

# Assignment #2: 编程练习

---

Updated 0953 GMT+8 Feb 24, 2024

2024 spring, Compiled by ==陈亚偲 工学院==

## 说明:

1) The complete process to learn DSA from scratch can be broken into 4 parts:

- Learn about Time and Space complexities
- Learn the basics of individual Data Structures
- Learn the basics of Algorithms
- Practice Problems on DSA

2) 请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用 word）。AC 或者没有AC，都请标上每个题目大致花费时间。

3) 课程网站是Canvas平台, <https://pku.instructure.com>, 学校通知3月1日导入选课名单后启用。**作业写好后，保留在自己手中，待3月1日提交。**

提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、“作业评论”区有上传的md或者doc附件。

4) 如果不能在截止前提交作业，请写明原因。

## 编程环境

==（请改为同学的操作系统、编程环境等）==

操作系统：Windows

Python编程环境：Spyder IDE 5.2.2

## 1. 题目

---

### 27653: Fraction类

[http://cs101.openjudge.cn/2024sp\\_routine/27653/](http://cs101.openjudge.cn/2024sp_routine/27653/)

思路：定义一个类，专门处理分数（self.num多余了，但忘删了）

## 代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Tue Feb 20 21:28:03 2024

@author: 陈亚德 2300011106
"""
def daza(n):
    ans=[0]*2+[1]*(n-1)
    kk=set()
    for i in range(2,n+1):
        if ans[i]==0:
            continue
        else:
            kk.add(i)
            for j in range(2*i,n+1,i):
                ans[j]=0
    return kk
class boza:
    def __init__(self,zi,mu):
        self.zi=zi
        self.mu=mu
        self.num=zi/mu
    def cyc(self):
        return str(self.zi)+'/'+str(self.mu)
def add(a,b,c,d):
    def ciza(k,m):
        mm=min([k,m])
        da=daza(mm)
        for i in da:
            while k%i==0 and m%i==0:
                k=k//i
                m=m//i
            return [k,m]
    return ciza(a*d+b*c,b*d)
a,b,c,d=map(int,input().split())
m,n=add(a,b,c,d)[0],add(a,b,c,d)[1]
eza=boza(m,n)
print(eza.cyc())
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Feb 20 21:28:03 2024

@author: 陈亚德 2300011106
"""
def daza(n):
    ans=[0]*2+[1]*(n-1)
    kk=set()
    for i in range(2,n+1):
        if ans[i]==0:
            continue
        else:
            kk.add(i)
            for j in range(2*i,n+1,i):
                ans[j]=0
    return kk
class boza:
    def __init__(self,zi,mu):
        self.zi=zi
        self.mu=mu
        self.num=zi/mu
    def cyc(self):
        return str(self.zi)+'/'+str(self.mu)
def add(a,b,c,d):
    def ciza(k,m):
        mm=min([k,m])
        da=daza(mm)
        for i in da:
            while k%i==0 and m%i==0:
                k=k//i
                m=m//i
        return [k,m]
    return ciza(a*d+b*c,b*d)
a,b,c,d=map(int,input().split())
m,n=add(a,b,c,d)[0],add(a,b,c,d)[1]
eza=boza(m,n)
print(eza.cyc())
```

## 04110: 圣诞老人的礼物-Santa Clau's Gifts

greedy/dp, <http://cs101.openjudge.cn/practice/04110>

思路: 直接大排位

最后的操作是四舍五入

## 代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 20:43:04 2023

@author: 陈亚偲2300011106
"""
n,m=map(int,input().split())
a=[]
for i in range(n):
    p,q=map(int,input().split())
    for sdvbhjf in range(q):
        a.append(float(p/q))
a.sort(reverse=True)

print(round(sum(a[:m])+0.00000001,1))
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 20:43:04 2023

@author: 陈亚偲2300011106
"""
n,m=map(int,input().split())
a=[]
for i in range(n):
    p,q=map(int,input().split())
    for sdvbhjf in range(q):
        a.append(float(p/q))
a.sort(reverse=True)

print(round(sum(a[:m])+0.00000001,1))
```

©2002-2022 POJ 京ICP备20010980号-1

## 18182: 打怪兽

implementation/sortings/data structures, <http://cs101.openjudge.cn/practice/18182/>

思路:

有技能就放, 挑猛的放

用字典做

## 代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Sat Feb 24 11:27:39 2024

@author: 陈亚德 2300011106
"""
nC=int(input())
anss=[]
for i in range(nC):
    ans='alive'
    n,m,b=map(int,input().split())
    a={}
    for w in range(n):
        p,q=map(int,input().split())
        if p in a.keys():
            a[p].append(q)
        else:
            a[p]=[q]
    a={f:g for f,g in sorted(a.items())}
    for z in a.keys():
        a[z].sort()
        for j in range(min([len(a[z]),m])):
            b-=a[z][-j-1]
        if b<=0:
            ans=z
            break
    anss.append(ans)
for i in anss:
    print(i)
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Sat Feb 24 11:27:39 2024

@author: 陈亚德 2300011106
"""
nC=int(input())
anss=[]
for i in range(nC):
    ans='alive'
    n,m,b=map(int,input().split())
    a={}
    for w in range(n):
        p,q=map(int,input().split())
        if p in a.keys():
            a[p].append(q)
        else:
            a[p]=[q]
    a={f:g for f,g in sorted(a.items())}
    for z in a.keys():
        a[z].sort()
        for j in range(min([len(a[z]),m])):
            b-=a[z][-j-1]
        if b<=0:
            ans=z
            break
    anss.append(ans)
for i in anss:
    print(i)
```

## 230B. T-primes

binary search/implementation/math/number theory, 1300, <http://codeforces.com/problemset/problem/230/B>

思路：找质数的平方

结尾用set () 快

代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 12 14:38:42 2023

@author: 陈亚德2300011106
"""

n=int(input())
a=[int(i) for i in input().split()]
```

```

d=set()
p=1000000
b=[False]+[True]*(p-1)
for i in range(p):
    if b[i]:
        for j in range(2,p//(i+1)+1):
            b[j*(i+1)-1]=False
for i in range(p):
    if b[i]:
        d.add((i+1)*(i+1))
for i in a:
    if i in d:
        print('YES')
    else:
        print('NO')

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

General				
#	Author	Problem	Lang	Verdict
227805816	Practice: Kaliyaboza	<a href="#">230B</a> - 28	Python 3	Accepted

→ [Source](#)

```

# -*- coding: utf-8 -*-
"""
Created on Thu Oct 12 14:38:42 2023

@author: 陈亚德2300011106
"""

n=int(input())
a=[int(i) for i in input().split()]
d=set()
p=1000000
b=[False]+[True]*(p-1)
for i in range(p):
    if b[i]:
        for j in range(2,p//(i+1)+1):
            b[j*(i+1)-1]=False
for i in range(p):
    if b[i]:
        d.add((i+1)*(i+1))
for i in a:
    if i in d:
        print('YES')
    else:
        print('NO')

```

[Click](#) to see test details

## 1364A. XXXXX

brute force/data structures/number theory/two pointers, 1200, <https://codeforces.com/problemset/problem/1364/A>

思路:

在头尾找即可，最后几行是精髓，这样可以两边向中间扫

代码

```
#
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 9 16:14:28 2023

@author: 陈亚偲2300011106
"""
A=[]
n=int(input())
for i in range(n):
    x,y=map(int,input().split())
    a=[int(i) for i in input().split()]
    ct=True
    for apple in a:
        if apple % y != 0:
            ct=False
    if ct:
        A.append(-1)
    elif (sum(a))%y!=0:
        A.append(x)
    else:
        for i in range(x//2+1):
            if a[i] % y!=0 or a[x-i-1] %y!=0:
                A.append(x-i-1)
                break
for i in A:
    print(i)
```

代码运行截图 == (AC代码截图，至少包含有"Accepted") ==



## General

#	Author	Problem	Lang	Verdict
227293851	Practice: Kaliyaboza	<a href="#">1364A</a> - 15	Python 3	Accepted

## → Source

```

# -*- coding: utf-8 -*-
"""
Created on Mon Oct  9 16:14:28 2023

@author: 陈亚德2300011106
"""
A=[]
n=int(input())
for i in range(n):
    x,y=map(int,input().split())
    a=[int(i) for i in input().split()]
    ct=True
    for apple in a:
        if apple % y != 0:
            ct=False
    if ct:
        A.append(-1)
    elif (sum(a))%y!=0:
        A.append(x)
    else:
        for i in range(x//2+1):
            if a[i] % y!=0 or a[x-i-1] %y!=0:
                A.append(x-i-1)
                break
for i in A:
    print(i)

```

[Click](#) to see test details

## 18176: 2050年成绩计算

<http://cs101.openjudge.cn/practice/18176/>

思路：

暴力打表（其实可以用欧拉筛，但打表一定更快）

代码

```

#

```

b={4, 9, 25, 49, 121, 169, 289, 361, 529, 841, 961, 1369, 1681, 1849, 2209, 2809, 3481, 3721, 4489, 5041, 5329, 6241, 6889, 7921, 9409, 10201, 10609, 11449, 11881, 12769, 16129, 17161, 18769, 19321, 22201, 22801, 24649, 26569, 27889, 29929, 32041, 32761, 36481, 37249, 38809, 39601, 44521, 49729, 51529, 52441, 54289, 57121, 58081, 63001, 66049, 69169, 72361, 73441, 76729, 78961, 80089, 85849, 94249, 96721, 97969, 100489, 109561, 113569, 120409, 121801, 124609, 128881, 134689, 139129, 143641, 146689, 151321, 157609, 160801, 167281, 175561, 177241, 185761, 187489, 192721, 196249, 201601, 208849, 212521, 214369, 218089, 229441, 237169, 241081, 249001, 253009, 259081, 271441, 273529, 292681, 299209, 310249, 316969, 323761, 326041, 332929, 344569, 351649, 358801, 361201, 368449, 375769, 380689, 383161, 398161, 410881, 413449, 418609, 426409, 434281, 436921, 452929, 458329, 466489, 477481, 491401, 502681, 516961, 528529, 537289, 546121, 552049, 564001, 573049, 579121, 591361, 597529, 619369, 635209, 654481, 657721, 674041, 677329, 683929, 687241, 703921, 727609, 734449, 737881, 744769, 769129, 776161, 779689, 786769, 822649, 829921, 844561, 863041, 877969, 885481, 896809, 908209, 935089, 942841, 954529, 966289, 982081, 994009, 1018081, 1026169, 1038361, 1042441, 1062961, 1067089, 1079521, 1100401, 1104601, 1125721, 1129969, 1142761, 1181569, 1190281, 1194649, 1203409, 1216609, 1229881, 1247689, 1261129, 1274641, 1324801, 1329409, 1352569, 1371241, 1394761, 1408969, 1423249, 1442401, 1471369, 1481089, 1495729, 1510441, 1515361, 1530169, 1560001, 1585081, 1630729, 1635841, 1646089, 1661521, 1666681, 1682209, 1692601, 1697809, 1708249, 1739761, 1745041, 1760929, 1852321, 1868689, 1885129, 1907161, 1957201, 1985281, 2024929, 2036329, 2042041, 2053489, 2070721, 2093809, 2105401, 2111209, 2128681, 2163841, 2193361, 2199289, 2211169, 2217121, 2229049, 2247001, 2283121, 2319529, 2343961, 2380849, 2399401, 2411809, 2430481, 2455489, 2468041, 2493241, 2505889, 2550409, 2563201, 2582449, 2588881, 2601769, 2621161, 2627641, 2647129, 2679769, 2745649, 2765569, 2778889, 2785561, 2866249, 2879809, 2886601, 2920681, 2961841, 2968729, 3003289, 3031081, 3052009, 3073009, 3094081, 3157729, 3179089, 3193369, 3200521, 3243601, 3279721, 3323329, 3352561, 3411409, 3463321, 3485689, 3500641, 3508129, 3523129, 3530641, 3568321, 3613801, 3636649, 3659569, 3728761, 3736489, 3798601, 3806401, 3892729, 3916441, 3948169, 3972049, 3988009, 3996001, 4012009, 4044121, 4068289, 4108729, 4116841, 4157521, 4214809, 4255969, 4280761, 4330561, 4338889, 4355569, 4363921, 4405801, 4456321, 4464769, 4532641, 4541161, 4566769, 4583881, 4592449, 4635409, 4669921, 4748041, 4853209, 4870849, 4897369, 4932841, 5004169, 5013121, 5031049, 5067001, 5139289, 5148361, 5166529, 5202961, 5230369, 5257849, 5276209, 5331481, 5340721, 5442889, 5470921, 5480281, 5508409, 5527201, 5555449, 5621641, 5650129, 5669161, 5678689, 5707321, 5726449, 5755201, 5812921, 5841889, 5870929, 5938969, 5958481, 5987809, 6046681, 6086089, 6115729, 6135529, 6265009, 6355441, 6405961, 6446521, 6466849, 6497401, 6507601, 6538249, 6651241, 6713281, 6723649, 6806881, 6848689, 6869641, 6932689, 7006609, 7059649, 7070281, 7091569, 7134241, 7166329, 7198489, 7219969, 7230721, 7252249, 7284601, 7327849, 7349521, 7360369, 7392961, 7447441, 7458361, 7513081, 7557001, 7579009, 7656289, 7711729, 7778521, 7789681, 7823209, 7845601, 7856809, 7946761, 8025889, 8048569, 8082649, 8128201, 8162449, 8185321, 8288641, 8334769, 8392609, 8427409, 8462281, 8508889, 8567329, 8637721, 8720209, 8743849, 8779369, 8814961, 8826841, 8994001, 9006001, 9066121, 9114361, 9138529, 9223369, 9247681, 9296401, 9369721, 9406489, 9480241, 9504889, 9541921, 9665881, 9728161, 9740641, 9840769, 10004569, 10029889, 10042561, 10118761, 10156969, 10182481, 10259209, 10297681, 10349089, 10374841, 10426441, 10569001, 10582009, 10608049, 10621081, 10699441, 10883401, 10896601, 10936249, 10975969, 11015761, 11042329, 11082241, 11095561, 11175649, 11202409, 11282881, 11296321, 11363641, 11377129, 11485321, 11498881, 11607649, 11648569, 11785489, 11895601, 11950849, 11978521, 11992369, 12020089, 12033961, 12187081, 12243001, 12327121, 12369289, 12439729, 12453841, 12482089, 12524521, 12538681, 12581209, 12652249, 12666481, 12752041, 12823561, 12837889, 12909649, 13010449, 13053769, 13082689, 13126129, 13184161, 13227769, 13271449, 13388281, 13476241, 13490929, 13520329, 13623481, 13667809, 13697401, 13756681, 13830961, 13890529, 13935289,

13980121, 14145121, 14190289, 14205361, 14280841, 14386849, 14417209, 14462809,  
14600041, 14615329, 14691889, 14799409, 14830201, 14845609, 14922769, 15031129,  
15062161, 15124321, 15264649, 15295921, 15342889, 15358561, 15389929, 15437041,  
15452761, 15547249, 15578809, 15737089, 15912121, 16008001, 16024009, 16056049,  
16104169, 16152361, 16168441, 16216729, 16394401, 16410601, 16459249, 16589329,  
16638241, 16736281, 16752649, 16801801, 16900321, 17032129, 17048641, 17081689,  
17131321, 17247409, 17280649, 17297281, 17447329, 17648401, 17732521, 17783089,  
17799961, 17884441, 17901361, 17986081, 18003049, 18088009, 18139081, 18156121,  
18241441, 18258529, 18344089, 18395521, 18464209, 18722929, 18809569, 18826921,  
18913801, 18983449, 19035769, 19123129, 19280881, 19333609, 19439281, 19545241,  
19562929, 19722481, 19775809, 19811401, 19864849, 19918369, 20079361, 20097289,  
20187049, 20313049, 20367169, 20403289, 20421361, 20457529, 20675209, 20693401,  
20802721, 20857489, 21003889, 21077281, 21132409, 21187609, 21353641, 21501769,  
21520321, 21557449, 21613201, 21631801, 21687649, 21743569, 21836929, 21893041,  
22005481, 22118209, 22287841, 22306729, 22363441, 22401289, 22572001, 22648081,  
22877089, 22915369, 22934521, 22972849, 23030401, 23049601, 23164969, 23203489,  
23338561, 23629321, 23726641, 23785129, 23902321, 24039409, 24098281, 24196561,  
24314761, 24334489, 24373969, 24433249, 24512401, 24571849, 24671089, 24690961,  
24730729, 24870169, 24930049, 24990001, 25030009, 25090081, 25110121, 25210441,  
25230529, 25391521, 25512601, 25593481, 25775929, 25816561, 25877569, 25999801,  
26020201, 26081449, 26142769, 26204161, 26491609, 26553409, 26697889, 26739241,  
26822041, 26925721, 27008809, 27133681, 27321529, 27363361, 27384289, 27426169,  
27678121, 27804529, 27867841, 27888961, 28058209, 28121809, 28185481, 28334329,  
28440889, 28590409, 28633201, 28955161, 29019769, 29084449, 29149201, 29235649,  
29300569, 29343889, 29365561, 29495761, 29560969, 29604481, 29626249, 29691601,  
29931841, 29997529, 30019441, 30063289, 30261001, 30283009, 30327049, 30459361,  
30481441, 30547729, 30591961, 30880249, 30946969, 31013761, 31058329, 31147561,  
31259281, 31618129, 31798321, 31820881, 31888609, 31933801, 31956409, 32001649,  
32024281, 32137561, 32296489, 32364721, 32410249, 32501401, 32615521, 32684089,  
32913169, 32959081, 32982049, 33051001, 33396841, 33443089, 33535681, 33651601,  
33721249, 33790969, 33884041, 33953929, 34093921, 34140649, 34210801, 34234201,  
34304449, 34351321, 34421689, 34445161, 34562641, 34586161, 34774609, 34845409,  
35081929, 35129329, 35271721, 35438209, 35772361, 35844169, 36084049, 36132121,  
36348841, 36445369, 36517849, 36566209, 36638809, 36808489, 36881329, 36954241,  
37075921, 37100281, 37222201, 37368769, 37466641, 37589161, 37613689, 37736449,  
37834801, 37982569, 38105929, 38402809, 38427601, 38477209, 38576521, 38651089,  
38700841, 38800441, 39025009, 39150049, 39225169, 39300361, 39325441, 39400729,  
39526369, 39677401, 39702601, 39828721, 39904489, 39980329, 40056241, 40157569,  
40233649, 40360609, 40436881, 40462321, 40538689, 40615129, 40691641, 40819321,  
40921609, 41229241, 41306329, 41589601, 41615401, 41847961, 41899729, 42003361,  
42133081, 42523441, 42627841, 42863209, 42915601, 42941809, 43072969, 43151761,  
43178041, 43256929, 43309561, 43546801, 43652449, 43811161, 44049769, 44262409,  
44342281, 44368921, 44528929, 44609041, 44742721, 44769481, 44903401, 44930209,  
45010681, 45144961, 45333289, 45387169, 45711121, 45738169, 45954841, 45981961,  
46117681, 46144849, 46280809, 46553329, 46607929, 46635241, 46689889, 46799281,  
47018449, 47100769, 47183161, 47210641, 47375689, 47596201, 47706649, 47761921,  
47844889, 48260809, 48288601, 48427681, 48455521, 48539089, 48594841, 48678529,  
48762289, 48874081, 48958009, 49014001, 49182169, 49266361, 49378729, 49547521,  
49603849, 49801249, 49970761, 50112241, 50452609, 50537881, 50708641, 50794129,  
50822641, 51136801, 51251281, 51509329, 51652969, 51739249, 51940849, 51998521,  
52027369, 52113961, 52258441, 52374169, 52461049, 52519009, 52606009, 53042089,  
53246209, 53392249, 53421481, 53597041, 53743561, 53772889, 54007801, 54037201,  
54302161, 54656449, 54922921, 55011889, 55249489, 55517401, 55606849, 55636681,  
55905529, 55965361, 56055169, 56085121, 56235001, 56355049, 56505289, 56595529,  
56685841, 56806369, 56866681, 56957209, 56987401, 57138481, 57168721, 57350329,  
57410929, 57501889, 57592921, 57623281, 57805609, 57866449, 58079641, 58354321,

```

58415449, 58507201, 58813561, 58874929, 58997761, 59089969, 59151481, 59274601,
59336209, 59552089, 59644729, 59706529, 59923081, 60109009, 60171049, 60202081,
60668521, 60730849, 61105489, 61199329, 61293241, 61481281, 61669609, 61889689,
61984129, 62047129, 62078641, 62141689, 62425801, 62520649, 62710561, 62837329,
62932489, 62995969, 63186601, 63218401, 63409369, 63888049, 64144081, 64176121,
64272289, 64625521, 64850809, 64947481, 65108761, 65302561, 65399569, 65431921,
65496649, 65626201, 65788321, 65885689, 65983129, 66373609, 66601921, 66699889,
66765241, 66896041, 67092481, 67387681, 67551961, 67584841, 67749361, 67782289,
67848169, 67947049, 68277169, 68376361, 68442529, 68674369, 68740681, 68773849,
68840209, 69072721, 69172489, 69372241, 69772609, 69939769, 70040161, 70174129,
70341769, 70375321, 70879561, 70946929, 71048041, 71081761, 71284249, 71351809,
71588521, 71690089, 72267001, 72471169, 72607441, 72709729, 72880369, 72914521,
72982849, 73324969, 73496329, 73633561, 73908409, 73942801, 74114881, 74356129,
74425129, 74459641, 74666881, 74770609, 75047569, 75151561, 75290329, 75359761,
75498721, 75568249, 75672601, 75811849, 75916369, 76020961, 76230361, 76335169,
76405081, 76510009, 76615009, 76755121, 77070841, 77141089, 77492809, 77563249,
77774761, 77810041, 77986561, 78092569, 78127921, 78304801, 78517321, 78552769,
78623689, 78978769, 79085449, 79619929, 79727041, 79798489, 79941481, 80120401,
80335369, 80442961, 80478841, 80982001, 81018001, 81126049, 81198121, 81234169,
81522841, 81739681, 81775849, 81884401, 82065481, 82210489, 82646281, 82864609,
82973881, 83302129, 83411689, 83484769, 83740801, 83850649, 83923921, 84143929,
84290761, 84400969, 84621601, 84695209, 84805681, 85026841, 85137529, 85359121,
85396081, 85692049, 86062729, 86136961, 86174089, 86359849, 86694721, 86843761,
86918329, 87179569, 87254281, 87291649, 87403801, 87815641, 87928129, 88190881,
88303609, 88416409, 88604569, 88717561, 88755241, 88943761, 88981489, 89056969,
89094721, 89510521, 89548369, 89624089, 89737729, 89851441, 90079081, 90193009,
90459121, 90649441, 90878089, 90992521, 91145209, 91221601, 91910569, 92179201,
92409769, 92525161, 92602129, 92717641, 92756161, 92987449, 93103201, 93334921,
93644329, 93683041, 93876721, 94031809, 94458961, 94497841, 94731289, 94848121,
94926049, 95043001, 95394289, 95433361, 95667961, 95785369, 95863681, 96098809,
96255721, 96373489, 96609241, 96687889, 96805921, 97042201, 97160449, 97199881,
97436641, 97673689, 97752769, 98029801, 98148649, 98465929, 98585041, 98624761,
98823481, 98982601, 99341089, 99460729}
aa=[]
m,n=map(int,input().split())
for i in range(m):
    k=[int(j) for j in input().split()]
    ww=len(k)
    for pp in range(ww):
        if k[pp] not in b:
            k[pp]=0
    ans=round(sum(k)/ww,2)
    ank=list(str(ans))
    if str(ans)=='0.0':
        aa.append(0)
    else:
        if ank[-2]=='.':
            ank.append('0')
        aa.append(''.join(ank))
for i in aa:
    print(i)

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
b={4, 9, 25, 49, 121, 169, 289, 361, 529, 841, 961, 1369, 1681, 1849, 2209}
aa=[]
m,n=map(int,input().split())
for i in range(m):
    k=[int(j) for j in input().split()]
    ww=len(k)
    for pp in range(ww):
        if k[pp] not in b:
            k[pp]=0
    ans=round(sum(k)/ww,2)
    ank=list(str(ans))
    if str(ans)=='0.0':
        aa.append(0)
    else:
        if ank[-2]=='.':
            ank.append('0')
        aa.append(''.join(ank))
for i in aa:
    print(i)
```

## 2. 学习总结和收获

==如果作业题目简单，有否额外练习题目，比如：OJ“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。==

set () 非常快，能用就用

元组给我的感觉是比表格快的不多，更多时候表格是元组的上位