variable-sized object may not be initialized

C99变长数组不能直接用a[n] = {0};这样的方式初始化

要用memset；

注意：

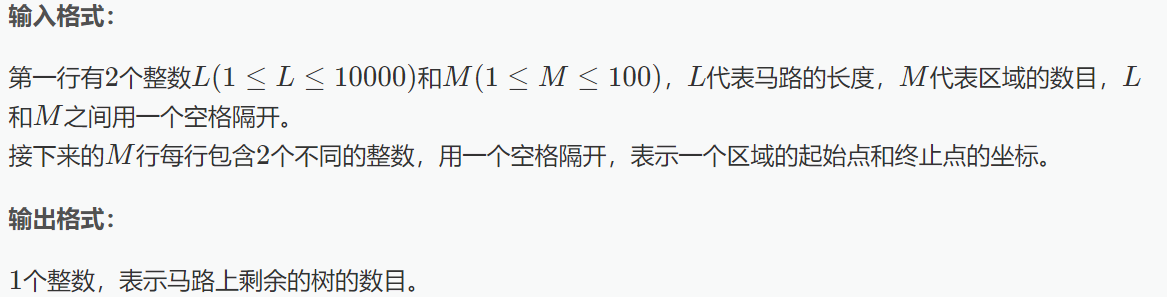
memset(count, 0, sizeof(count));//if not sizeof(count) then it would make mistakes!

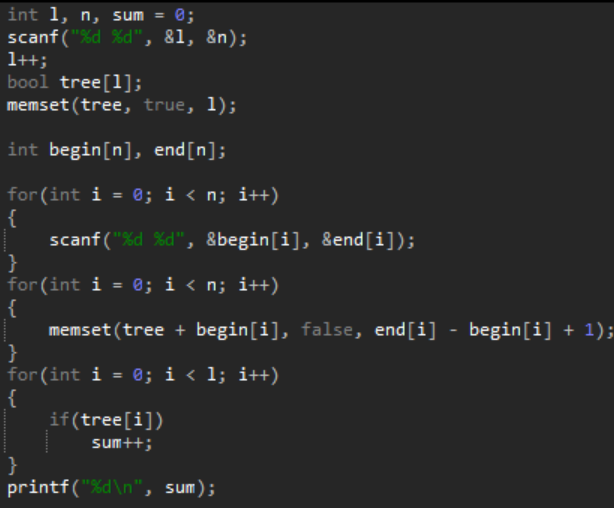
Memset的在数组中的用处不止在初始化，在数组元素置值，主要是置零，或者char、bool这样一个字节的变量的批量操作。

例P1047校门外的树

某校大门外长度为L的马路上有一排树，每两棵相邻的树之间的间隔都是1米。我们可以把马路看成一个数轴，马路的一端在数轴0的位置，另一端在L的位置；数轴上的每个整数点，即0,1,2,…,L都种有一棵树。

由于马路上有一些区域要用来建地铁。这些区域用它们在数轴上的起始点和终止点表示。已知任一区域的起始点和终止点的坐标都是整数，区域之间可能有重合的部分。现在要把这些区域中的树（包括区域端点处的两棵树）移走。你的任务是计算将这些树都移走后，马路上还有多少棵树。





※在像P1047这样多次覆盖但只算一次不复改变的可以用bool表示

※重复情况的考虑

P2141 珠心算测验

2014NOIP普及T1

/\*

某学校的珠心算老师采用一种快速考察珠心算加法能力的测验方法。他随机生成一个正整数集合，集合中的数各不相同，然后要求学生回答：其中有多少个数，恰好等于集合中另外两个（不同的）数之和？

最近老师出了一些测验题，请你帮忙求出答案。

共两行，第一行包含一个整数nn，表示测试题中给出的正整数个数。

第二行有nn个正整数，每两个正整数之间用一个空格隔开，表示测试题中给出的正整数。

\*/

int main(void)

{

int n, count = 0;

scanf("%d", &n);

int a[n];

bool re[n];

int add[n][n];

memset(add, 0, sizeof(add));

memset(re, false, sizeof(re));

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

{

scanf("%d", &a[i]);

}

for(int i = 0; i < n - 1; i++)

{

for(int j = i + 1; j < n; j++)

{

add[i][j] = a[i] + a[j];

}

}

for(int i = 0; i < n - 1; i++)

{

for(int j = i + 1; j < n; j++)

{

for(int k = 0; k < n; k++)

{

if(a[k] == add[i][j] && !re[k])

{

count++;

re[k] = true;

}

}

}

}

printf("%d", count);

return 0;

}