题目：积分的换购

要求：

1、用户在购置网站上有消费积分m，最近网站推出积分换购的活动。换购商品的种类数是不确定的，如：“李宁双肩包”下架了，但又增加了“香皂”、“牙刷”等。

2、每种商品换购的次数不限，如：换购“新款毛巾”10条、12条、100条，都可以。

3、用户积分换购后几乎都会有剩余，找出换购后积分损失最小的前5种组合。

4、商品清单在文件“商品清单.txt”中。

5、结果存入文本文件“换购清单.txt”中。

商品名称 消耗积分

----------------------------------------------------------------

李宁双肩包 8200

新款毛巾 600

金吉星软包抽纸 1040

2018洗车大毛巾 1300

小米移动电源2 7900

不锈钢保温杯 8690

滋润型护肤脂 720

真好芦荟胶 1900

维达手帕纸 2090

佳洁士 390

百雀羚 720

----------------------------------------------------------------

1. 用户积分：5500。

现取4种商品：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 名称 | 单价 | 循环变量 | 最小数量 | 最大数量 |
| 0 | 李宁双肩包 | 8200 | a0 | 0  (0表示该商品不买) | 0  (=5500/8200) |
| 1 | 新款毛巾 | 600 | a1 | 0 | 9  (=5500/600) |
| 2 | 金吉星软包抽纸 | 1040 | a2 | 0 | 5  (=5500/1040) |
| 3 | 洗车大毛巾 | 1300 | a3 | 0 | 4  (=5500/1300) |

基本算法：循环嵌套。

for a0=0～0

for a1=0～9

for a2=0～5

for a3=0～4

sum\_jf=8200\*a0+600\*a1+1040\*a2+1300\*a3

if sum\_jf<=用户积分:

排序

1. 实际商品数量是不确定的、动态的，用普通变量a1、a2、a3、a4、a5、a5…等作为循环变量，变量到底有多少个？无法编写。
2. 可用数组int a[1000]代替普通变量来循环嵌套：a[0]、a[1]、a[2]、a[3]、a[4]、a[5]…,表示1000种商品。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 循环变量（老） | 循环变量（新） |
| 0 | 李宁双肩包 | a0 | a[0] |
| 1 | 新款毛巾 | a1 | a[1] |
| 2 | 金吉星软包抽纸 | a2 | a[2] |
| 3 | 洗车大毛巾 | a3 | a[3] |

for a[0]=0～0

for a[1]=0～9

for a[2]=0～5

for a[3]=0～4 #这是边界

sum\_jf=8200\*a[0] + 600\*a[1] + 1040\*a[2] + 1300\*a[3]

if sum\_jf<=用户积分:

排序

1. 延伸定义一：
2. 定义int a[100][2]，表示最多100种商品。

a[50][0]表示第50号，买的当前数，就是循环变量。

a[50][1]表示第50号，能买的最大数，循环变量的结束值，等于“用户积分/50号商品单价”。

1. 定义char mc[100][40]，表示100种商品的名称。
2. 定义float dj[100]，表示100种商品的单价。
3. 延伸定义二：

商品的名称、单价、当前数量、最大数量，是一个整体，可以整合存储，可以定义成列表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 商品编号(下标) | 名称/0列 | 单价/1列 | 当前数量/2列 | 最大数量/3列 |
| 0 | 李宁双肩包 | 8200 |  | 0 |
| 1 | 新款毛巾 | 600 |  | 9 |
| 2 | 金吉星软包抽纸 | 1040 |  | 5 |
| 3 | 洗车大毛巾 | 1300 |  | 4 |

分别是：名称、单价、当前数量、最大数量。

[

['李宁双肩包', 8200, 0, 0],

['新款毛巾', 600, 0, 9],

['金吉星软包抽纸', 1040, 0, 5],

['洗车大毛巾', 1250, 0, 4]

]

1. 用数组（列表）解决了商品数量动态问题。但for指令的个数就是商品个数，也是动态的，怎么编写？

上述for代码只能解决4种商品的问题，只需4个循环即可。但实际上，商品个数是不确定的，循环嵌套个数也必然确定不了,显然，上述的代码，解决不了问题。怎么办？可以用递归解决。

1. 递归
2. 参数

xh是商品的序号

1. 边界：

条件：xh是否小于“最大商品数”

1. 通项公式

没有到达边界前，要继续下一个商品的循环，就是a[xh+1]的循环。

1. 汇总积分消耗：到达边界后，才能计算、排序。

**def** deal\_sp(xh): #xh商品序号

**if** xh<最大商品数:

**for** a[xh+1][0]=0～a[xh+1][1]:

deal\_sp(xh+1)

**else**:

sum\_sp()

**return**

1. 换购组合的如何存储？

假设有100种商品，每个商品都可以换10个，就需要100个循环嵌套，总10100 种组合，内存开销、速度开销太大，无法存储，更无法排序。

1. 只定义存储5个商品的空间
2. 找到最小积分及其下标
3. 当前方案是否大于最小积分，如果大于就替换该元素。再“找到最小积分及其下标”。

如下图所示：

5000

|  |
| --- |
| 5200 |
| 5100 |
| 5060 |
| 4900 |
| 4800 |

当前最小积分4800，5000比4800大，用5000替换4800，得下表。

|  |
| --- |
| 5200 |
| 5100 |
| 5060 |
| 4900 |
| 5000 |

当前最小积分4900，5150比4900大，用5150替换4900，得下表。

5150

4000

|  |
| --- |
| 5200 |
| 5100 |
| 5060 |
| 5150 |
| 5000 |

当前最小积分5000，4000比5000小，此4000积分无用，不替换。

完整的替换步骤见文件“积分换算 换购组合.xlsx”。