1.对于一个整数x来说，我们规定它的数位之和是这样来计算的：如果x是一个个位数，那么它的数位之和就是它本身；否则的话，x的数位之和就等于它的各位数字相加所得到的那个整数的数位之和。例如：5555的数位之和等于2，它是按照以下的步骤来计算的：  
5555 → 5 + 5 + 5 + 5 → 20 → 2 + 0 → 2  
编写一个程序，输入一个正整数（该整数可以用long long 存储），输出它的数位之和。  
输入格式：输入只有一行，即一个正整数。  
输出格式：输出相应的数位之和。 **注意：此题要求递归求解，如果出现非递归提交，无论是否通过，不论其他提交是否用了递归求解，此题均为0分。**

#### 输入样例

5555

#### 输出样例

2

#include<stdio.h>

int jisuan(int s);

int main(){

long long int s,shu;

scanf("%lld",&s);

shu=jisuan(s);

printf("%lld",shu);

return 0;

}

int jisuan(int s){

int a[10000];

int i=0,sum=0,k;

while(s>0){

a[i]=s%10;

s=s/10;

i++;

}

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

sum=sum+a[k];

}

if(sum>9){

return jisuan(sum);

}

else{

return sum;

}

}

2.美国职业橄榄球大联盟（National Football League，简称NFL）从1994年开始实施工资帽(Salary cap)制度，规定了每队球员工资总额的上限，而且是硬工资帽(Hardcap)，即任何球队每年的球员工资总额都不能超过工资帽。现请你写一段程序来计算一下在没有其他条件限制的情况下，理论上一个球队本赛季最多可以签多少名球员。

输入：第一行为两个用空格分隔的整数m和n（0<m,0<n<100）,依次代表硬工资帽（以万美元为单位），和球员总数。  
第二行为n个用空格分隔的整数，代表分别每个球员本赛季的工资（以万美元为单位）。

输出：为一个整数，最多可以签球员的个数。  
测试用例保证1、输入合法；2、所有整数都可以用int存储，3、球员总工资大于等于工资帽，4、至少能签1名球员。

#### 输入样例

17700 23

100 200 300 400 500 700 800 900 1000 1100 1500 1600 2000 2500 3000 3500 3700 4000 4500 4700 5000 5500 6000

#### 输出样例

15

#include<stdio.h>

int paixu(int a[],int n);

int shaixu(int a[],int n,int m);

int main(){

int m,n;

int i,j,k;

int sum;

int a[1000];

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

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

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

}

paixu(a,n);

sum=shaixu(a,n,m);

printf("%d",sum);

return 0;

}

int paixu(int a[],int n){

int i,k,t;

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

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

if(a[k]>a[k+1]){

t=a[k+1];

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

a[k]=t;

}

}

}

}

int shaixu(int a[],int n,int m){

int i=0,su=0;

while(su<m){

su=su+a[i];

i++;

}

return i-1;

}

3.根据给定阶数，输出方阵。具体格式见测试用例。

输入：为一个整数n(0<n<100)，代表方阵阶数。

输出：一个n\*n阶方阵，具体格式见测试用例（注意，每个数之间用一个空格分隔，每行的最后无空格）。

#### 输入样例

4

#### 输出样例

1 2 5 10

4 3 6 11

9 8 7 12

16 15 14 13

#include<stdio.h>

int main(){

int n;

int num[110][110]={};

scanf("%d",&n);

//先填第一列

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

num[i][1]=i\*i;

}

//由每列第一个向右延申

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

for(int j=2;j<=i;j++){

num[i][j]=num[i][j-1]-1;

}

}

//填第一行

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

num[1][i]=num[i-1][1]+1;

}

//由每行第一个数向下延申

for(int j=3;j<=n;j++){

for(int i=2;i<j;i++){

num[i][j]=num[i-1][j]+1;

}

}

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

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

printf("%d ",num[i][j]);

}

printf("%d",num[i][n]);

printf("\n");

}

return 0;

}

4.在桥牌中大牌点的英文全称是High Card Point,简写为hcp 。它是衡量一手牌好坏的标准。我们规定A=4hcp K=3hcp Q=2hcp J=1hcp.由此，拿到一手牌我们可以很快地计算出他的hcp。我们也可以很快知道，一副牌一共是40Hcp。这个值对于每副牌来说，就是东南西北4个人手上大牌点的总和。

现我们这里有另一种扑克的玩法，其计算牌点的方式如下，请你写一段程序来衡量一下一手排的好坏。  
牌点计算方式：2，3，4均为1点，5，6均为2点，7，8均为3点，9，T（代表10）均为4点，J为6点，Q为7点，K为8点，A为10点。

输入：  
第一行为一个整数n（0<n<50）,共n手牌；  
后边为n行，每行为一个长度不超过20的字符串，代表一手牌（字符串中只可能包含字符23456789TJQKA）。

输入：  
共n行，将输入的n手牌按牌的点数依次从小到大输出（测试用例保证没有点数相同的牌）。

#### 输入样例

5

7287K2TQ5

556AKK54K

KQJ96AQ45

42J954TJK

23A49A27A

#### 输出样例

7287K2TQ5

42J954TJK

23A49A27A

556AKK54K

KQJ96AQ45

#include<stdio.h>

#include<string.h>

int main(){

int n,i,j=0,t;

char str[50][21];

int s[50]={0};

scanf("%d",&n);

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

scanf("%s",&str[i]);

}

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

for(j=0;str[i][j]!='\0';j++){

if(str[i][j]=='2'||str[i][j]=='3'||str[i][j]=='4')

s[i]+=1;

else if(str[i][j]=='5'||str[i][j]=='6')

s[i]+=2;

else if(str[i][j]=='7'||str[i][j]=='8')

s[i]+=3;

else if(str[i][j]=='9'||str[i][j]=='T')

s[i]+=4;

else if(str[i][j]=='J')

s[i]+=6;

else if(str[i][j]=='Q')

s[i]+=7;

else if(str[i][j]=='K')

s[i]+=8;

else if(str[i][j]=='A')

s[i]+=10;

}

}

char w[50];

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

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

if(s[j]>s[j+1]){

t=s[j+1];

s[j+1]=s[j];

s[j]=t;

strcpy(w,str[j]);

strcpy(str[j],str[j+1]);

strcpy(str[j+1],w);

}

}

}

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

printf("%s\n",str[i]);

}

}5.现在有N个集装箱的货物要运走，你的任务是把这N箱的货物装到M艘船上。约束条件为：  
1、集装箱是依次一箱一箱的送到码头，来到码头后必须立刻装到船上。船也是依次一艘一艘的来到码头，一艘船装完后立刻离开，然后才能装下一艘；  
2、集装箱不能拆箱；  
3、所有的船具有相同的载重量，假设为W，船上装的集装箱的总重量不能超过这个载重量；  
你的任务是计算出W的最小值。

样例解释：第1艘船装（10，40）、第2艘船装（30，10）、第3艘船装（50）、第4艘船装（11）、第5艘船装（40）。这种情况下W为50，其他方式W均要大于50。

输入：第一行为用空格分隔的2个整数，分别代表N和M。（1 <= N <= 100000 ，1 <= M <= N），接下来N行，每行为一个整数，依次(输入的顺序即是集装箱到达码头的顺序)代表每个集装箱的重量（该重量大于等于1且小于等于10000）。

输出：只有一行，为一个整数，代表此条件下W的最小值。

提示：二分查找

输入样例

7 5

10

40

30

10

50

11

40

输出样例

50