

## Großeinkauf

Nachdem du befürchtest, dass aus deiner Informatiker-Karriere wohl nichts wird, weil die AOI-Aufgaben so schwierig sind, möchtest du stattdessen einen Supermarkt aufmachen. Daher willst du nun im Großhandel jene Produkte günstig einkaufen (zum **Einkaufspreis**), die später in deinem Geschäft zu erwerben sein werden (zum **Verkaufspreis**). Wie es in Österreich so üblich ist, bietet der Großhandel eine Vielzahl von Rabattpickerln an, welche jeweils auf genau 1 Produkt anwendbar sind. Da dein Budget nur beschränkt ist, möchtest du mit dem, was du hast, Produkte einkaufen und die Rabattpickerl dabei so geschickt anwenden, dass die Summe des Verkaufspreises jener Waren, die du gekauft hast, möglichst hoch ist.



Rabattpickerl-Chaos in Österreich

Konkret läuft das so ab: Im Sortiment des Großhandels gibt es  $P$  verschiedene Produktarten. Jedes Produkt hat einen Einkaufspreis  $e_p$  in Cent (so viel musst du dafür zahlen), einen Verkaufspreis  $v_p$  in Cent (um so viel kannst du es weiterverkaufen) und es sind  $a_p$  Stück auf Lager (du kannst maximal so viel davon einkaufen). Außerdem gibt es  $R$  Rabattpickerl, wobei jedes einen Rabatt von  $p_r$  ( $p_r$  ist eine Ganzzahl mit  $1 \leq p_r \leq 100$ ) Prozent auf genau 1 Stück eines beliebigen Produkts gewährt. Pro Stück kann maximal 1 Pickerl eingelöst werden. Der Preisnachlass funktioniert wie gewohnt: Kostet ein Produkt beispielsweise 49 Cent und wird ein Rabatt von 30% darauf angewandt, werden einfach 30% von den 49 Cent abgezogen, was 34,3 Cent ergeben würde – dieser Betrag wird dann **immer abgerundet**, falls er nicht ganzzahlig ist – in unserem Fall ergibt der Endpreis also 34 Cent.

Du möchtest nun jene Produkte aus dem Lagerbestand des Großhandels aussuchen und dich jeweils dafür entscheiden, wie viele du davon kaufst, sodass der Gesamtverkaufspreis  $G$  (also die Summe der Verkaufspreise  $v_p$  der eingekauften Produkte) so hoch wie möglich ist – ohne dein Anfangsbudget  $B$  (in Cent gegeben) zu überschreiten. Natürlich kannst du von einer Produktsorte nur maximal so viel Stück kaufen, wie auf Lager sind – also  $a_p$ . Dabei kannst du die verfügbaren Pickerl verwenden, um den Einkaufspreis je eines Stücks zu reduzieren.

## Eingabe

Die erste Zeile des Inputs enthält 3 natürliche Zahlen  $P$ ,  $B$  und  $R$  – die Anzahl der Produkttypen im Sortiment, dein Startbudget in Cent und die Anzahl der Rabattpickerl. Danach folgen  $P$  Zeilen mit je 3 natürlichen Zahlen:  $e_p$ ,  $v_p$  und  $a_p$  – der Einkaufspreis, Verkaufspreis und die lagernde Stückzahl des  $p$ -ten Produkts. Danach folgt eine Zeile mit  $R$  natürlichen Zahlen  $p_r$  – der Betrag des Rabatts des  $r$ -ten Pickerls in Prozent.

## Ausgabe

Gib nur eine Zahl aus – den optimalen Gesamtverkaufswert, wie oben beschrieben.

## Beispiele

Eingabe	Ausgabe	Anmerkungen
4 11 1 5 12 1 7 8 1 2 10 1 4 6 1 50	30	Entspricht den Constraints von Subtask 4.  Um den optimalen Gesamtwert von 30 zu erhalten, können beispielsweise die ersten 3 Produkte gekauft und das -50% Pickerl auf das erste angewandt werden.

Eingabe	Ausgabe	Anmerkungen
4 20 2 5 12 3 7 8 1 2 10 2 4 6 5 25 27	62	Entspricht den Constraints von Subtask 7.  Um den optimalen Gesamtwert von 62 zu erhalten, können beispielsweise 3 Stück des ersten Produkts gekauft werden, worauf auf eines das -25% Rabattpickerl angewandt wird, dazu 2 Stück des dritten Produkts und 1 Stück des vierten Produkts, auf welches das -27% Pickerl angewandt wird.

## Subtasks

Allgemein gilt:

- $1 \leq P \leq 200$
- $0 \leq B \leq 500$
- $0 \leq R \leq 50$
- $1 \leq a_p \leq 500$
- $1 \leq e_p, v_p \leq 1000$
- $1 \leq p_r \leq 100$

**Subtask 1 (1 Punkt):**  $B = R = 0$

**Subtask 2 (8 Punkte):**  $R = 0, P \leq 15, a_p = 1$

**Subtask 3 (17 Punkte):**  $R = 0, a_p = 1$

**Subtask 4 (10 Punkte):**  $R = 1, a_p = 1$

**Subtask 5 (15 Punkte):**  $R \leq 7, a_p = 1$

**Subtask 6 (19 Punkte):**  $a_p = 1$

**Subtask 7 (9 Punkte):**  $a_p \leq 10$

**Subtask 8 (21 Punkte):** Keine Einschränkungen

## Limits

**Zeitlimit:** 1 s

**Speicherlimit:** 256 MB