

A Fierljeppers

Een fierljepper wil een mooie wandeling gaan maken. Het natuurlandschap in kwestie is een lange rechte weg, keurig opgedeeld in vakjes. Sommige vakjes zijn echter moeilijk begaanbaar, en dan komt zijn kunstje goed van pas. Hij heeft namelijk een stok bij zich, waarmee hij als hij wil een sprong kan maken. Hij slaat dan vakjes over. De fierljepper kan nooit de bocht om.

Per vakje is aangegeven hoeveel energie het kost om er naartoe te lopen. Als je naar een vakje springt zul je die kosten plus de sprongkosten moeten betalen, maar de vakjes waar je overheen springt uiteraard niet. Minder ver springen is door de aard van de sprong niet mogelijk. Onze held begint voor het eerste vakje, en kan dan meteen kiezen of hij gaat springen of lopen. Zijn wandeling eindigt als hij in het laatste vakje is aangekomen.

Probleem

Maak een programma dat de kosten van de goedkoopste route afdrukt. Gegeven zijn een sprongafstand, sprongkosten, lengte van de weg, kosten per vakje onderweg.

Invoer

Een regel, met daarop een enkele positieve integer: het aantal runs.

Vervolgens per run:

- Een regel met daarop 3 integers gescheiden door spaties.
De sprongafstand sa $1 \leq sa < 10^3$ $sa \in \mathbb{Z}$
De sprongkosten sk $1 \leq sk < 10^3$ $sk \in \mathbb{Z}$
De lengte van de weg len $1 \leq len < 10^4$ $len \in \mathbb{Z}$
- Een regel met daarop len integers gescheiden door spaties.
De kosten om een vakje binnen te lopen/springen k $1 \leq k < 10^3$ $k \in \mathbb{Z}$

Uitvoer

Per run een regel, met daarop een integer: de kosten van de goedkoopste route.

Voorbeeld invoer

```
| 2  
| 2 10 4  
| 1 100 1 1  
| 3 1 6  
| 12 12 12 12 12 12
```

Bijbehorende uitvoer

```
| 13  
| 26
```

B Konijntjes

Een wiskundige bioloog kweekt wiskundige konijntjes. Wiskundige konijntjes zijn heel bijzondere konijntjes. Ze hebben geen geslacht, paren nooit en zien er altijd hetzelfde uit. Maar toch krijgen ze, als ze volwassen zijn, ieder jaar één kind. Bovendien worden alle wiskundige konijntjes even oud en worden ze altijd op dezelfde leeftijd volwassen.

De bioloog is benieuwd hoeveel konijntjes hij volgend jaar zal hebben. Dit is voor hem lastig te bepalen, omdat hij niet weet hoe oud zijn konijntjes zijn. Hij weet dus ook niet hoeveel er volwassen zijn of hoeveel er volgend jaar dood gaan. Gelukkig heeft hij wel bijgehouden hoeveel konijntjes hij eerdere jaren had. En hij heeft in de encyclopedie opgezocht hoe oud wiskundige konijntjes worden, op welke leeftijd volwassen en hoelang ze volwassen blijven.

Probleem

Maak een programma dat kan extrapoleren hoeveel konijntjes er volgend jaar zullen zijn. Hierbij is gegeven hoe lang een konijntje jong, volwassen en oud is. Een volwassen konijntje krijgt aan het eind van het jaar een kind. Een oud konijntje krijgt geen kinderen meer.

Invoer

Een regel, met daarop een enkele positieve integer: het aantal runs.
Vervolgens per run:

- Een regel met daarop 4 integers, gescheiden door spaties.
Het aantal jaren j dat een konijntje jong is $0 \leq j \leq 6$
Het aantal jaren v dat een konijntje volwassen is $2 \leq v \leq 6$
Het aantal jaren o dat een konijntje oud is $0 \leq o \leq 6$
Het aantal jaren n dat de bioloog de populatie heeft bijgehouden $j + v + o \leq n \leq 24$
- Een regel met daarop n integers, gescheiden door spaties.
De populatiegroottes p $0 \leq p < 10^3$ $p \in \mathbb{Z}$

Gegeven is dat n zo groot is, dat de populatiegrootte in het volgende jaar uniek bepaald is.

Uitvoer

Per run een regel, met daarop een integer: het aantal konijntjes in het eerstvolgende jaar.

Voorbeeld invoer

```
| 3
| 0 2 0 6
| 2 3 5 8 13 21
| 0 1 0 4
| 1 1 1 1
| 0 1 2 5
| 1 2 3 3 3
```

Bijbehorende uitvoer

```
| 34
| 1
| 3
```

C Europakket

Vlak voor de invoering van de Euro kreeg iedereen een Euro-kit met van iedere munt er eentje. Ook kon je voor fl.25,- zakjes kopen met 32 euromunten ter waarde van €11,35 (\approx fl.25,01 hoera 1 cent winst). Ze zeiden er in de reclame alleen niet bij hoeveel je er van iedere munt kreeg. Dat was dus iets waar je fijn over kon speculeren. Maar hoe groot is de kans dat je het goed hebt als je raadt in welke verdeling de munten in het zakje zitten? Of anders geformuleerd: hoeveel mogelijkheden zijn er?

Probleem

Schrijf een programma dat gegeven een munten stelsel, een totale waarde en het aantal munten berekent op hoeveel manieren je een zakje kunt vullen.

Invoer

Een regel, met daarop een enkele positieve integer: het aantal runs.
Vervolgens per run:

- Een regel met daarop 3 integers gescheiden door spaties.
De totale waarde w $1 \leq w < 3000$ $w \in \mathbb{Z}$
Het totale aantal munten n $1 \leq n < 50$ $n \in \mathbb{Z}$
Het aantal munten in het stelsel m $1 \leq m < 20$ $m \in \mathbb{Z}$
- Een regel met daarop m verschillende integers gescheiden door spaties.
De waarden van de munten in het stelsel w $1 \leq w \leq 1000$ $w \in \mathbb{Z}$

Uitvoer

Per run een regel, met daarop het aantal manieren waarop het zakje geconstrueerd kan zijn.

Voorbeeld Invoer

```
| 2  
| 10 5 3  
| 1 2 5  
| 100 9 4  
| 1 2 5 10
```

Bijbehorende uitvoer

```
| 2  
| 0
```

D Vouwen

Neem een A4'tje en vouw het in 8-en, zoals de lijntjes in het eerste plaatje aangeven. Zet vervolgens ook de cijfers 1 tot en met 8 in de vakjes zoals aangegeven. Probeer het geheel nu zo op te vouwen, dat je van boven naar beneden precies 1 tot en met 8 hebt. Dit hoeft niet heel makkelijk te zijn, en mag best een beetje frummelwerk vergen, maar uiteindelijk is het mogelijk. De oplossing bevat geen nieuwe vouwen. Maar als je begint met de nummers zoals in het tweede plaatje kun je de nummers niet in volgorde op elkaar krijgen.

1	4	5	8
2	3	6	7

Figuur 1: Een voorbeeld dat wel kan

1	2	3	4
5	6	7	8

Figuur 2: Een voorbeeld dat niet kan

Probleem

Maak een programma dat gegeven een verdeling van de getallen over een A4'tje bepaalt of het mogelijk is de getallen in volgorde op elkaar te krijgen.

Invoer

Een regel, met daarop een enkele positieve integer: het aantal runs.

Vervolgens per run:

- Een regel met daarop 2 integers gescheiden door spaties.
Het aantal rijen r $1 \leq r < 10^2$ $r \in \mathbb{Z}$
Het aantal kolommen k $1 \leq k < 10^2$ $k \in \mathbb{Z}$
- r regels met elk k integers gescheiden door spaties.
het getal op rij r , kolom k
Uiteindelijk komen alle getallen vanaf 1 t/m $r * k$ een keer voor.

Uitvoer

Per run een regel, met daarop Ja of Nee.

Voorbeeld invoer

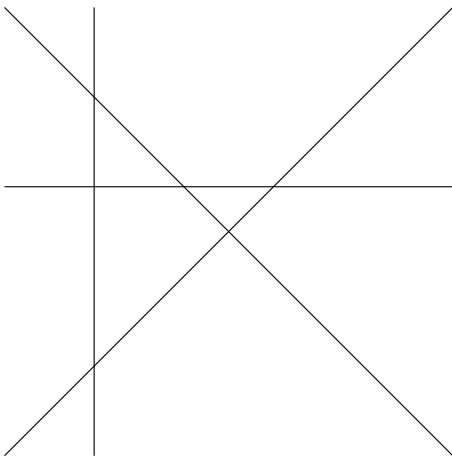
3
2 4
1 4 5 8
2 3 6 7
2 4
1 2 3 4
5 6 7 8
2 2
1 2
4 3

Bijbehorende uitvoer

Ja
Nee
Ja

E Gevangenis

De regering wil een nieuwe gevangenis bouwen. Ze hebben een aannemer gevonden die per muur betaald wil worden en dat lijkt ze wel wat omdat het de aannemer niet uit maakt hoe lang of hoog de muur is. De aannemer vindt het ook geen probleem om muren door elkaar heen te bouwen. Helaas mag de gevangenis maar één etage hoog worden, maar de regering heeft net een oneindig grote polder drooggelegd dus dat is geen probleem. Het is wel heel belangrijk om van te voren te weten hoeveel cellen de gevangenis uiteindelijk heeft in verband met voorbereidingen op organisatorisch niveau. De verantwoordelijke minister heeft een aantal ontwerpen gemaakt en wil van jou per ontwerp weten hoeveel cellen het bevat. Een cel heeft altijd een positieve, eindige oppervlakte.



Figuur 3: Een gevangenis met 3 cellen

Probleem

Maak een programma dat bij een gegeven set muren aangeeft hoeveel cellen de gevangenis heeft.

Invoer

Een regel, met daarop een enkele positieve integer: het aantal runs.

Vervolgens per run:

- Een regel met daarop één integer.
Het aantal muren m $1 \leq m < 10^2$ $m \in \mathbb{Z}$

- m regels met elk 4 integers gescheiden door spaties.

Coördinaat $x1$ $-50 \leq x1 \leq 50$ $x1 \in \mathbb{Z}$

Coördinaat $y1$ $-50 \leq y1 \leq 50$ $y1 \in \mathbb{Z}$

Coördinaat $x2$ $-50 \leq x2 \leq 50$ $x2 \in \mathbb{Z}$

Coördinaat $y2$ $-50 \leq y2 \leq 50$ $y2 \in \mathbb{Z}$

De muur is de lijn door $(x1, y1)$ en $(x2, y2)$. Deze punten zijn niet aan elkaar gelijk.

Uitvoer

Per run een regel, met daarop het aantal cellen dat het ontwerp van de gevangenis heeft.

Voorbeeld invoer

```
| 2  
| 2  
| 0 0 1 1  
| 1 0 0 1  
| 4  
| 0 0 0 1  
| 0 0 1 0  
| 0 1 1 0  
| 1 -1 2 0
```

Bijbehorende uitvoer

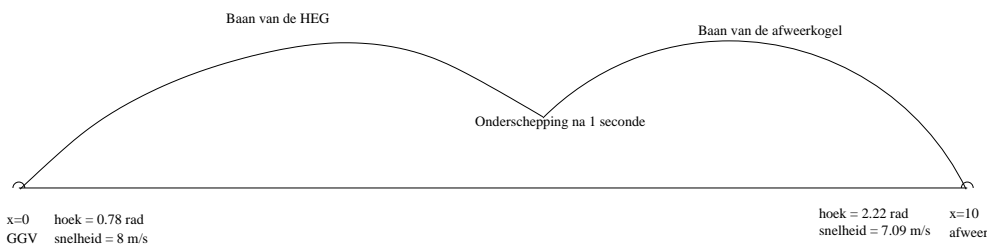
```
| 0  
| 3
```


G GGV's en HEG's

In een vlak terrein staat de Grote Gemene Vijand te schieten met tanks. Vroeger was dat al vervelend genoeg, maar nu hebben ze ook nog een nieuw soort HEG-munitie (Heel Erg Gemene munitie). Deze exploderen na een bepaalde tijd in honderden scherven, en richten zo veel schade aan.

Gelukkig is er wel een verdediging mogelijk tegen het HEG-wapen. Door een explosie te laten plaatsvinden precies op het tijdstip en de plek van de HEG-explosie, wordt alles teniet gedaan. Gelukkig leveren je spionnen je alle benodigde informatie, iedere keer dat er een HEG wordt afgeschoten. Op hetzelfde moment staat jouw afweergeschut klaar een vernietigingskogel af te schieten; je moet ze alleen nog even doorgeven waarheen en hoe hard.

Weerstand bestaat niet, we leven in vacuum. Zwaartekracht is constant en overal gelijk, versnelling -9.8 m/s^2 .



Figuur 4: Een irritant klein plaatje bij de 1e voorbeeld-input

Probleem

Maak een programma dat bij een gegeven landschap en HEG uitrekent hoe deze onderschept kan worden.

Invoer

Een regel, met daarop een enkele positieve integer: het aantal runs.

Vervolgens per run:

- Een regel met daarop 5 doubles, gescheiden door spaties.
 - De horizontale positie x_1 van de GGV $0 \leq x_1 < 10^4$ in meters
 - De hoek h waaronder de GGV schiet $0 < h < \pi$ in radialen, 0 =rechts, $\frac{\pi}{2}$ = boven, π = links.
 - De snelheid s waarmee de GGV schiet $0 \leq s < 10^3$ in m/s
 - De horizontale positie x_2 van jezelf $0 \leq x_2 < 10^4$ in meters
 - De tijdsduur d , waarna deze HEG gaat exploderen $0 \leq d \leq 30$ in seconden

Gegeven is dat de HEG altijd zo wordt afgevuurd, dat hij ergens boven de grond zou exploderen.

Uitvoer

Per run een regel, met daarop twee doubles gescheiden door spaties. Eerst de hoek, dan de snelheid van het afweergeschut. Beide afgerond op twee decimalen achter de komma.

Voorbeeld invoer

```
| 2  
| 0 0.78 8 10 1  
| 10 2.6 4 3 0.1
```

Bijbehorende uitvoer

```
| 2.22 7.09  
| 0.03 66.60
```

H Scheiden

Maximus en Willemina gaan scheiden, en proberen de inboedel te verdelen. Maar ja, wie krijgt het paleis, wie de kroon en wie de hond? Het moeilijke hieraan is dat geen van beide minder wil krijgen dan de ander, iedereen minstens de helft wil en het ook nog eens zonde is om iets dat jij heel graag wil aan die ander te laten als het hem niet zo veel uitmaakt. Dat laatste kan toch een grote hulp zijn bij de verdeling. Het is niet altijd mogelijk om de boel gelijk te verdelen, bijvoorbeeld als de hond voor beide meer gevoelswaarde heeft dan alle andere spullen bij elkaar. Maar zo goed mogelijk kan natuurlijk altijd...

Om het eerlijk te houden, spreken Maximus en Willemina van te voren af hoeveel gevoelswaarde ze over alle spullen mogen verdelen.

Probleem

Maximus en Willemina geven elk aan wat hun gevoelswaarde is voor ieder object in de inboedel. De score voor een persoon is de som over de gevoelswaarden van alle objecten die hij/zij krijgt. De score van een verdeling is de laagste van de twee persoonlijke scores. Schrijf een programma dat berekent wat de maximale score is die een verdeling kan hebben.

Invoer

Een regel, met daarop een enkele positieve integer: het aantal runs.
Vervolgens per run:

- Een regel met daarop twee integers.
Het aantal te verdelen objecten `aantal` $0 \leq \text{aantal} < 10^2$ `aantal` $\in \mathbb{Z}$
De totale afgesproken gevoelswaarde `totaal` $0 \leq \text{totaal} < 10^4$ `totaal` $\in \mathbb{Z}$
- Een regel met daarop `aantal` integers gescheiden door spaties.
De gevoelswaarden van Maximus `m` $0 \leq m < 10^2$ `m` $\in \mathbb{Z}$
- Een regel met daarop `aantal` integers gescheiden door spaties.
De gevoelswaarden van Willemina `w` $0 \leq w < 10^2$ `w` $\in \mathbb{Z}$

Uitvoer

Per run een regel, met daarop de maximale score die mogelijk is voor een verdeling.

Voorbeeld invoer

```
| 2
| 5
| 1 2 3 4 5
| 5 4 3 2 1
| 4
| 1 1 2 2
| 2 2 1 1
```

Bijbehorende uitvoer

```
| 9
| 4
```