

c. 炒鸡矿工

(chicken.cpp/c/pas, 1s, 256MB)

【问题背景】

炒鸡矿工不是黄金矿工，而是dst在QQ小游戏中看到的一个沙雕小游戏。



【问题描述】

这个游戏的玩法建立在一个挖矿系统上。

我们认为游戏从第0分钟开始，每过 p 分钟，炒鸡矿工可以完成一次挖矿，每次可以挖重量为 c 的金矿，准确的说，在一次挖矿中，dst会在第 p 分钟末收获重量为 c 的金矿。炒鸡矿工在开局后会不断地重复挖矿操作，不能休息。

金矿可以储存或用于升级挖矿系统。

开局时，挖矿系统的等级为1级。挖矿系统最多升到 $n + 1$ 级。升级操作不消耗时间，但只能在一次挖矿开始前进行。每次升级会从第 i 级升级到第 $i + 1$ 级($1 \leq i \leq n$)，需要花费重量为 w_i 的金矿，可以使每次挖矿的重量增加 v_i ，使每次挖矿的时间变成 s_i 。由于升级不消耗时间，dst可以在一瞬间多次升级。

开局时，dst拥有重量为 m 的金矿。他想知道，在开局后恰好 t 分钟时，他最多能拥有的金矿重量 g 是多少。

【输入】

输入共4行。

第1行包含5个非负整数 p, c, n, m, t 。

第2行包含 n 个非负整数，第 i 个数表示 w_i 。

第3行包含 n 个非负整数，第 i 个数表示 v_i 。

第4行包含 n 个正整数，第 i 个数表示 s_i 。

若 $n = 0$ ，则第2行，第3行，第4行为空行。

【输出】

输出共1行，包含1个非负整数 g 。

【输入输出样例】

chicken.in	chicken.out
3 2 2 1 6	17

1 3	
3 0	
3 1	

【输入输出样例说明】

下面给出一种可行的方案（同一行内相同颜色标记表示相关联的变化）：

	等级	收矿倒计时	拥有金矿重量
开局	1		1
0s	$1 + 1 = 2$	3	$1 - 1 = 0$
1s	2	2	0
2s	2	1	0
3s	$2 + 1 = 3$	$0 + 1 = 1$	$0 + 5 - 3 = 2$
4s	3	$0 + 1 = 1$	$2 + 5 = 7$
5s	3	$0 + 1 = 1$	$7 + 5 = 12$
6s	3	0	$12 + 5 = 17$

【数据规模与约定】

对于5%的数据， $n = 0$ 。

另外10%的数据， $w_i = 0$ 。

另外10%的数据， $v_i = 0$ 。

另外10%的数据， $s_i = p$ 。

另外10%的数据， $t \leq 100$ 。

另外10%的数据， $m, c, w_i, v_i \leq 10$ 。

对于80%的数据， $p, s_i \leq 10$ 。

对于100%的数据， $1 \leq p, s_i, t \leq 10^3; 0 \leq n \leq 10^3; 0 \leq m, c, w_i, v_i \leq 10^9$ 。