Assignment #3: 语法练习

Updated 1600 GMT+8 Sep 25, 2023

2023 fall, Complied by 刘思瑞 元培学院

说明：

1）第2周课上讲到了计算机相关的历史，介绍了ASCII表。

2）请把每个题目解题思路（可选），源码Python, 或者C++/C（已经在Codeforces/Openjudge上AC），截图（包含Accepted, 学号），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）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、作业评论有md或者doc。

4）同学完成作业的时候，就是这个模版文件中修改补充好。为便于助教批改作业，请尽量不要删除其他文字。

5）如果不能在截止前提交作业，请写明原因。

编程环境

操作系统：Windows 11 22H2 22621.2283

Python编程环境：Visual Studio (1.82.2); python 3.11.3

C/C++编程环境：无

1. 必做题目

118A. String Task

implementation/strings, 1000, http://codeforces.com/problemset/problem/118/A

Petya started to attend programming lessons. On the first lesson his task was to write a simple program. The program was supposed to do the following: in the given string, consisting if uppercase and lowercase Latin letters, it:

- deletes all the vowels,
- inserts a character "." before each consonant,
- replaces all uppercase consonants with corresponding lowercase ones.

Vowels are letters "A", "O", "Y", "E", "U", "I", and the rest are consonants. The program's input is exactly one string, it should return the output as a single string, resulting after the program's processing the initial string.

Help Petya cope with this easy task.

Input

The first line represents input string of Petya's program. This string only consists of uppercase and lowercase Latin letters and its length is from 1 to 100, inclusive.

Output

Print the resulting string. It is guaranteed that this string is not empty.

Examples

input

```
tour
```

output

```
.t.r
```

input

```
Codeforces
```

output

```
.c.d.f.r.c.s
```

input

```
aBAcAba
```

output

```
.b.c.b
```

【刘思瑞，元培物理方向，2023年秋】

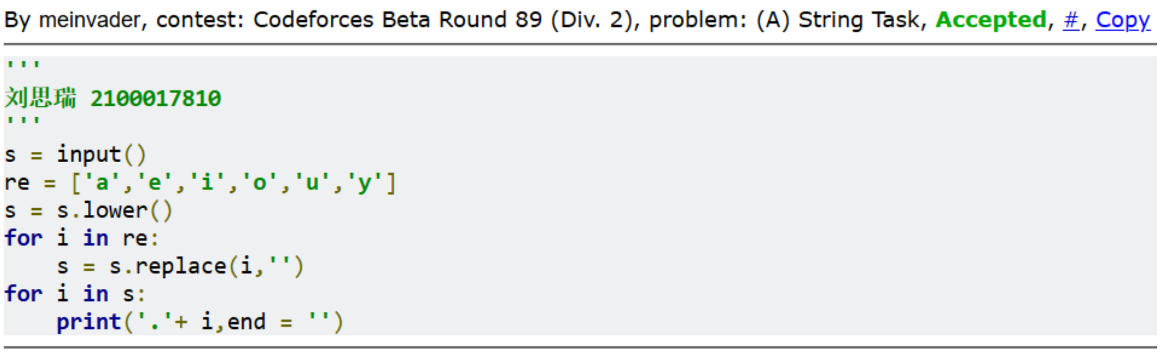
思路：

直接把读入的字符串小写然后调用replace方法（一开始不知道这个感觉自己写实在太麻烦所以上网搜了一下hhh）

代码

```python
'''
刘思瑞 2100017810
'''
s = input()
re = ['a','e','i','o','u','y']
s = s.lower()
for i in re:
    s = s.replace(i,'')
for i in s:
    print('.'+ i,end = '')
```

代码运行截图



By meinvader, contest: Codeforces Beta Round 89 (Div. 2), problem: (A) String Task, **Accepted**, #, Copy

```python
'''
刘思瑞 2100017810
'''
s = input()
re = ['a','e','i','o','u','y']
s = s.lower()
for i in re:
    s = s.replace(i,'')
for i in s:
    print('.'+ i,end = '')
```

263A. Beautiful Matrix

implementation, 800, http://codeforces.com/problemset/problem/263/A

You've got a 5 × 5 matrix, consisting of 24 zeroes and a single number one. Let's index the matrix rows by numbers from 1 to 5 from top to bottom, let's index the matrix columns by numbers from 1 to 5 from left to right. In one move, you are allowed to apply one of the two following transformations to the matrix:

1. Swap two neighboring matrix rows, that is, rows with indexes i and $i + 1$ for some integer i ($1 \le i < 5$).
2. Swap two neighboring matrix columns, that is, columns with indexes j and $j + 1$ for some integer j ($1 \le j < 5$).

You think that a matrix looks *beautiful*, if the single number one of the matrix is located in its middle (in the cell that is on the intersection of the third row and the third column). Count the minimum number of moves needed to make the matrix beautiful.

Input

The input consists of five lines, each line contains five integers: the j-th integer in the i-th line of the input represents the element of the matrix that is located on the intersection of the i-th row and the j-th column. It is guaranteed that the matrix consists of 24 zeroes and a single number one.

Output

Print a single integer — the minimum number of moves needed to make the matrix beautiful.

Examples

input

```
0 0 0 0 0
0 0 0 0 1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

output

```
3
```

input

```
0 0 0 0 0
0 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
```

output

```
1
```

【刘思瑞，元培物理方向，2023年秋】

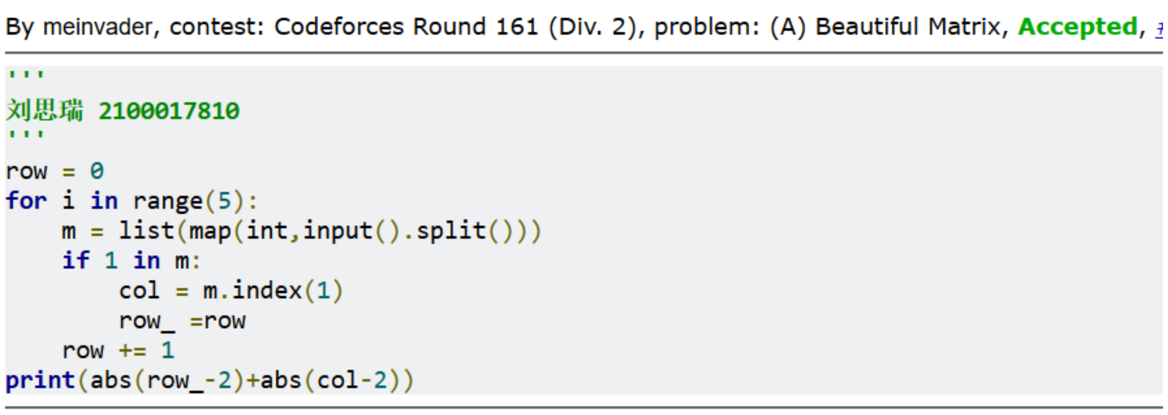
思路：

这个题本来很简单，但是一定要把所有数据都读进去（（

代码

```python
'''
刘思瑞 2100017810
'''
row = 0
for i in range(5):
    m = list(map(int,input().split()))
    if 1 in m:
        col = m.index(1)
        row_ =row
    row += 1
print(abs(row_-2)+abs(col-2))
```

代码运行截图



By meinvader, contest: Codeforces Round 161 (Div. 2), problem: (A) Beautiful Matrix, **Accepted**,

```python
'''
刘思瑞 2100017810
'''
row = 0
for i in range(5):
    m = list(map(int,input().split()))
    if 1 in m:
        col = m.index(1)
        row_ =row
    row += 1
print(abs(row_-2)+abs(col-2))
```

281A. Word Capitalization

implementation/strings, 800, http://codeforces.com/problemset/problem/281/A

Capitalization is writing a word with its first letter as a capital letter. Your task is to capitalize the given word.

Note, that during capitalization all the letters except the first one remains unchanged.

Input

A single line contains a non-empty word. This word consists of lowercase and uppercase English letters. The length of the word will not exceed 10^3.

Output

Output the given word after capitalization.

Examples

input

```
ApPLe
```

output

```
ApPLe
```

input

```
konjac
```

output

```
Konjac
```

【刘思瑞，元培物理方向，2023年秋】

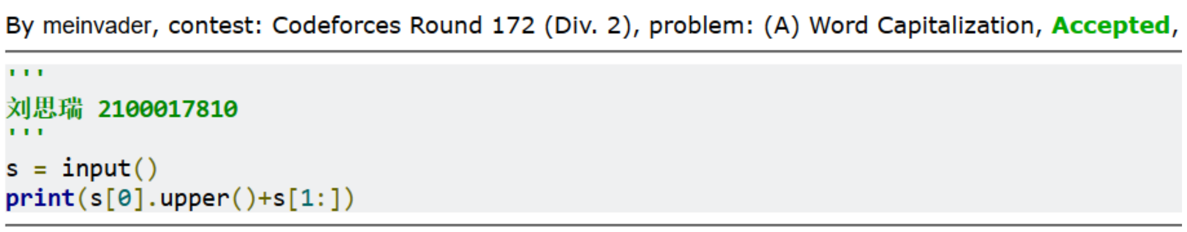
思路：

切片

代码

```python
'''
刘思瑞 2100017810
'''
s = input()
print(s[0].upper()+s[1:])
```

代码运行截图



By meinvader, contest: Codeforces Round 172 (Div. 2), problem: (A) Word Capitalization, Accepted,

```python
'''
刘思瑞 2100017810
'''
s = input()
print(s[0].upper()+s[1:])
```

282A. Bit++

implementation, 800, http://codeforces.com/problemset/problem/282/A

The classic programming language of Bitland is Bit++. This language is so peculiar and complicated.

The language is that peculiar as it has exactly one variable, called *x*. Also, there are two operations:

- Operation ++ increases the value of variable *x* by 1.
- Operation -- decreases the value of variable *x* by 1.

A statement in language Bit++ is a sequence, consisting of exactly one operation and one variable *x*. The statement is written without spaces, that is, it can only contain characters "+", "-", "X". Executing a statement means applying the operation it contains.

A programme in Bit++ is a sequence of statements, each of them needs to be executed. Executing a programme means executing all the statements it contains.

You're given a programme in language Bit++. The initial value of *x* is 0. Execute the programme and find its final value (the value of the variable when this programme is executed).

Input

The first line contains a single integer *n* ($1 \le n \le 150$) — the number of statements in the programme.

Next *n* lines contain a statement each. Each statement contains exactly one operation (++ or --) and exactly one variable *x* (denoted as letter «X»). Thus, there are no empty statements. The operation and the variable can be written in any order.

Output

Print a single integer — the final value of *x*.

Examples

input

```
1
++X
```

output

```
1
```

input

```
2
X++
--X
```

output

```
0
```

【刘思瑞，元培物理方向，2023年秋】

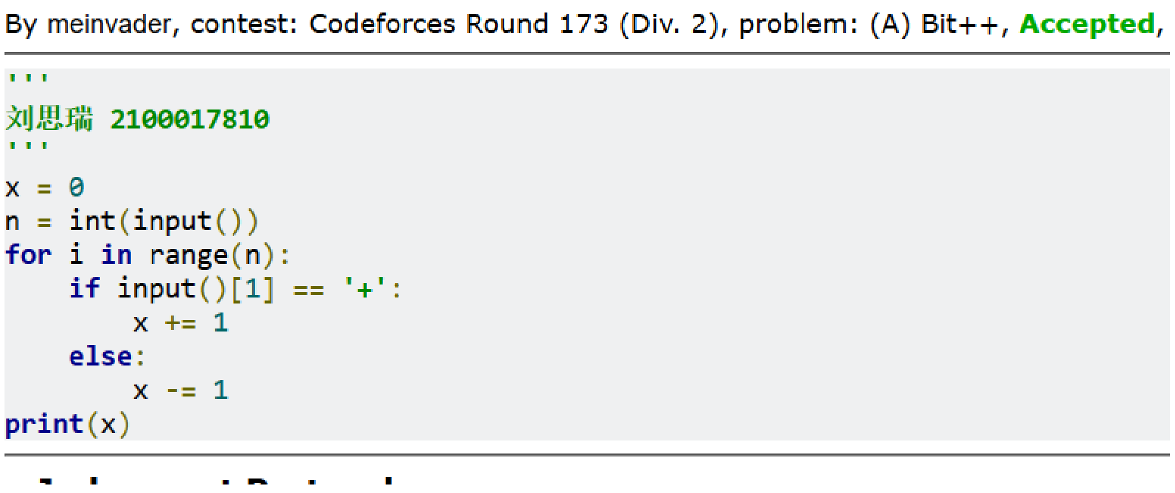
思路：

看第二位的字符最简便

代码

```python
'''
刘思瑞 2100017810
'''
x = 0
n = int(input())
for i in range(n):
    if input()[1] == '+':
        x += 1
    else:
        x -= 1
print(x)
```

代码运行截图


By meinvader, contest: Codeforces Round 173 (Div. 2), problem: (A) Bit++, Accepted,

```python
'''
刘思瑞 2100017810
'''
x = 0
n = int(input())
for i in range(n):
    if input()[1] == '+':
        x += 1
    else:
        x -= 1
print(x)
```

339A. Helpful Maths

greedy/implementation/sortings/strings, 800, http://codeforces.com/problemset/problem/339/A

Xenia the beginner mathematician is a third year student at elementary school. She is now learning the addition operation.

The teacher has written down the sum of multiple numbers. Pupils should calculate the sum. To make the calculation easier, the sum only contains numbers 1, 2 and 3. Still, that isn't enough for Xenia. She is only beginning to count, so she can calculate a sum only if the summands follow in non-decreasing order. For example, she can't calculate sum 1+3+2+1 but she can calculate sums 1+1+2 and 3+3.

You've got the sum that was written on the board. Rearrange the summans and print the sum in such a way that Xenia can calculate the sum.

Input

The first line contains a non-empty string *s* — the sum Xenia needs to count. String *s* contains no spaces. It only contains digits and characters "+". Besides, string *s* is a correct sum of numbers 1, 2 and 3. String *s* is at most 100 characters long.

Output

Print the new sum that Xenia can count.

Examples

input

```
3+2+1
```

output

```
1+2+3
```

input

```
1+1+3+1+3
```

output

```
1+1+1+3+3
```

input

```
2
```

output

```
2
```

【刘思瑞，元培物理方向，2023年秋】

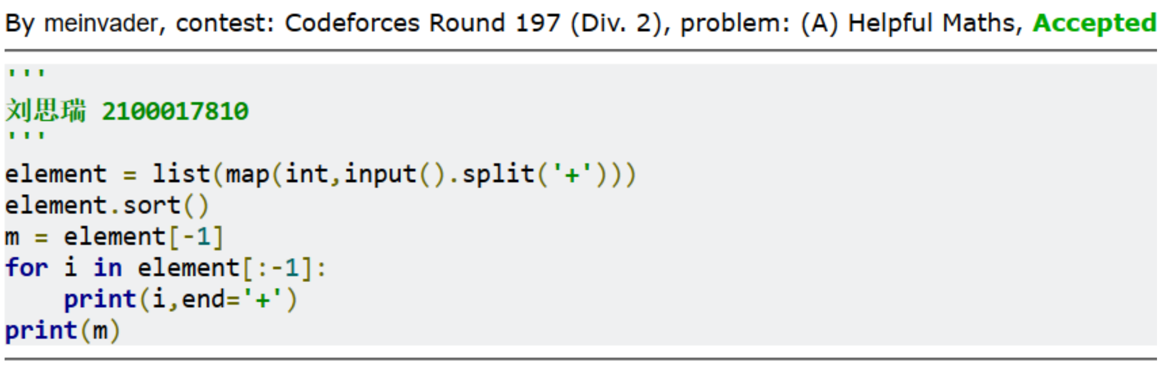
思路：

直接排序就好

代码

```python
'''
刘思瑞 2100017810
'''
element = list(map(int,input().split('+')))
element.sort()
m = element[-1]
for i in element[:-1]:
    print(i,end='+')
print(m)
```

代码运行截图



By meinvader, contest: Codeforces Round 197 (Div. 2), problem: (A) Helpful Maths, **Accepted**

```python
'''
刘思瑞 2100017810
'''
element = list(map(int,input().split('+')))
element.sort()
m = element[-1]
for i in element[:-1]:
    print(i,end='+')
print(m)
```

2. 选做题目

OJ01017: 装箱问题

greedy, http://cs101.openjudge.cn/practice/01017

一个工厂制造的产品形状都是长方体，它们的高度都是h，长和宽都相等，一共有六个型号，他们的长宽分别为1*1, 2*2, 3*3, 4*4, 5*5, 6*6。这些产品通常使用一个 6*6*h 的长方体包裹包装然后邮寄给客户。因为邮费很贵，所以工厂要想方设法的减小每个订单运送时的包裹数量。他们很需要有一个好的程序帮他们解决这个问题从而节省费用。现在这个程序由你来设计。

输入：输入文件包括几行，每一行代表一个订单。每个订单里的一行包括六个整数，中间用空格隔开，分别为1*1至6*6这六种产品的数量。输入文件将以6个0组成的一行结尾。

输出：除了输入的最后一行6个0以外，输入文件里每一行对应着输出文件的一行，每一行输出一个整数代表对应的订单所需的最小包裹数。

样例输入

```
0 0 4 0 0 1
7 5 1 0 0 0
0 0 0 0 0 0
```

样例输出

```
2
1
```

来源：Central Europe 1996

【刘思瑞，元培物理方向，2023年秋】

思路：

greedy 先把6全部算上，再看5能不能被1填满，再看4能不能被2填满，不能的话看1能不能补上剩下的空缺。接下来是最关键的3.先四个一组填满，然后分三种情况讨论，最后把2和1按相同的思路填满就好。

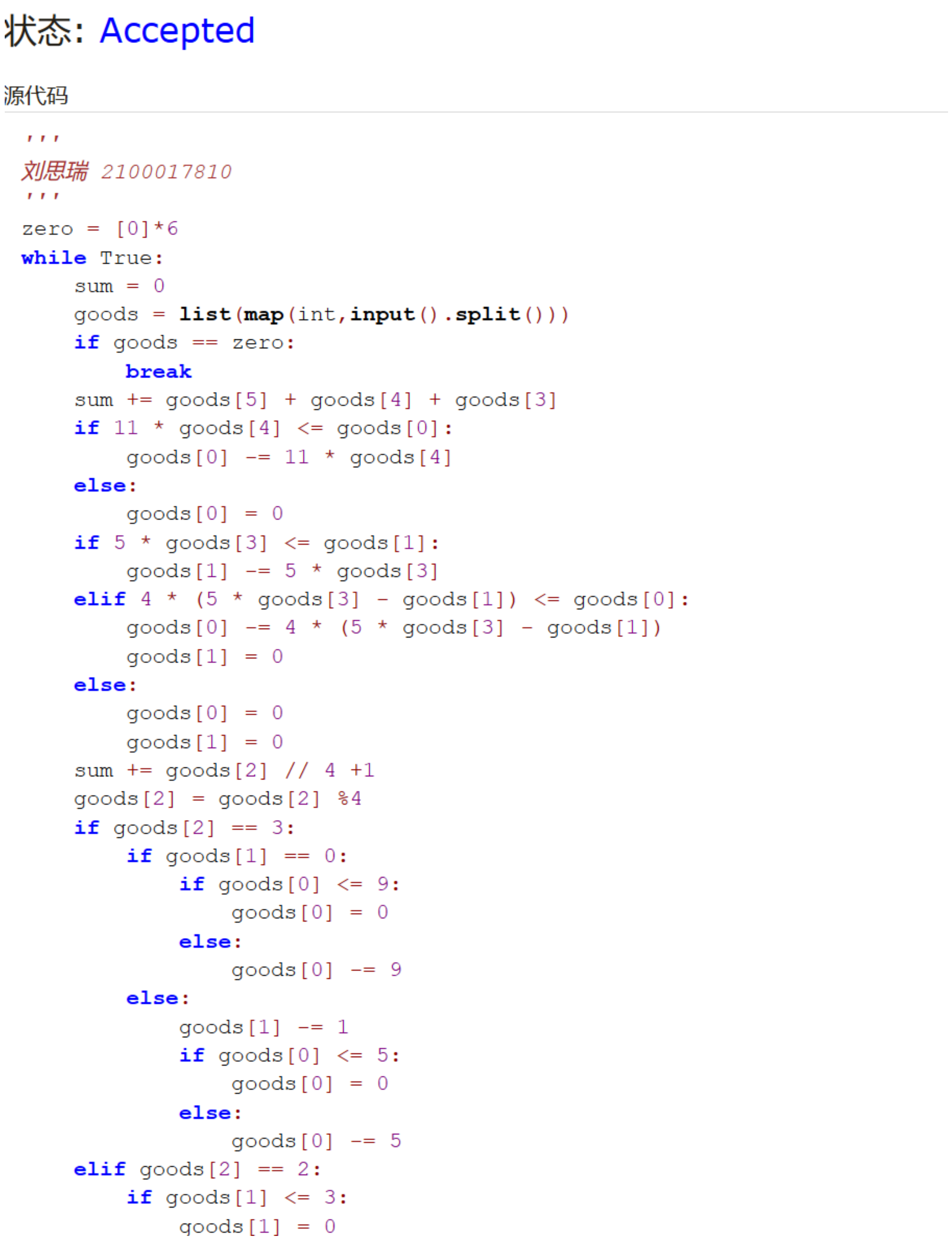
代码写的非常之丑陋，以至于我用了测试数据才找到bug，最后ac了。。。感觉这个题是不是有比较巧妙的方法啊

代码

```python
'''
刘思瑞 2100017810
'''
zero = [0]*6
while True:
    sum = 0
    goods = list(map(int,input().split()))
    if goods == zero:
            break
        sum += goods[5] + goods[4] + goods[3]
        if 11 * goods[4] <= goods[0]:
            goods[0] -= 11 * goods[4]
        else:
            goods[0] = 0
        if 5 * goods[3] <= goods[1]:
            goods[1] -= 5 * goods[3]
        elif 4 * (5 * goods[3] - goods[1]) <= goods[0]:
            goods[0] -= 4 * (5 * goods[3] - goods[1])
            goods[1] = 0
        else:
            goods[0] = 0
            goods[1] = 0
        sum += goods[2] // 4 +1
        goods[2] = goods[2] %4
        if goods[2] == 3:
            if goods[1] == 0:
                if goods[0] <= 9:
                    goods[0] = 0
                else:
                    goods[0] -= 9
            else:
                goods[1] -= 1
                if goods[0] <= 5:
                    goods[0] = 0
                else:
                    goods[0] -= 5
        elif goods[2] == 2:
            if goods[1] <= 3:
                goods[1] = 0
                if goods[0] <= 18 - 4 * goods[1]:
                    goods[0] = 0
                else:
                    goods[0] -= 18 - 4 * goods[1]
            else:
                goods[1] -= 3
                if goods[0] <= 6:
                    goods[0] = 0
                else:
                    goods[0] -= 6
        elif goods[2] == 1:
            if goods[1] <= 5:
                goods[1] = 0
                if goods[0] <= 27 - 4 * goods[1]:
                    goods[0] = 0
                else:
                    goods[0] -= 27 - 4 * goods[1]
            else:
                goods[1] -= 5
                if goods[0] <= 7:
                    goods[0] = 0
                else:
                    goods[0] -= 7
        else:
            sum -= 1
```

```python
    sum += goods[1]//9 +1
    goods[1] = goods[1] % 9
    if goods[1] == 0:
        sum += -(-goods[0]//36)
        sum -= 1
    if goods[0] > 36 - goods[1] * 4:
        sum += -(-(goods[0] - (36 - goods[1] * 4))//36)
    print(sum)
```

代码运行截图



状态: Accepted

源代码

```python
'''
刘思瑞 2100017810
'''
zero = [0]*6
while True:
    sum = 0
    goods = list(map(int,input().split()))
    if goods == zero:
        break
    sum += goods[5] + goods[4] + goods[3]
    if 11 * goods[4] <= goods[0]:
        goods[0] -= 11 * goods[4]
    else:
        goods[0] = 0
    if 5 * goods[3] <= goods[1]:
        goods[1] -= 5 * goods[3]
    elif 4 * (5 * goods[3] - goods[1]) <= goods[0]:
        goods[0] -= 4 * (5 * goods[3] - goods[1])
        goods[1] = 0
    else:
        goods[0] = 0
        goods[1] = 0
    sum += goods[2] // 4 +1
    goods[2] = goods[2] %4
    if goods[2] == 3:
        if goods[1] == 0:
            if goods[0] <= 9:
                goods[0] = 0
            else:
                goods[0] -= 9
        else:
            goods[1] -= 1
            if goods[0] <= 5:
                goods[0] = 0
            else:
                goods[0] -= 5
    elif goods[2] == 2:
        if goods[1] <= 3:
            goods[1] = 0
```

3. 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“练习”中每日推出的3个选做、CF、洛谷等网站题目。

作业题目相对较为简单，选做题更多的是麻烦而不是完全无法解决。

这星期练习了oj上每天发的一些题目，印象最深刻的是前缀中的周期，这个题目我因为考虑的不周到导致一直没有发现bug（）

最开始我还尝试了re表达式，但好像re比较占内存（（