

# Pizzabonnen - categorie 4



## Opgave

Je wil voor een grote groep mensen pizza's bestellen maar je budget is eerder beperkt. Gelukkig heb je een reeks kortingsbonnen ter beschikking. Een bon is telkens van de vorm “ $a$  pizza's kopen,  $b$  gratis”, genoteerd als “een  $a + b$  bon”. Je wil nu uitrekenen welke bonnen je moet toepassen op welke pizza's om zo weinig mogelijk te betalen voor de bestelling.

Beschouw het volgende voorbeeld:

$$\underbrace{25\text{€} \quad 12\text{€} \quad 17\text{€}}_{2+1} \quad \underbrace{9\text{€} \quad 13\text{€}}_{1+1}$$

Hier worden 5 pizza's besteld en heb je twee bonnen ter beschikking: de 2+1 bon pas je toe op de pizza's met prijzen 25€, 12€ en 17€. De gratis pizza's zijn altijd de goedkoopste uit de groep, wat betekent dat je voor deze drie pizza's  $25 + 17 = 42\text{€}$  betaalt en je 12€ korting hebt gekregen. Hetzelfde geldt voor de laatste twee pizza's: je betaalt voor de pizza van 13€ en je krijgt die van 9€ gratis. In totaal betaal je dus  $42 + 13 = 55\text{€}$ .

Het is echter goedkoper om de bonnen anders te gebruiken:

$$\underbrace{13\text{€} \quad 25\text{€}}_{1+1} \quad \underbrace{17\text{€} \quad 12\text{€} \quad 9\text{€}}_{2+1}$$

Hier betaal je slechts  $25 + (17 + 12) = 54\text{€}$ .

Het doel van deze opgave is dus om gegeven een bestelling (voorgesteld door een lijst pizzaprijzen) en een reeks kortingsbonnen (voorgesteld door een reeks  $a + b$  koppels) de optimale pizza's/bon-toekenning te vinden die leidt tot de minimale prijs.

- Het is mogelijk dat je te weinig kortingsbonnen hebt; in dit geval moet je de volle prijs betalen voor de overblijvende pizza's.
- Het kan tevens voorkomen dat je bonnen te veel hebt: je hoeft ze dan niet allemaal te gebruiken.
- Om een  $a + b$  bon te kunnen gebruiken, moeten er ook  $a + b$  pizza's besteld worden. Het is toegelaten om extra pizza's te bestellen om een kortingsbon te kunnen gebruiken. Bijvoorbeeld, beschouw een bestelling van twee pizza's van 15€ en 20€, met als enige kortingsbon een 1 + 2 bon. Zonder de bon te gebruiken betaal je 35€, maar door een extra pizza bij te kopen, kan je de bon gebruiken en hoef je slechts 20€ te betalen.

## Invoer

De eerste regel van het invoerbestand bevat het aantal testgevallen. Elk testgeval volgt het volgende patroon:

$$\begin{array}{l} n \ p_1 \ p_2 \ \dots \ p_n \\ m \\ a_1 \ b_1 \\ a_2 \ b_2 \\ \vdots \\ a_m \ b_m \end{array}$$

Het gaat hier telkens om door spaties gescheiden natuurlijke getallen. De eerste lijn stelt de bestelling voor:  $n \in [1 \dots 1000]$  is het aantal te bestellen pizza's, en de daarop volgende  $p_i \in [1 \dots 10000]$  stellen de prijzen voor. De tweede lijn bevat  $m \in [1 \dots 100]$ , het aantal kortingsbonnen. De hierop volgende lijnen bevatten telkens twee natuurlijke getallen  $a_i$  en  $b_i \in [0 \dots 20]$ , die elk een  $a_i + b_i$  kortingsbon voorstellen.

**Belangrijk:** het aantal te bestellen pizza's kan hoog oplopen, evenals het aantal kortingsbonnen. Wel is het zo dat er weinig *verschillende soorten* kortingsbonnen zullen zijn.

## Uitvoer

Per testgeval moet je één lijn uitvoeren:

$$idx \ p$$

waarbij  $idx$  het volgnummer is van het testgeval (beginnen tellen bij 1) en  $p$  de laagst mogelijke kost is voor de bestelling.

**Let op!** Zorg ervoor dat je uitvoer geen overbodige tekens bevat, bijvoorbeeld een spatie op het einde van een regel of een lege regel op het einde van de uitvoer. Dat zorgt er immers voor dat je uitvoer als foutief wordt beschouwd.

## Voorbeeld

### Invoer

```
2
1 5
1
1 0
4 10 5 15 20
2
1 1
1 1
```

### Uitvoer

```
1 5
2 30
```