**Problem A**

**Problem Description**

People in Silverland use square coins. Not only they have square shapes but also their values are square numbers. Coins with values of all square numbers up to 289 (=17^2), i.e., 1-credit coins, 4-credit coins, 9-credit coins, ..., and 289-credit coins, are available in Silverland.   
There are four combinations of coins to pay ten credits:   
  
ten 1-credit coins,  
one 4-credit coin and six 1-credit coins,  
two 4-credit coins and two 1-credit coins, and  
one 9-credit coin and one 1-credit coin.   
  
Your mission is to count the number of ways to pay a given amount using coins of Silverland.

**Input**

The input consists of lines each containing an integer meaning an amount to be paid, followed by a line containing a zero. You may assume that all the amounts are positive and less than 300.

**Output**

For each of the given amount, one line containing a single integer representing the number of combinations of coins should be output. No other characters should appear in the output.

**Sample Input**

2

10

30

0

**Sample Output**

1

4

27

**代码：**

#include<stdio.h>

int main(){

int n;

int c1[301];

int c2[301];

int i,j,k;

for(i=0;i<=300;i++){

c1[i]=1;

c2[i]=0;

}

for(i=2;i<=17;i++){

for(j=0;j<=300;j++){

for(k=0;k+j<=300;k=k+i\*i){

c2[j+k]=c2[j+k]+c1[j];

}

}

for(j=0;j<=300;j++){

c1[j]=c2[j];

c2[j]=0;

}

}

scanf("%d",&n);

while(n!=0){

printf("%d\n",c1[n]);

scanf("%d",&n);

}

return 0;

}

**Problem B**

**Problem Description**

We all know that Bin-Laden is a notorious terrorist, and he has disappeared for a long time. But recently, it is reported that he hides in Hang Zhou of China!   
“Oh, God! How terrible! ”



Don’t be so afraid, guys. Although he hides in a cave of Hang Zhou, he dares not to go out. Laden is so bored recent years that he fling himself into some math problems, and he said that if anyone can solve his problem, he will give himself up!   
Ha-ha! Obviously, Laden is too proud of his intelligence! But, what is his problem?  
“Given some Chinese Coins (硬币) (three kinds-- 1, 2, 5), and their number is num\_1, num\_2 and num\_5 respectively, please output the minimum value that you cannot pay with given coins.”  
You, super ACMer, should solve the problem easily, and don’t forget to take $25000000 from Bush!

**Input**

Input contains multiple test cases. Each test case contains 3 positive integers num\_1, num\_2 and num\_5 (0<=num\_i<=1000). A test case containing 0 0 0 terminates the input and this test case is not to be processed.

**Output**

Output the minimum positive value that one cannot pay with given coins, one line for one case.

**Sample Input**

1 1 3

0 0 0

**Sample Output**

4

**代码：**

#include<stdio.h>

int main(){

int a,b,c;

scanf("%d",&a);

scanf("%d",&b);

scanf("%d",&c);

while(1){

if(a==0&&b==0&&c==0){

break;

}

if(a==0)printf("1\n");

else{

if(a+2\*b<4)printf("%d\n",a+2\*b+1);

else{

printf("%d\n",a+2\*b+5\*c+1);

}

}

scanf("%d",&a);

scanf("%d",&b);

scanf("%d",&c);

}

return 0;

}

**Problem C**

**Problem Description**

"Well, it seems the first problem is too easy. I will let you know how foolish you are later." feng5166 says.  
  
"The second problem is, given an positive integer N, we define an equation like this:  
  N=a[1]+a[2]+a[3]+...+a[m];  
  a[i]>0,1<=m<=N;  
My question is how many different equations you can find for a given N.  
For example, assume N is 4, we can find:  
  4 = 4;  
  4 = 3 + 1;  
  4 = 2 + 2;  
  4 = 2 + 1 + 1;  
  4 = 1 + 1 + 1 + 1;  
so the result is 5 when N is 4. Note that "4 = 3 + 1" and "4 = 1 + 3" is the same in this problem. Now, you do it!"

**Input**

The input contains several test cases. Each test case contains a positive integer N(1<=N<=120) which is mentioned above. The input is terminated by the end of file.

**Output**

For each test case, you have to output a line contains an integer P which indicate the different equations you have found.

**Sample Input**

4

10

20

**Sample Output**

5

42

627

**代码：**

#include<stdio.h>

int main(){

int n;

int c1[121];

int c2[121];

int i,j,k;

while(scanf("%d",&n)!=EOF){

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

c1[i]=1;

c2[i]=0;

}

for(i=2;i<=n;i++){

for(j=0;j<=n;j++){

for(k=0;k+j<=n;k=k+i){

c2[j+k]=c2[j+k]+c1[j];

}

}

for(j=0;j<=n;j++){

c1[j]=c2[j];

c2[j]=0;

}

}

printf("%d\n",c1[n]);

}

return 0;

}

**Problem D**

**Problem Description**

Nowadays, we all know that Computer College is the biggest department in HDU. But, maybe you don't know that Computer College had ever been split into Computer College and Software College in 2002.  
The splitting is absolutely a big event in HDU! At the same time, it is a trouble thing too. All facilities must go halves. First, all facilities are assessed, and two facilities are thought to be same if they have the same value. It is assumed that there is N (0<N<1000) kinds of facilities (different value, different kinds).

**Input**

Input contains multiple test cases. Each test case starts with a number N (0 < N <= 50 -- the total number of different facilities). The next N lines contain an integer V (0<V<=50 --value of facility) and an integer M (0<M<=100 --corresponding number of the facilities) each. You can assume that all V are different.  
A test case starting with a negative integer terminates input and this test case is not to be processed.

**Output**

For each case, print one line containing two integers A and B which denote the value of Computer College and Software College will get respectively. A and B should be as equal as possible. At the same time, you should guarantee that A is not less than B.

**Sample Input**

2

10 1

20 1

3

10 1

20 2

30 1

-1

**Sample Output**

20 10

40 40

**代码：**

#include<stdio.h>

#include<string.h>

int a[250010],s[250010];

int v[55],m[55];

int main()

{

int n,i,j,k,sum,ans;

while(scanf("%d",&n)&&n>0)

{

sum=0;

memset(s,0,sizeof(s));

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

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

{

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

sum+=v[i]\*m[i];

}

for(i=0;i<=v[1]\*m[1];i+=v[1])

{

s[i]=1;

}

for(i=2;i<=n;i++)

{

for(j=0;j<=sum;j++)

{

for(k=0;k+j<=sum&&k<=v[i]\*m[i];k+=v[i])

{

a[k+j]+=s[j];

}

}

for(k=0;k<=sum;k++)

{

s[k]=a[k];

a[k]=0;

}

}

for(i=sum/2;i>=0;i--)

{

if(s[i])

{

printf("%d %d\n",sum-i,i);

break;

}

}

}

return 0;

}

**Problem E**

**Problem Description**

在一个国家仅有1分，2分，3分硬币，将钱N兑换成硬币有很多种兑法。请你编程序计算出共有多少种兑法。

**Input**

每行只有一个正整数N，N小于32768。

**Output**

对应每个输入，输出兑换方法数。

**Sample Input**

2934

12553

**Sample Output**

718831

13137761

**代码:**

#include<stdio.h>

int main(){

int n;

int c1[32769];

int c2[32769];

int i,j,k;

for(i=0;i<=32768;i++){

c1[i]=1;

c2[i]=0;

}

for(i=2;i<=3;i++){

for(j=0;j<=32768;j++){

for(k=0;k+j<=32768;k=k+i){

c2[j+k]=c2[j+k]+c1[j];

}

}

for(j=0;j<=32768;j++){

c1[j]=c2[j];

c2[j]=0;

}

}

while(scanf("%d",&n)!=EOF){

printf("%d\n",c1[n]);

}

return 0;

}