i=1;i<=n;i++){
            for(int j=1;j<=m;j++){
                cin>>mp[i][j];
                if(mp[i][j]=='S'){
                    starti=i;
                    startj=j;
                }
                if(mp[i][j]=='D'){
                    endi=i;
                    endj=j;
                }
                if(mp[i][j]=='X') cnt++;
            }
        }
        if(abs(starti+startj+endi+endj-t)%2==1){
            printf("NO\n");
            continue;
        }
        flag=0;
        dfs(starti,startj,0);
        if(flag) printf("YES\n");
        else printf("NO\n");
    }
}

```

D - A strange lift

题解：这题是最简单最纯粹的 bfs 广搜，只要作一下标记去过的地方，不能再去，然后只要查找到了就跳出就行了。

代码：

```
#include<bits/stdc++.h>
using namespace std;

int n,a,b;
int k[255];
int ans[255];

int main(){
    while(cin>>n){
        if(n==0) break;
        cin>>a>>b;
        for(int i=1;i<=n;i++){
            cin>>k[i];
        }
        queue<int>q;
        q.push(a);
        memset(ans,0,sizeof(ans));
        ans[a]=1;
        int t;
        while(!q.empty()){
            t=q.front();
            q.pop();
            if(t==b) break;
            int tmp=t+k[t];
            if(tmp>=1&&tmp<=n&&ans[tmp]==0){
                q.push(tmp);
                ans[tmp]=ans[t]+1;
            }
            tmp=t-k[t];
            if(tmp>=1&&tmp<=n&&ans[tmp]==0){
                q.push(tmp);
                ans[tmp]=ans[t]+1;
            }
        }
        if(t!=b) ans[b]=0;
        cout<<ans[b]-1<<endl;
    }
}
```

E - Oil Deposits

这道题可以使用 dfs 也可以使用 bfs，但是 dfs 会相对比较简单，找到每一个 '@'，然后遍历它周围的所有 '@' 标记好好避免再次访问即可，最后统计一下一共找了几次就可。

代码：

```
#include<bits/stdc++.h>
using namespace std;

char grid[105][105];
int idx[105][105];
int m,n;

void dfs(int r,int c,int id){
    if(r<0||r>=m||c<0||c>=n) return;
    if(idx[r][c]>0||grid[r][c]!='@') return;
    idx[r][c]=id;
    for(int dr=-1;dr<=1;dr++){
        for(int dc=-1;dc<=1;dc++){
            if(dr!=0||dc!=0) dfs(r+dr,c+dc,id);
        }
    }
}

int main(){
    while(scanf("%d%d",&m,&n)==2&&m&&n){
        for(int i=0;i<m;i++){
            scanf("%s",grid[i]);
        }
        memset(idx,0,sizeof(idx));
        int cnt=0;
        for(int i=0;i<m;i++){
            for(int j=0;j<n;j++){
                if(idx[i][j]==0&&grid[i][j]=='@') dfs(i,j,++cnt);
            }
        }
        printf("%d\n",cnt);
    }
    return 0;
}
```

F - Prime Ring Problem

题解：这道题也是一个经典的 dfs，很容易分析出来，但是这道题目需要使用回溯。

代码：

```
#include<bits/stdc++.h> //万能头文件
using namespace std;

int a[25];
bool arr[25];
int n;

bool is_prime(int x){
    for(int i=2;i*i<=x;i++){
        if(x%i==0) return false;
    }
    return true;
}

void dfs(int k){
    if(k==n+1&&is_prime(a[n+1])){
        for(int i=1;i<=n;i++){
            printf("%d ",a[i]);
        }
        printf("%d\n",a[n]);
    }
    else {
        for(int i=2;i<=n;i++){
            if(arr[i]==false&&is_prime(i+a[k-1])){
                arr[i]=true;
                a[k]=i;
                dfs(k+1);
                arr[i]=false;
            }
        }
    }
}

int main(){
    int cnt=1;
    while(cin>>n){
        a[1]=1;
        printf("Case %d:\n",cnt++);
        dfs(2);
        cout<<endl;
    }
}
```

