

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

题号为杭州电子科技大学在线评判系统对应题号：<http://acm.hdu.edu.cn/>

2008 年第一题(1877):

Problem Description

输入两个不超过整型定义的非负10进制整数A和B($\leq 2^{31}-1$)，输出A+B的m($1 < m < 10$)进制数。

Input

输入格式：测试输入包含若干测试用例。每个测试用例占一行，给出m和A，B的值。
当m为0时输入结束。

Output

输出格式：每个测试用例的输出占一行，输出A+B的m进制数。

Sample Input

```
8 1300 48
2 1 7
0
```

Sample Output

```
2504
1000
```

源代码:

```
#include <stdio.h>
#include <string.h>
int main()
{
    int a,b,m,c,k,i;
    int s[100];
    while(scanf("%d",&m) && m)
    {
        memset(s,0,sizeof(s));
        k=1;
        scanf("%d%d",&a,&b);
        c=a+b;
        if(c==0)
        {
            printf("0\n");
            continue;
        }
        while(c)
```

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

```
{
    s[k++]=c%m;
    c=c/m;
}
for(i=k-1;i>0;i--)
    printf("%d",s[i]);
    printf("\n");
}
return 0;
}
```

2008 年第二题(1878):

Problem Description

欧拉回路是指不令笔离开纸面，可画过图中每条边仅一次，且可以回到起点的一条回路。现给定一个图，问是否存在欧拉回路？

Input

测试输入包含若干测试用例。每个测试用例的第1行给出两个正整数，分别是节点数 N ($1 < N < 1000$)和边数 M ；随后的 M 行对应 M 条边，每行给出一对正整数，分别是该条边直接连通的两个节点的编号（节点从1到 N 编号）。当 N 为0时输入结束。

Output

每个测试用例的输出占一行，若欧拉回路存在则输出1，否则输出0。

Sample Input

```
3 3
1 2
1 3
2 3
3 2
1 2
2 3
0
```

Sample Output

```
1
0
```

源代码:

```
#include <stdio.h>
#include <string.h>
#define N 1001
int degree[N];
int pre[N];
int find(int r)
```

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

```
{
    while(r!=pre[r])
        r=pre[r];
    return r;
}
int main()
{

    int n,m,i,a,b,tag,total;
    while(scanf("%d",&n) && n)
    {
        memset(degree,0,sizeof(degree));
        tag=1;
        total=n-1;
        scanf("%d",&m);
        for(i=1;i<=n;i++)
            pre[i]=i;
        for(i=1;i<=m;i++)
        {
            scanf("%d%d",&a,&b);
            degree[a]++;
            degree[b]++;
            if(find(a)!=find(b))
            {
                pre[find(a)]=find(b);
                total--;
            }
        }
        for(i=1;i<=n;i++)
            if(degree[i]%2!=0)
            {
                tag=0;
                break;
            }

        if(tag && total==0)
            printf("1\n");
        else
            printf("0\n");
    }
    return 0;
}
```

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

2008 年第三题(1879):

省政府“畅通工程”的目标是使全省任何两个村庄间都可以实现公路交通（但不一定有直接的公路相连，只要能间接通过公路可达即可）。现得到城镇道路统计表，表中列出了任意两城镇间修建道路的费用，以及该道路是否已经修通的状态。现请你编写程序，计算出全省畅通需要的最低成本。

Input

测试输入包含若干测试用例。每个测试用例的第1行给出村庄数目N ($1 < N < 100$)；随后的 $N(N-1)/2$ 行对应村庄间道路的成本及修建状态，每行给4个正整数，分别是两个村庄的编号（从1编号到N），此两村庄间道路的成本，以及修建状态：1表示已建，0表示未建。

当N为0时输入结束。

Output

每个测试用例的输出占一行，输出全省畅通需要的最低成本。

Sample Input

```
3
1 2 1 0
1 3 2 0
2 3 4 0
3
1 2 1 0
1 3 2 0
2 3 4 1
3
1 2 1 0
1 3 2 1
2 3 4 1
0
```

Sample Output

```
3
1
0
```

源代码:

```
#include <stdio.h>
#include <string.h>
#define inf 100001
#define N 101
int map[N][N], dist[N], visited[N];
int total, MIN;
int i, x;
int n, j, m;
int prim()
{
    for(i=0; i<n; i++)
    {
        dist[i]=map[0][i];
        visited[i]=0;
```

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

```
}
    visited[0]=1;
    total=0;
    dist[0]=0;
    for(i=1;i<n;i++)
    {
        MIN=inf;
        for(j=1;j<n;j++)
        {
            if(!visited[j] && dist[j] < MIN)
            {
                MIN=dist[j];
                x=j;
            }
        }
        if(MIN==inf)
            break;
        visited[x]=1;
        total+=MIN;
        for(j=0;j<n;j++)
        {
            if( !visited[j] && map[x][j] < dist[j])
                dist[j]=map[x][j];
        }
    }
    return total;
}

int main()
{
    int a,b,c,k;
    while(scanf("%d",&n) && n)
    {
        m=n*(n-1)/2;
        for(i=0;i<m;i++)
        {
            scanf("%d %d %d %d",&a,&b,&c,&k);
            if(k)
                map[a-1][b-1]=map[b-1][a-1]=0;
            else
                map[a-1][b-1]=map[b-1][a-1]=c;
        }
        printf("%d\n",prim());
    }
}
```

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

```
}  
return 0;  
}
```

2008 年第四题(1880):

哈利波特在魔法学校的必修课之一就是学习魔咒。据说魔法世界有100000种不同的魔咒，哈利很难全部记住，但是为了对抗强敌，他必须在危急时刻能够调用任何一个需要的魔咒，所以他需要你的帮助。

给你一部魔咒词典。当哈利听到一个魔咒时，你的程序必须告诉他那个魔咒的功能；当哈利需要某个功能但不知道该用什么魔咒时，你的程序要替他找到相应的魔咒。如果他要的魔咒不在词典中，就输出“what?”

Input

首先列出词典中不超过100000条不同的魔咒词条，每条格式为：

[魔咒] 对应功能

其中“魔咒”和“对应功能”分别为长度不超过20和80的字符串，字符串中保证不包含字符“[”和“]”，且“]”和后面的字符串之间有且仅有一个空格。词典最后一行以“@END@”结束，这一行不属于词典中的词条。词典之后的一行包含正整数N（≤1000），随后是N个测试用例。每个测试用例占一行，或者给出“[魔咒]”，或者给出“对应功能”。

Output

每个测试用例的输出占一行，输出魔咒对应的功能，或者功能对应的魔咒。如果魔咒不在词典中，就输出“what?”

Sample Input

```
[expelliarmus] the disarming charm  
[rictusempra] send a jet of silver light to hit the enemy  
[tarantallegra] control the movement of one's legs  
[serpensortia] shoot a snake out of the end of one's wand  
[lumos] light the wand  
[obliviate] the memory charm  
[expecto patronum] send a Patronus to the dementors  
[accio] the summoning charm  
@END@  
4  
[lumos]  
the summoning charm  
[arha]  
take me to the sky
```

Sample Output

```
light the wand  
accio  
what?  
what?
```

源代码:

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

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

```
using namespace std;
```

```
typedef struct
```

```
{  
    char a[25];  
    char b[85];  
}Node;
```

```
Node c[100005];
```

```
int main()
```

```
{  
    int i,len,k,n,len1,j,tag;  
    char s[105];  
    k=0;  
    while(gets(s) && strcmp(s,"@END@") !=0)  
    {
```

```
        len=strlen(s);  
        for(i=0;i<len;i++)  
            if(s[i]=='  
                break;  
        strncpy(c[k].a , s , i+1);  
        strncpy(c[k].b , s+i+2 , len-i);  
        k++;
```

```
    }  
    scanf("%d",&n);  
    getchar();
```

```
    while(n--)
```

```
    {  
        tag=0;  
        gets(s);  
        if(s[0]!='  
        {  
            for(i=0;i<k;i++)  
                if(strcmp(s,c[i].a)==0)  
                {  
                    printf("%s\n",c[i].b);  
                    tag=1;  
                    break;  
                }  
        }  
        else
```

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

```
{
    for(i=0;i<k;i++)
        if(strcmp(s,c[i].b)==0)
        {
            len1=strlen(c[i].a);
            for(j=1;j<len1-1;j++)
            {
                printf("%c",c[i].a[j]);
                tag=1;
            }
            printf("\n");
        }
    }
    if(tag==0)
        printf("what?\n");
}

return 0;
}
```

2008 年第五题(1881):

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbiji.com，为计算机考研学子打造的交流平台！

Problem Description

每年毕业的季节都会有大量毕业生发起狂欢，好朋友们相约吃散伙饭，网络上称为“bg”。参加不同团体的bg会有不同的感觉，我们可以用一个非负整数为每个bg定义一个“快乐度”。现给定一个bg列表，上面列出每个bg的快乐度、持续长度、bg发起人的离校时间，请你安排一系列bg的时间使得自己可以获得最大的快乐度。

例如有4场bg：

第1场快乐度为5，持续1小时，发起人必须在1小时后离开；

第2场快乐度为10，持续2小时，发起人必须在3小时后离开；

第3场快乐度为6，持续1小时，发起人必须在2小时后离开；

第4场快乐度为3，持续1小时，发起人必须在1小时后离开。

则获得最大快乐度的安排应该是：先开始第3场，获得快乐度6，在第1小时结束，发起人来得及时离开；再开始第2场，获得快乐度10，在第3小时结束，发起人正好来得及离开。此时已经无法再安排其他的bg，因为发起人都已经离开了学校。因此获得的最大快乐度为16。

注意bg必须在发起人离开前结束，你不可以中途离开一场bg，也不可以中途加入一场bg。

又因为你的人缘太好，可能有多达30个团体bg你，所以你需要写个程序来解决这个时间安排的问题。

Input

测试输入包含若干测试用例。每个测试用例的第1行包含一个整数N($N \leq 30$)，随后有N行，每行给出一场bg的信息：

h t t

其中 h 是快乐度，t是持续时间（小时），t是发起人离校时间。数据保证t不大于t，因为若发起人必须在t小时后离开，bg必须在主人离开前结束。

当N为负数时输入结束。

Output

每个测试用例的输出占一行，输出最大快乐度。

Sample Input

```
3
6 3 3
3 2 2
4 1 3
4
5 1 1
10 2 3
6 1 2
3 1 1
-1
```

Sample Output

```
7
16
```

源代码：

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

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

```
typedef struct
{
    int h,l,t;
}Node;

Node bg[32];
int dp[32],n,max1;

bool cmp( Node a , Node b )    //按照截止时间由小到大排序
{
    return a.t < b.t;
}

void Dfs( int t , int h , int time ) //t 为检查到第几个 bg 信息，h 为到当前为止得到的欢乐度，
time 为到当前为止的结束时间
{
    if( t >= n )    //全部检查完后看是否会替换最大值 max1
    {
        if( h > max1 )    max1 = h;
        return;
    }
    Dfs( t+1 , h , time );    //不考虑当前这场 bg 信息
    if( time + bg[t].l <= bg[t].t ) //考虑当前这场 bg 信息
        Dfs( t+1 , h+bg[t].h , time+bg[t].l );
}

int main()
{
    int i;
    while( scanf( "%d" , &n ) && n >= 0 )
    {
        //input
        for( i = 0 ; i < n ; i++ )
            scanf( "%d%d%d" , &bg[i].h , &bg[i].l , &bg[i].t );

        //sort
        sort( bg , bg+n , cmp );

        max1 = 0;
        Dfs( 0 , 0 , 0 );
        printf( "%d\n" , max1 );
    }
    return 1;
}
```

此文档由天勤论坛整理，转载请注明出处。天勤论坛：
www.csbjji.com，为计算机考研学子打造的交流平台！

}

注明：此题仍然为回溯法中的修枝剪叶算法！