## A 篮球联赛

```cpp
#include <stdio.h>
#include <iostream>
using namespace std;

int t;
char s[5][5];
int ans;

void check() {
  int score[5];

  for (int i = 0;i < 4;++i)
    score[i] = 0;
  for (int i = 0;i < 3;++i)
    for (int j = i+1;j < 4;++j)
      if (s[i][j] == 'L') score[j]++;
      else score[i]++;

  int tmp = 4;
  for (int i = 1;i < 4;++i)
    if (score[0] >= score[i]) tmp--;
  ans = min(tmp, ans);
}

void dfs(int x, int y) {
  int x1, y1;

  if (x == 3) {
    check();
    return ;
  }
  x1 = y == 4 ? x + 1 : x;
  y1 = y == 4 ? x + 2 : y + 1;
  if (s[x][y] == '?') {
    s[x][y] = 'W';
    dfs(x1, y1);
    s[x][y] = 'L';
    dfs(x1, y1);
    s[x][y] = '?';
  }else {
    dfs(x1, y1);
  }
}

int main() {
  scanf("%d%*c", &t);
  while (t-- > 0) {
    for (int i = 0;i < 4;++i) {
      for (int j = 0;j < 4;++j)
        scanf("%c", &s[i][j]);
      scanf("%*c");
    }
    ans = 4;
    dfs(0, 1);
    printf("%d\n", ans);
  }

```

```
  return 0;
}
```

## B 夺宝探险

```
#include <iostream>
#include <cstdlib>
#include <algorithm>
#include <cstring>

using namespace std;

int m, n, k;
int map[30][30];
int next_pos[4][2] = {{-1, 0}, {1, 0}, {0, -1}, {0, 1}};
bool vis[30][30];
bool treasure[110];
int treasure_sum;
int ans;

void dfs(int x, int y) {
    treasure[map[x][y]] = true;
    treasure_sum++;
    vis[x][y] = true;
    if (ans < treasure_sum)
        ans = treasure_sum;
    for (int i = 0; i < 4; i++) {
        int nx = x + next_pos[i][0];
        int ny = y + next_pos[i][1];
        if (1 <= nx && nx <= m && 1 <= ny && ny <= n) {
            if (!vis[nx][ny] && !treasure[map[nx][ny]]) {
                dfs(nx, ny);
            }
        }
    }
    treasure_sum--;
    vis[x][y] = false;
    treasure[map[x][y]] = false;
}

int main() {
    cin >> m >> n >> k;
    for (int i = 1; i <= m; i++)
        for (int j = 1; j <= n; j++)
            cin >> map[i][j];
    memset(vis, 0, sizeof(vis));
    memset(treasure, 0, sizeof(treasure));
    treasure_sum = 0;
    ans = 0;
    dfs(1, 1);
    cout << ans << endl;
}
```

## C 寻找边缘

```
#include <cstdio>
#include <iostream>
```

```cpp
#include <cstring>
using namespace std;

const int N = 512;
const int dx[4] = {0,1,0,-1};
const int dy[4] = {1,0,-1,0};
int T,n,m;
bool f[N][N];
char c[N][N];
struct unit{
    int i,j;
}q[N*N];
int tot,now,nowi,nowj,nexti,nextj;

int main()
{
    scanf("%d", &T);
    while (T --){
        scanf("%d%d",&n,&m);
        memset(f,0,sizeof(f));
        tot = 0;
        for(int i = 0;i < n; ++ i){
            scanf("%s",c[i]);
            for(int j = 0;j < m; ++ j){
                if (c[i][j] == '0' && (i == 0 || j == 0 || i == n - 1 || j == m - 1)){
                    q[tot].i = i;
                    q[tot].j = j;
                    f[i][j] = true;
                    tot ++;
                }
            }
        }
        now = 0;
        while (now < tot){
            nowi = q[now].i;
            nowj = q[now].j;
            for(int i = 0; i < 4; ++ i){
                nexti = nowi + dx[i];
                nextj = nowj + dy[i];
                if (nexti >= 0 && nexti < n && nextj >= 0 && nextj < m && c[nexti][nextj] == '0' && f[nexti][nextj] == false){
                    f[nexti][nextj] = true;
                    q[tot].i = nexti;
                    q[tot].j = nextj;
                    tot ++;
                }
            }
            now++;
        }
        for(int i = 0;i < n; ++ i){
            for(int j = 0;j < m; ++ j){
                if (f[i][j] == true)
                    printf("%c",'0');
                else
                    printf("%c",'X');
            }
            puts("");
        }
        puts("");
```

```
    }
}
```

## D 猴子摘桃

```cpp
#include <cstdio>
#include <cstdlib>
using namespace std;

int p[100000], b[100000], h, n, sump, sumb, l, maxp;

int main(int argc, const char * argv[]) {
    while (true) {
        scanf("%d", &h);
        if (h == -1) break;
        sump = sumb = l = maxp = 0;
        for (n = 0; true; ++n) {
            scanf("%d%d", p + n, b + n);
            if (p[n] == -1 && b[n] == -1) break;
            sump += p[n];
            sumb += b[n];
            while (l <= n && sumb > h) {
                sump -= p[l];
                sumb -= b[l];
                ++l;
            }
            if (sumb <= h && sump > maxp)
                maxp = sump;
        }
        printf("%d\n", maxp);
    }
    return 0;
}
```

## E 分形盒

```cpp
#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;

char map[731][731];

void dfs(int n,int x,int y)
{
    int size;
    if(n==1)
    {
        map[x][y]='X';
        return ;
    }
    size=pow(3.0,n-2);
    dfs(n-1,x,y);       //左上角
    dfs(n-1,x,y+2*size);       //右上角
    dfs(n-1,x+size,y+size);    //中间
    dfs(n-1,x+2*size,y);       //左下角
    dfs(n-1,x+2*size,y+2*size);   //右下角
```

```
}
int main(void)
{
    int i,j,n,size;
    while(scanf("%d",&n)!=EOF)
    {
        if(n==-1)
            break;
        size=pow(3.0,n-1);
        for(i=1;i<=size;i++)
        {
            for(j=1;j<=size;j++)
                map[i][j]=' ';
        }
        dfs(n,1,1);

        for(i=1;i<=size;i++)
        {
            for(j=1;j<=size;j++)
                printf("%c",map[i][j]);
            printf("\n");
        }
        printf("-\n");
    }
    return 0;
}
```

F42 点

```
#include <iostream>
#include <cstdlib>
#include <algorithm>

using namespace std;

int num[20];
int n;
bool found;

void dfs(int op1, int op2, int index) {
    if (index == n || found) {
        if (op1 == 42 || op2 == 42)
            found = true;
        return;
    }
    dfs(op1 + op2, num[index + 1], index + 1);
    dfs(op1 * op2, num[index + 1], index + 1);
    dfs(op1 - op2, num[index + 1], index + 1);
    if (op2 != 0 && op1 % op2 == 0)
        dfs(op1 / op2, num[index + 1], index + 1);
    dfs(op1, op2 + num[index + 1], index + 1);
    dfs(op1, op2 - num[index + 1], index + 1);
    dfs(op1, op2 * num[index + 1], index + 1);
    if (num[index + 1] != 0 && op2 % num[index + 1] == 0)
        dfs(op1, op2 / num[index + 1], index + 1);
}

int main() {
```

```
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> num[i];
    }
    found = false;
    if (n == 1) {
        found = (num[0] == 42);
    } else {
        sort(num, num + n);
        do {
            dfs(num[0], num[1], 1);
        } while (next_permutation(num, num + n) && !found);
    }
    if (found)
        cout << "YES" << endl;
    else
        cout << "NO" << endl;
}
```

## G 上机

```
#include <cstdio>
#include <iostream>

using namespace std;

const int N = 10012;
int a[N], b[N], c[N];
int f[N][2];
int n;
int main()
{
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);
    for (int i = 0; i < n; i++)
        scanf("%d", &b[i]);
    for (int i = 0; i < n; i++)
        scanf("%d", &c[i]);
    f[0][0] = a[0];
    f[0][1] = b[0];
    for (int i = 1; i < n; i++) {
        f[i][0] = max(f[i-1][0]+b[i], f[i-1][1]+a[i]);
        f[i][1] = max(f[i-1][0]+c[i], f[i-1][1]+b[i]);
    }
    cout << f[n-1][0] << endl;
    return 0;
}
```

## H 迷宫入口

```
#include <stdio.h>
#include <iostream>
using namespace std;

#define MAXM (10+5)
#define MAXS (40+5)
```

```
7
8   int t, s, m, n;
9   int card[MAXM];
10  int row[MAXS];
11
12  bool dfs(int k) {
13    int now = 0, extend;
14    //printf("%d\n", k);
15    if (k == n) return true;
16    //select row with smallest size
17    for (int i = 0;i < s;++i)
18      if (row[i] < row[now]) now = i;
19
20    for (int i = now;i < s;++i)
21      if (row[i] <= row[now]) extend = i - now + 1;
22      else break;
23    extend = min(m, min(extend, s - row[now]));
24
25    for (int i = 1;i <= extend;++i){
26      if (card[i] > 0) {
27        for (int j = now;j < now + i;++j)
28          row[j] += i;
29        --card[i];
30
31        if (dfs(k + 1)) return true;
32
33        ++card[i];
34        for (int j = now;j < now + i;++j)
35          row[j] -= i;
36      }
37    }
38
39    return false;
40
41  }
42
43  int main() {
44    int tot, k;
45
46    scanf("%d", &t);
47    while (t-- > 0) {
48      scanf("%d %d", &s, &n);
49      for (int i = 0;i <= 10; ++i) card[i] = 0;
50      for (int i = 0;i <= MAXS;++i) row[i] = 0;
51      tot = 0;
52      m = 0;
53      for (int i = 0;i < n;++i) {
54        scanf("%d", &k);
55        card[k]++;
56        tot += k*k;
57        if (k > m) m = k;
58      }
59      if (tot == s * s and dfs(0)) printf("YES\n");
60      else printf("NO\n");
61    }
62
63    return 0;
64  }
```

## I 变换的迷宫

```cpp
#include <cstdio>
#include <cstring>
#include <queue>

using namespace std;

struct Node {
  int x, y, t;
  Node() { }
  Node(int x, int y, int t) : x(x), y(y), t(t) { }
  bool isIn(int r, int c) {
    return x < r && x >= 0 && y < c && y >= 0;
  }
  bool operator == (const Node &p) const {
    return x == p.x && y == p.y;
  }
};

char mat[0x100][0x100];
bool vis[0x100][0x100][0x10];

int bfs(Node &st, Node &ed, int r, int c, int k) {
  queue<Node> que;
  memset(vis, 0, sizeof(vis));
  int dx[] = {0, 1, 0, -1};
  int dy[] = {1, 0, -1, 0};

  que.push(st);
  vis[st.x][st.y][st.t] = true;
  while (!que.empty()) {
    Node cur = que.front();
    que.pop();

    if (cur == ed) return cur.t;
    for (int i = 0; i < 4; ++i) {
      Node nxt(cur.x + dx[i], cur.y + dy[i], cur.t + 1);
      if (nxt.isIn(r, c) && !vis[nxt.x][nxt.y][nxt.t % k]) {
        if (nxt.t % k == 0 || mat[nxt.x][nxt.y] != '#') {
          que.push(nxt);
          vis[nxt.x][nxt.y][nxt.t % k] = true;
        }
      }
    }
  }
  return -1;
}

int main() {


  int t, r, c, k;
  Node st, ed;

  scanf("%d", &t);
  while (t--) {
    scanf("%d%d%d", &r, &c, &k);
    for (int i = 0; i < r; ++i) {
```

```
      scanf("%s", mat[i]);
      for (int j = 0; j < c; ++j) {
        if (mat[i][j] == 'S') {
          st = Node(i, j, 0);
        }
        else if (mat[i][j] == 'E') {
          ed = Node(i, j, -1);
        }
      }
    }

    int ret = bfs(st, ed, r, c, k);
    if (ret == -1) printf("Oop!\n");
    else printf("%d\n", ret);
  }
  return 0;
}

```

## J 游览规划

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int f[1001][2473];
char type[10];
int m, n, s, Hi, Ci, Ti, ans;

int main() {
    scanf("%d%d%d", &m, &n, &s);
    memset(f, -1, sizeof(f));
    m *= 24;
    f[s][0] = 0;
    for (int i = s; i >= 0; --i)
        for (int j = 0; j <= m; ++j)
            if (f[i][j] == 0) {
                if (i >= 100)
                    f[i - 100][j + 24] = 0;
                if (i >= 150)
                    f[i - 150][j + 48] = 0;
                if (i >= 200)
                    f[i - 200][j + 72] = 0;
            }
    while (n--) {
        scanf("%d%d%d%s", &Hi, &Ci, &Ti, type);
        if (strcmp(type, "limited") == 0) {
            for (int i = Ci; i <= s; ++i)
                for (int j = Ti; j <= m; ++j)
                    if (f[i][j] >= 0 && f[i][j] + Hi > f[i - Ci][j - Ti])
                        f[i - Ci][j - Ti] = f[i][j] + Hi;
        } else {
            for (int i = s; i >= Ci; --i)
                for (int j = m; j >= Ti; --j)
                    if (f[i][j] >= 0 && f[i][j] + Hi > f[i - Ci][j - Ti])
                        f[i - Ci][j - Ti] = f[i][j] + Hi;

        }
```

```
    }
    for (int i = 0; i <= s; ++i)
        for (int j = 0; j <= m; ++j)
            if (f[i][j] > ans)
                ans = f[i][j];
    printf("%d\n", ans);
}
```

## K 张三丰的传人

```
#include <cstdio>
#include <cstring>
#include <cmath>

const int maxn = 100;
const double eps = 1e-6;
int size, a[maxn], now, n;
bool ok;

void dfs( int sum, int x, int idx, double remain, int pro ){
  if( sum>57 ){ // 当数较大的时候，通过区分奇数偶数，进行规模缩小
    if( sum&1 ){
      dfs( (sum-9)/2, 1, idx+2, 1.0, pro*2 ); // 如果是奇数
      a[idx]=3*pro, a[idx+1]=6*pro;
      return;
    }
    else{
      dfs( (sum-2)/2, 1, idx+1, 1.0, pro*2 ); // 如果是偶数
      a[idx]=2*pro;
      return;
    }
  }
  if( sum==0 ){
    if( fabs(remain)<eps )
      size = idx, ok = true;
    return;
  }
  if( sum<x || remain<0 ) return;
  // 数字较小的时候，通过尝试的方法得到解
  dfs( sum-x, x, idx+1, remain-1.0/x, pro ); // 尝试第一个数为 x
  a[idx]=x*pro;
  if( ok ) return;
  dfs( sum, x+1, idx, remain, pro ); //尝试第一个数为 x+1
}

int main(){
    //freopen("equation.in","r",stdin);
    //freopen("out.txt", "w", stdout);
  int T;
  scanf("%d", &T);
  while( T-- ){
    scanf("%d", &n);
    ok = false;
    dfs( n, 1, 0, 1.0, 1 );
    if( ok ){
      printf("%d", size);
      for( int i=0; i<size; ++i ) printf(" %d", a[i] );
      printf("\n");
```

```
49        continue;
50      }
51      printf("-1\n");
52    }
53    return 0;
54  }
```