题意：议会由N个代表组成。代表被分到不同的组中，任意两个组的人数不能相等，任意一个代表只能在一个组中。每天每个组只能派一名代表去开会，每天参加会议的代表不能重复，只有这样议会才能正常工作。求如何分组使得议会正常工作的时间最长。\

我们首先先撇开“每组的人数不同”这一约束条件，我们发现如10可以拆成5+5，它的乘积最大，而5又可拆成2+3;2\*3>5;故我们发现每次的每一个数都拆成两个相等或相差1的两个数乘积最大。而最大的当然是一直拆到出现2；我们在回来考虑“每组的人数不同”这一约束条件；每组的人数不能相等，但我们又要使数拆成最小的部分（最好拆到出现2），比如说2.3.4~~，那我们大可从2加起，2+3+4+~~~一直加到接近所要求的数；如26我们可以用个循环得到2+3+4+5+6，然后让它停下，此时还剩余6（26-2-3-4-5-6）没用完；那我们就从加数序列的最大那个数开始倒序一次加一，这样过后加数序列就变成了3+4+5+6+7了，但还有1没用啊，当然加给最大的那个7（这个过程用一个取模运算就行了）；所以最后答案是3+4+5+6+8.

算法：数论。  
假设 N = A1+A2+...+An，那么议会正常工作的时间为A1\*A2\*...\*An，所求即为 A1\*A2\*...\*An的最大值。  
对任意一个整数a，a=b+c(b>1,c>1)，那么b\*c>=a，即任意一个数拆为两个数(都大于1)后其乘积大于该数。  
因此，我们的目标是求得 N=2+3+4+...+(n-1)+x  
因为拆分后的数不能重复，即最后剩余的x要拆为x个1，从后往前分别加到已拆的数中（如果从前往后会出现重复数值）。 \

例如：   
17=2+3+4+5+3   
2 3 4 5  
   1 1 1  
-----------

2 4 5 6

其实这样的话从小到大加 3 4 5 5，乘积要比2456大，但是题目要求不能有重复的

#include<stdio.h>

#include<iostream>

#include<cmath>

#include<cstring>

#include<set>

#include<queue>

#include<algorithm>

#include<vector>

#include<sstream>

using namespace std;

int main()

{

//freopen("input.txt","r",stdin);

int n;

int a[150];

while(~scanf("%d",&n))

{

memset(a,0,sizeof(a));

int k=1;

a[0]=2;

int sum=2;

while(sum+a[k-1]+1<=n)

{

a[k]=a[k-1]+1;

sum+=a[k];

k++;

}

int tmp=n-sum;

while(tmp>0)

for(int i=k-1;i>=0 && tmp>0;i--)

{

a[i]++;

tmp--;

}

printf("%d",a[0]);

for(int i=1;i<k;i++)

printf(" %d",a[i]);

printf("\n");

}

return 0;

}