# 2082找单词

**Problem Description**

假设有x1个字母A， x2个字母B,..... x26个字母Z，同时假设字母A的价值为1，字母B的价值为2,..... 字母Z的价值为26。那么，对于给定的字母，可以找到多少价值<=50的单词呢？单词的价值就是组成一个单词的所有字母的价值之和，比如，单词ACM的价值是1+3+14=18，单词HDU的价值是8+4+21=33。(组成的单词与排列顺序无关，比如ACM与CMA认为是同一个单词）。

**Input**

输入首先是一个整数N，代表测试实例的个数。  
然后包括N行数据，每行包括26个<=20的整数x1,x2,.....x26.

**Output**

对于每个测试实例，请输出能找到的总价值<=50的单词数,每个实例的输出占一行。

**Sample Input**

2

1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

9 2 6 2 10 2 2 5 6 1 0 2 7 0 2 2 7 5 10 6 10 2 10 6 1 9

**Sample Output**

7

379297

//#include<stdio.h>

//#include<string.h>

//int main()

//{

// int i,j,k,n,sum,num[27],s[27][51];

// scanf("%d",&n);

// while(n--)

// {

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

// for(i=1;i<=26;i++)

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

// for(i=0;i<=26;i++)

// s[i][0]=1;

// for(i=1;i<=26;i++)

// for(j=1;j<=50;j++)

// for(k=0;k<=num[i]&&j-k\*i>=0;k++)

// s[i][j]+=s[i-1][j-k\*i];

// for(i=1,sum=0;i<=50;i++)

// sum+=s[26][i];

// printf("%d\n",sum);

// }

// return 0;

//}

//#include<stdio.h>

//#include<string.h>

//#include<math.h>

//#include<algorithm>

//#include<iostream>

//using namespace std;

//int c1[60],c2[60];

//int a[27];

//int main()

//{

// int t;

// cin >> t;

// while(t --)

// {

// for(int i = 1; i <= 26; i ++)

// cin >> a[i];

// memset(c1,0,sizeof(c1));

// memset(c2,0,sizeof(c2));

// c1[0] = 1;///初始化

// for(int i = 1; i <= 26; i ++)///对应26个多项式

// {

// for(int j = 0; j <= 50; j ++) ///每个多项式中对应的指数

// for(int k = 0; k <= a[i] && k \* i + j <= 50; k ++) //k\*i表示被乘多项式各项的指数

// c2[j + k \* i] += c1[j];

// memcpy(c1,c2,sizeof(c2));///c2数组的值赋值给c1

// memset(c2,0,sizeof(c2));

// }

// int sum = 0;

// for(int i = 1; i <= 50; i ++)

// sum += c1[i];

// cout << sum << endl;

// }

// return 0;

//}

#include<iostream>

#include<string.h>

using namespace std;

int main()

{

int i,j,t,num,sum;

int X[27]; //保存Xi

int c1[51],c2[51]; /\*c1[51] 存储指数为1到 50 的项的系数 例如c1[5]表示 ax^5这一项的系数a\*/

scanf("%d",&t);

while(t--)

{

for(i=1;i<=26;i++)

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

memset(c1,0,sizeof(c1));

memset(c2,0,sizeof(c2));

for(i=1;i<=26;i++) //首先对第一个因子的各个系数初始化（称上面的式子中用一个括号括起来的为一个因子）

if(X[i]!=0) //Xi为0表示第i个因子为0(不存在) ;用第一个因子初始化c1

{

for(num=0;num<=X[i];num++)

{

if(num\*i>50)

break; //指数相乘如果大于50 则初始化完毕跳出

c1[num\*i]=1; //第一个因子里面的各个项的系数当然为1

}

break;

}

for(i=i+1;i<=26;i++)//从第二个乘数因子开始 字母开始做接下去的26-i次乘法

{

//每次循环模拟两个因子相乘, c1储存了前i个因子相乘得到的结果(每一项的系数)

for(j=0;j<=50;j++) //每一次模拟的计算范围控制在指数小于50

if(c1[j]!=0) //c1[j]==0 ,说明指数为j的那一项的系数为零，即该项不存在

for(num=0;num<=X[i];num++)

{

if(j+i\*num>50)

break; //两个相乘的数的指数相加，第i个因子的第k项的指数为num\*i

c2[j+i\*num]+=c1[j]\*1;

}

for(j=0;j<=50;j++)//把两个因子相乘之后的结果当做一个因子，再重新保存至c1

{

c1[j]=c2[j];

c2[j]=0;

}

}

for(sum=0,i=1;i<=50;i++)//前50项之和

sum+=c1[i];

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

}

return 0;

}

# 2110 Crisis of HDU

**Problem Description**

话说上回讲到HDU大战东洋小苟，结果自然是中方大胜，这一战也使得海东集团在全球同行业中的地位更加巩固。随着集团的发展，很多创业时期的元老逐步功成身退，先是8600移民海外，然后是linle夫妇退隐山林，逐渐的，最初众多的元老只剩下XHD夫妇和Wiskey三人了。  
到了2020年，因为扩张过度加上老鼠数量逐年减少，公司的发展遇到了前所未有的危机，此时集团已经没有任何流动资金，更可怕的是，这个时候，wiskey也决定退出了！  
退出本身并不麻烦，麻烦的是，退出的人需要取走相应比例（1/3）金额的资产。  
假设公司此时一共有n种价值的资产，每种价值的资产数量已知，请帮助心烦意乱的XHD夫妇计算一共有多少种分割资产的方法。

**Input**

输入包含多个测试实例，每个实例的第一行是一个整数n(n<100)，表示一共有n种价值的资产，接着的n行每行包含两个整数pi和mi(0<pi,mi<10)，分别表示某种价值和对应的数量，n为0的时候结束输入。

**Output**

对于每个测试实例，请输出分割资产的方案数%10000，如果不能分割，请输出“sorry”，每个实例的输出占一行。

**Sample Input**

2

1 1

2 1

0

**Sample Output**

1

#include <cstdio>

#include <cstdlib>

#include <climits>

#include <cstring>

#include <algorithm>

using namespace std;

const int MAX = 102;

int p[MAX],m[MAX],ans[MAX\*MAX],tmp[MAX\*MAX];

int n;

void work(int aim)

{

int i,j,k;

memset(ans,0,sizeof(ans));

memset(tmp,0,sizeof(tmp));

for(i=0;i<=m[0] && i\*p[0]<=aim;++i)

ans[i\*p[0]] = 1;

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

{

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

{

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

{

tmp[j+k\*p[i]] += ans[j];

tmp[j+k\*p[i]] %= 10000;

}

}

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

{

ans[j] = tmp[j];

tmp[j] = 0;

}

}

}

int main()

{

int i,all;

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

{

if(n==0)break;

all = 0;

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

{

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

all += (p[i] \* m[i]);

}

if(all%3!=0)

{

printf("sorry\n");

continue;

}

all /= 3;

work(all);

if(ans[all]!=0)

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

else

printf("sorry\n");

}

return 0;

}

# 2189 悼念512汶川大地震遇难同胞——来生一起走

**Problem Description**

正如温家宝总理所说“多难兴邦”，这场灾难让我们很多80后的年轻人一下子成熟了起来，其中很多人以自愿者的身份走上了抗震救灾的第一线。  
今天，灾区又来了n位志愿者，抗震救灾指挥部需要将他们分为若干个小组，小组的数量不限，但是要求每个小组的人数必须为素数，请问我们有几种分组的方法呢？  
  
特别说明：  
1、可以只有一个组；  
2、分组的方法只和人数有关，而与具体的人员无关，即：你可以假设人是无区别的。

**Input**

输入数据首先包含一个正整数C，表示有C组测试用例，然后是C行数据，每行包含一个正整数n(2<=n<=150)，表示志愿者的总人数。

**Output**

对于每组测试数据，请输出分组的方案数目，每个输出占一行。

**Sample Input**

3

3

4

5

**Sample Output**

1

1

2

//#include <stdio.h>

//#include <string.h>

//#include <math.h>

//#include <algorithm>

//using namespace std;

//

//int main()

//{

// int t,n,m,i,j,l;

// int c1[155],c2[155];

// int prime[50],k=0;

// for(i=2;i<150;i++)

// {

// int flat=0;

// for(j=2;j<=sqrt(i);j++)

// {

// if(i%j==0 && i!=j)

// {

// flat=1;;

// break;

// }

// }

// if(flat==0) prime[k++]=i;

// }

// scanf("%d",&t);

// while(t--)

// {

// scanf("%d",&n);

// memset(c1,0,sizeof(c1));

// memset(c2,0,sizeof(c2));

// for(i=0;i<=n;i+=prime[0])

// c1[i]=1;

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

// {

// if(prime[i]>n) break;

// for(l=0;l<=n;l++)

// {

// if(c1[l]==0) continue;

// for(j=0;j+l<=n;j+=prime[i])

// c2[l+j]+=c1[l];

// }

// for(l=0;l<=n;l++)

// {

// c1[l]=c2[l];

// c2[l]=0;

// }

// }

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

// }

// return 0;

//}

#include<stdio.h>

#include<string.h>

int a[151]={0},b[151],c[151];//数组a存储数字的因子和,数组b存储素数,数组c存储每个数的分法

int main()

{

memset(c,0,sizeof(c));//初始化为零

int T,n,i,j,t=0;

for(i=1;i<151;i++){//筛选法

if(a[i]==1)//因子和为1的就是素数

b[t++]=i;

for(j=2;i\*j<151;j++)

a[i\*j]+=i;

}

c[0]=1;

for(i=0;i<t;i++)//预处理,背包

for(j=b[i];j<151;j++)

c[j]+=c[j-b[i]];

scanf("%d",&T);

while(T--){

scanf("%d",&n);

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

}

}

# 2152 Fruit

**Problem Description**

转眼到了收获的季节，由于有TT的专业指导，Lele获得了大丰收。特别是水果，Lele一共种了N种水果，有苹果，梨子，香蕉，西瓜……不但味道好吃，样子更是好看。  
  
于是，很多人们慕名而来，找Lele买水果。  
  
甚至连大名鼎鼎的HDU ACM总教头 lcy 也来了。lcy抛出一打百元大钞，"我要买由M个水果组成的水果拼盘，不过我有个小小的要求，对于每种水果，个数上我有限制，既不能少于某个特定值，也不能大于某个特定值。而且我不要两份一样的拼盘。你随意搭配，你能组出多少种不同的方案，我就买多少份！"  
  
现在就请你帮帮Lele，帮他算一算到底能够卖出多少份水果拼盘给lcy了。  
  
注意，水果是以个为基本单位，不能够再分。对于两种方案，如果各种水果的数目都相同，则认为这两种方案是相同的。  
  
最终Lele拿了这笔钱，又可以继续他的学业了

**Input**

本题目包含多组测试，请处理到文件结束(EOF)。  
每组测试第一行包括两个正整数N和M(含义见题目描述，0<N,M<=100)  
接下来有N行水果的信息，每行两个整数A,B(0<=A<=B<=100)，表示至少要买该水果A个，至多只能买该水果B个。

**Output**

对于每组测试，在一行里输出总共能够卖的方案数。  
题目数据保证这个答案小于10^9

**Sample Input**

2 3

1 2

1 2

3 5

0 3

0 3

0 3

**Sample Output**

2

12

#include <iostream>

#include <cstring>

using namespace std;

#define min(a,b) ((a)<(b)?(a):(b))

int N,M,n1[200],n2[200],i,j,k;

int a[200],b[200];

int main()

{

while (cin>>N>>M)

{

for (i=0;i<N;i++)

cin>>n1[i]>>n2[i];

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

a[0]=1;

for (i=0;i<N;i++)

{

memset(b,0,sizeof(b));

for (j=n1[i];j<=n2[i]&&j<=M;j++)

for (k=0;k<=M&&k+j<=M;k++)

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

memcpy(a,b,sizeof(b));

}

cout<<a[M]<<endl;

}

return 0;

}

//#include <iostream>

//#include <cstring>

//using namespace std;

//#define min(a,b) ((a)<(b)?(a):(b))

//int N,M,n1[200],n2[200],i,j,k,last,last2;

//int a[200],b[200];

//int main()

//{

// while (cin>>N>>M)

// {

// for (i=0;i<N;i++)

// cin>>n1[i]>>n2[i];

// a[0]=1;

// last=0;

// for (i=0;i<N;i++)

// {

// last2=min(last+n2[i],M);

// memset(b,0,sizeof(int)\*(last2+1));

// for (j=n1[i];j<=n2[i]&&j<=last2;j++)

// for (k=0;k<=last&&k+j<=last2;k++)

// b[k+j]+=a[k];

// memcpy(a,b,sizeof(int)\*(last2+1));

// last=last2;

// }

// if (last>=M)

// cout<<a[M]<<endl;

// else

// cout<<0<<endl;

// }

// return 0;

//}