

## 0/1 背包 (package.cpp)

总时间限制: 1s

内存限制: 64MB

### 【问题描述】

一个旅行者有一个最多能用  $m$  公斤的背包，现在有  $n$  件物品，它们的重量分别是  $W_1, W_2, \dots, W_n$ ，它们的价值分别为  $C_1, C_2, \dots, C_n$ ，若每种物品只有一件求旅行者能获得最大总价值。

### 【输入格式】

第一行：两个整数， $M$ (背包容量， $M \leq 200$ )和  $N$ (物品数量， $N \leq 30$ )；

第  $2..N+1$  行：每行二个整数  $W_i, C_i$ ，表示每个物品的重量和价值。

### 【输出格式】

仅一行，一个数，表示最大总价值。

### 【输入样例】(package.in)

```
10 4
2 1
3 3
4 5
7 9
```

### 【输出样例】(package.out)

```
12
```

---

## 完全背包问题 (knapsack.cpp)

总时间限制: 1s

内存限制: 64MB

### 【问题描述】

设有  $N$  种物品，每种物品有一个重量及一个价值。但每种物品的数量是无限的，同时有一个背包，最大载重量为  $M$ ，今从  $N$  种物品中选取若干件(同一种物品可以多次选取)，使其重量的和小于等于  $M$ ，而价值的和为最大。

### 【输入格式】

第一行：两个整数， $M$ (背包容量， $M \leq 200$ )和  $N$ (物品数量， $N \leq 30$ )；

第  $2..N+1$  行：每行二个整数  $W_i, C_i$ ，表示每个物品的重量和价值。

### 【输出格式】

仅一行，一个数，表示最大总价值。

### 【输入样例】(knapsack.in)

```
10 4
2 1
3 3
4 5
7 9
```

### 【输出样例】(knapsack.out)

```
max=12
```

## 多重背包问题---庆功会 (party.cpp)

总时间限制: 1s

内存限制: 64MB

### 【问题描述】

为了庆贺班级在校运动会上取得全校第一名成绩，班主任决定开一场庆功会，为此拨款购买奖品犒劳运动员。期望拨款金额能购买最大价值的奖品，可以补充他们的精力和体力。

### 【输入格式】

第一行二个数  $n(n \leq 500)$ ， $m(m \leq 6000)$ ，其中  $n$  代表希望购买的奖品的种数， $m$  表示拨款金额。

接下来  $n$  行，每行 3 个数， $v$ 、 $w$ 、 $s$ ，分别表示第  $i$  种奖品的价格、价值（价格与价值是不同的概念）和购买的数量（买 0 件到  $s$  件均可），其中  $v \leq 100$ ， $w \leq 1000$ ， $s \leq 10$ 。

### 【输出格式】

一个数，表示此次购买能获得的最大的价值（注意：不是价格！）。

### 【输入样例】(party.in)

```
5 1000
80 20 4
40 50 9
30 50 7
40 30 6
20 20 1
```

### 【输出样例】(party.out)

```
1040
```

---

## 混合背包 (mix.cpp)

总时间限制: 1s

内存限制: 64MB

### 【问题描述】

一个旅行者有一个最多能用  $V$  公斤的背包，现在有  $n$  件物品，它们的重量分别是  $W_1, W_2, \dots, W_n$ ，它们的价值分别为  $C_1, C_2, \dots, C_n$ 。有的物品只可以取一次（01 背包），有的物品可以取无限次（完全背包），有的物品可以取的次数有一个上限（多重背包）。求解将哪些物品装入背包可使这些物品的费用总和不超过背包容量，且价值总和最大。

### 【输入格式】

第 1 行：二个整数， $V$ (背包容量， $V \leq 200$ )， $N$ (物品数量， $N \leq 30$ )；

第 2.. $N+1$  行：每行三个整数  $W_i, C_i, P_i$ ，前两个整数分别表示每个物品的重量，价值，第三个整数若为 0，则说明此物品可以购买无数件，若为其他数字，则为此物品可购买的最多件数( $P_i$ )。

### 【输出格式】

仅一行，一个数，表示最大总价值。

### 【输入样例】(mix.in)

```
10 4
2 1 0
3 3 1
4 5 4
```

### 【输出样例】(mix.out)

```
11
```

## 二维费用的背包问题 (gas.cpp)

总时间限制：1s

内存限制：64MB

### 【问题描述】

潜水员为了潜水要使用特殊的装备。他有一个带 2 种气体的气缸：一个为氧气，一个为氮气。让潜水员下潜的深度需要各种的数量的氧和氮。潜水员有一定数量的气缸。每个气缸都有重量和气体容量。潜水员为了完成他的工作需要特定数量的氧和氮。他完成工作所需气缸的总重的最低限度的是多少？

例如：潜水员有 5 个气缸。每行三个数字为：氧，氮的（升）量和气缸的重量：

```
3 36 120
10 25 129
5 50 250
1 45 130
4 20 119
```

如果潜水员需要 5 升的氧和 60 升的氮则总重最小为 249（1，2 或者 4，5 号气缸）。

你的任务就是计算潜水员为了完成他的工作需要的气缸的权重的最低值。

### 【输入格式】

第一行有 2 整数  $m, n$  ( $1 \leq m \leq 21, 1 \leq n \leq 79$ )。它们表示氧，氮各自需要的量。

第二行为整数  $k$  ( $1 \leq k \leq 1000$ ) 表示气缸的个数。

此后的  $k$  行，每行包括  $a_i, b_i, c_i$  ( $1 \leq a_i \leq 21, 1 \leq b_i \leq 79, 1 \leq c_i \leq 800$ ) 3 整数。这些各自是：第  $i$  个气缸里的氧和氮的容量及汽缸重量。

### 【输出格式】

仅一行包含一个整数，为潜水员完成工作所需的气缸的重量总和的最低值。

### 【输入样例】(gas.in)

```
5 60
5
3 36 120
10 25 129
5 50 250
1 45 130
4 20 119
```

### 【输出样例】(gas.out)

```
249
```

## 分组背包 (group.cpp)

总时间限制: 1s      内存限制: 64MB

### 【问题描述】

一个旅行者有一个最多能用  $V$  公斤的背包，现在有  $n$  件物品，它们的重量分别是  $W_1, W_2, \dots, W_n$ ，它们的价值分别为  $C_1, C_2, \dots, C_n$ 。这些物品被划分为若干组，每组中的物品互相冲突，最多选一件。求解将哪些物品装入背包可使这些物品的费用总和不超过背包容量，且价值总和最大。

### 【输入格式】

第 1 行: 三个整数,  $V$ (背包容量,  $V \leq 200$ ),  $N$ (物品数量,  $N \leq 30$ )和  $T$ (最大组号,  $T \leq 10$ );

第 2.. $N+1$  行: 每行三个整数  $W_i, C_i, P$ , 表示每个物品的重量, 价值, 所属组号。

### 【输出格式】

仅一行, 一个数, 表示最大总价值。

### 【输入样例】(group.in)

```
10 6 3
2 1 1
3 3 1
4 8 2
6 9 2
2 8 3
3 9 3
```

### 【输出样例】(group.out)

```
20
```

---

## 背包问题的方案总数---货币系统 (money.cpp)

总时间限制: 1s      内存限制: 64MB

### 【问题描述】

给你一个  $n$  种面值的货币系统，求组成面值为  $m$  的货币有多少种方案。

### 【输入格式】

第 1 行: 二个整数,  $n$  和  $m$ ;

第 2.. $N+1$  行: 每行 1 个整数, 表示每种货币面值。

### 【输出格式】

仅一行, 一个数, 表示方案数。

### 【输入样例】(money.in)

```
3 10
1
2
5
```

### 【输出样例】(money.out)

```
10
```

## 砝码称重 (weight.cpp)

总时间限制: 1s      内存限制: 64MB

### 【问题描述】

设有 1g, 2g, 3g, 5g, 10g, 20g 的砝码各若干枚（其总重 $\leq 1000$ g）。

### 【输入格式】

a1, a2, a3, a4, a5, a6 (表示 1g 砝码有 a1 个, 2g 砝码有 a2 个, ..., 20g 砝码有 a6 个)。

### 【输出格式】

Total=N (N 表示用这些砝码能称出的不同重量的个数, 但不包括一个砝码也不用的情况)。

### 【输入样例】(weight.in)

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

### 【输出样例】(weight.out)

```
Total=3
```

```
//表示可以称出 1g, 2g, 3g 三种不同的重量
```

### 【样例说明】

```
Total=3                      //表示可以称出 1g, 2g, 3g 三种不同的重量
```

---

## 开心的金明 (happy.cpp)

总时间限制: 1s      内存限制: 64MB

### 【问题描述】

金明今天很开心, 家里购置的新房就要领钥匙了, 新房里有一间他自己专用的很宽敞的房间。更让他高兴的是, 妈妈昨天对他说: “你的房间需要购买哪些物品, 怎么布置, 你说了算, 只要不超过 N 元钱就行”。今天一早金明就开始做预算, 但是他想买的东西太多了, 肯定会超过妈妈限定的 N 元。于是, 他把每件物品规定了一个重要度, 分为 5 等: 用整数 1~5 表示, 第 5 等最重要。他还从因特网上查到了每件物品的价格 (都是整数元)。他希望在不超过 N 元 (可以等于 N 元) 的前提下, 使每件物品的价格与重要度的乘积的总和最大。

设第 j 件物品的价格为  $v[j]$ , 重要度为  $w[j]$ , 共选中了 k 件物品, 编号依次为  $j_1, j_2, \dots, j_k$ , 则所求的总和为:

$v[j_1] * w[j_1] + v[j_2] * w[j_2] + \dots + v[j_k] * w[j_k]$ 。(其中 \* 为乘号)

请你帮助金明设计一个满足要求的购物单。

### 【输入格式】

输入文件 happy.in 的第 1 行, 为两个正整数, 用一个空格隔开:

N M (其中  $N < 30000$ , 表示总钱数;  $M < 25$ , 为希望购买物品的个数。)

从第 2 行到第 m+1 行, 第 j 行给出了编号为 j-1 的物品的基本数据, 每行有 2 个非负整数。

V P (其中 V 表示该物品的价格 ( $V \leq 10000$ ), P 表示该物品的重要度 ( $1 \leq P \leq 5$ ))。

### 【输出格式】

输出文件 happy.out 只有一个正整数, 为不超过总钱数的物品的价格与重要度乘积的总和的最大值 ( $< 100000000$ )。

### 【输入样例】(happy.in)

```
1000 5
800 2
400 5
300 5
400 3
200 2
```

### 【输出样例】(happy.out)

```
3900
```

## 装箱问题(boxes.cpp)

总时间限制: 1s

内存限制: 64MB

### 【问题描述】

有一个箱子容量为  $v$  (正整数,  $0 \leq v \leq 20000$ ), 同时有  $n$  个物品 ( $0 < n \leq 30$ ), 每个物品有一个体积 (正整数)。要求从  $n$  个物品中, 任取若干个装入箱内, 使箱子的剩余空间为最小。

### 【输入格式】

第一行是箱子的容量  $v$ 。

第二行是物品数  $n$ 。

接下来  $n$  行, 分别表示这  $n$  个物品的体积。

### 【输出格式】

箱子剩余空间。

### 【输入样例】(boxes.in)

```
24
6
8
3
12
7
9
7
```

### 【输出样例】(boxes.out)

```
0
```

---

## 采药(medic.cpp)

总时间限制: 1s

内存限制: 64MB

### 【问题描述】

辰辰是个天资聪颖的孩子, 他的梦想是成为世界上最伟大的医师。为此, 他想拜附近最有威望的医师为师。医师为了判断他的资质, 给他出了一个难题。医师把他带到一个到处都是草药的山洞里对他说: “孩子, 这个山洞里有一些不同的草药, 采每一株都需要一些时间, 每一株也有它自身的价值。我会给你一段时间, 在这段时间里, 你可以采到一些草药。如果你是一个聪明的孩子, 你应该可以让采到的草药的总价值最大。”

如果你是辰辰, 你能完成这个任务吗?

### 【输入格式】

输入文件 medic.in 的第一行有两个整数  $T$  ( $1 \leq T \leq 1000$ ) 和  $M$  ( $1 \leq M \leq 100$ ), 用一个空格隔开,  $T$  代表总共能够用来采药的时间,  $M$  代表山洞里的草药的数目。

接下来的  $M$  行每行包括两个在 1 到 100 之间 (包括 1 和 100) 的整数, 分别表示采摘某株草药的时间和这株草药的价值。

### 【输出格式】

输出文件 medic.out 包括一行, 这一行只包含一个整数, 表示在规定的时间内, 可以采到的草药的最大总价值。

### 【输入样例】(medic.in)

```
70 3
71 100
69 1
1 2
```

### 【输出样例】(medic.out)

```
3
```

### 【数据规模】

对于 30% 的数据,  $M \leq 10$ ;

对于 100% 的数据,  $M \leq 100$ 。

## 竞赛总分 (inflate.cpp)

总时间限制: 1s      内存限制: 64MB

### 【问题描述】

学生在我们 USACO 的竞赛中的得分越多我们越高兴。我们试着设计我们的竞赛以便人们能尽可能多得分。

现在要进行一次竞赛，总时间  $T$  固定，有若干类型可选择的题目，每种类型题目可选入的数量不限，每种类型题目有一个  $si$ （解答此题所得的分数）和  $ti$ （解答此题所需的时间），现要选择若干题目，使解这些题的总时间在  $T$  以内的前提下，所得的总分最大。

输入包括竞赛的时间  $M(1 \leq M \leq 10000)$  和题目类型数目  $N(1 \leq N \leq 10000)$ 。

后面的每一行将包括两个整数来描述一种"题型"：

第一个整数说明解决这种题目能得的分数( $1 \leq \text{points} \leq 10000$ )，第二整数说明解决这种题目所需的时间( $1 \leq \text{minutes} \leq 10000$ )。

### 【输入格式】

第 1 行：两个整数：竞赛的时间  $M$  和题目类型数目  $N$ 。

第  $2-N+1$  行：两个整数：每种类型题目的分数和耗时。

### 【输出格式】

单独的一行，在给定固定时间里得到的最大的分数。

### 【输入样例】(inflate.in)

```
300 4
100 60
250 120
120 100
35 20
```

### 【输出样例】(inflate.out)

```
605
```

### 【样例解释】

605 表示从第 2 种类型中选两题和第 4 种类型中选三题

---

## 最小乘车费用(busses.cpp)

总时间限制: 1s      内存限制: 64MB

### 【问题描述】

某条街上每隔一公里就有一汽车站，乘车费用如下表：

公里数	1	2	3	4	5	6	7	8	9	10
费用	12	21	31	40	49	58	69	79	90	101

而一辆汽车从不行驶超过 10 公里。某人想行驶  $n$  公里，假设他可以任意次换车，请你帮他找到一种乘车方案使费用最小（10 公里的费用比 1 公里小的情况是允许的）。

### 【输入格式】

输入文件共两行，第一行为 10 个不超过 100 的整数，依次表示行驶 1~10 公里的费用，相邻两数间用空格隔开；第二行为某人想要行驶的公里数。

### 【输出格式】

输出文件仅一行包含一个整数，表示该测试点的最小费用。

### 【输入样例】(busses.in)

```
12 21 31 40 49 58 69 79 90 101
15
```

### 【输出样例】(busses.out)

```
147
```