

Opgave A - Hamsters

Babette heeft net een AIO-plaats gekregen op het Biologische Hamster Instituut. Haar taak is te onderzoeken hoe de snelheid waarmee hamsters hun voer eten verandert als ze meer of minder buurhamsters hebben. Om dit te doen heeft ze een groot aantal vergelijkingen opgesteld waarbij ze alle variabelen van waarden voorzien heeft. Alleen de waarde van de schroksnelheid x moet nog berekent worden.

Gelukkig zijn de vergelijkingen niet zo moeilijk. Babette houdt namelijk niet zo van delen en aftrekken. Er komen dus alleen vermenigvuldigingen en optellingen in de vergelijkingen voor. Er kunnen wel haakjes voorkomen, Babette wil haar rekenwerk namelijk zo klein mogelijk houden.

Aan u is nu de taak Babette bij haar AIO werk te helpen, door automatisch de vergelijkingen voor haar op te lossen.

Opgave A - Hamsters (I/O)

Source

A.C
A.PAS

Invoer

Op de eerste regel is een getal a te vinden dat aangeeft hoeveel verschillende vergelijkingen opgelost moeten worden. Op elke volgende regel staat steeds een vergelijking die alleen de volgende tekens kan bevatten:

[0,1,2,3,4,5,6,7,8,9,+,*=,x,(,)]

Uitvoer

Schrijf voor elke vergelijking op de invoer een regel naar de uitvoer van de vorm "x=..." waarbij van x op twee decimalen nauwkeurig de waarde gegeven wordt die x moet aannemen om de vergelijking kloppend te maken.

Voorbeeld

invoer

3

$x*6+(2*1)=8+4$

$(2*x)+3=10$

$6+2*(1+1)=5*x$

uitvoer

x=1.67 <= links alligned, 2 cijfers achter de komma

x=3.50

x=2.00

Opgave B - Vouwen

Een artiest heeft op een groot doorzichtig vel een abstracte tekening gemaakt die verdeeld is over een aantal vakken. Hij wil nu van deze poster een ansichtkaartenboekje maken. Voor het gemak zijn de vakken van de poster die gevuld zijn met tekeningen genummerd van links naar rechts en van boven naar beneden:

1	2	3	4
5	6	7	8
9	10	11	12

Artiesten hebben altijd een eigen willetje, en deze artiest wil beslist dat de ansichtkaarten op een door hem te bepalen volgorde worden bekeken.

Om dit te bereiken moet de poster opgevouwen worden zodat de kaarten van de poster van boven naar beneden bekeken op een bepaalde volgorde liggen. Merk op dat het niet uitmaakt of de kaarten in spiegelbeeld of ondersteboven bekeken worden aangezien het toch abstracte afbeeldingen zijn.

De poster kan op verschillende manieren gevouwen worden:

- lrn : de poster vouwen langs de Nde mogelijke verticale vouw, met links over rechts.
- rln : de poster vouwen langs de Nde mogelijke verticale vouw, met rechts over links.
- bon : de poster vouwen langs de Nde mogelijke horizontale vouw, met boven over onder.
- obn : de poster vouwen langs de Nde mogelijke horizontale vouw, met onder over boven.

Opgave B- Vouwen (I/O)

Source

B.PAS

B.C

Invoer

Op de invoer staat allereerst een getal a dat aangeeft hoeveel posters er gevouwen dienen te worden.

Per poster volgen dan drie regels met de volgende informatie

1. Het aantal horizontale vakjes h op de poster
2. Het aantal verticale vakjes v op de poster
3. Een regel met $h*v$ getallen die gewenste volgorde aangeven

De invoer zal altijd zodanig zijn dat er slechts één unieke oplossing is.

Uitvoer

Schrijf naar de uitvoer een regel die aangeeft op welke wijze de poster gevouwen dient te worden om de gewenste volgorde van boven naar beneden te bereiken.

Indien geen oplossing mogelijk is dient de tekst 'onmogelijk' gegenereerd te worden.

Voorbeeld

invoer

1

3

4

8 4 12 6 5 1 2 9 10 11 3 7

uitvoer

lr1 bo1 ob1 lr1 rl1 <-hoofdletters, gescheiden door
één spatie

Opgave C - Legpuzzel

Bij het opruimen van de zolder heb je een houten blokkendoos gevonden. Nadere bestudering leert dat het eigenlijk geen echte blokkendoos is, het is meer een soort van legpuzzel waarbij de puzzelstukjes gemaakt zijn door een aantal kubusjes aan elkaar te plakken.

Het geheel ziet er heel erg vrolijk uit aangezien elk puzzelstukje een andere kleur heeft, en ook de doos er van alle kanten anders uit ziet.

De doos heeft een rechthoekige vorm, en de lengten van de zijden zijn een veelvoud van de zijdelengten van de eenheidskubusjes.

Je vraagt je af op hoeveel manieren je de blokjes in de doos terug kunt stoppen.

Opgave C - Legpuzzel (I/O)

Source

C.PAS
C.C

Invoer

Op de invoer vind je allereerst een getal a dat aangeeft van hoeveel legpuzzels e het aantal oplossingen uit wilt rekenen.

Per legpuzzel vind je vervolgens de volgende informatie:

Allereerst een regel met twee getallen i, j die het formaat van de puzzel aangeven ($1 \leq i, j \leq 5$).

Op de volgende regel staat een getal n dat aangeeft uit hoeveel stukjes de legpuzzel bestaat.

Hierna volgen n regels die de vorm van de verschillende stukjes beschrijven.

Een omschrijving bestaat uit een getal m dat aangeeft uit hoeveel kubusjes een blokje bestaat, gevolgd door m x en y coördinaten die aangeven waar de eenheidsblokjes zich bevinden op een denkbeeldig coördinatenstelsel. ($0 \leq x, y \leq 4$).

De omschrijving 9 0 0 0 1 0 2 1 0 1 1 1 2 2 0 2 1 2 2 geeft bijvoorbeeld een vierkant blokje van 3×3 aan.

Er is altijd tenminste één kubus met x -coördinaat 0, en één kubus met y -coördinaat 0.

Uitvoer

Per legpuzzel een regel met een getal dat het aantal mogelijke rangschikkingen geeft.

Voorbeeld

invoer

```
1
2 2
3
1 0 0
1 0 0
2 0 0 0 1
```

uitvoer

```
8
```


Opgave D - Flip's verjaardagstaart

Flippie zit de hele dag achter zijn computer. Door alle straling van zijn monitor heeft hij een hersenbeschadiging opgelopen, waardoor zijn geheugen hem vaak op pijnlijke momenten in de steek laat. Een pijnlijk voorval is de verjaardag van zijn waarde vriend Joep. Flip was niet vergeten dat Joep jarig was, maar hij was totaal vergeten hoe oud Joep zou worden. Dit had geen probleem hoeven zijn, maar Flip had de goede gewoonte een taart voor Joep te kopen, met daarop evenveel kaarsjes als het aantal jaren dat Joep oud zou worden.

Dus ook dit jaar kocht Flip een grote taart, die bestond uit N ($N \leq 6$) punten. Op elke punt stond een aantal kaarsjes K ($K \leq 20$). Flip was dan weliswaar vergeten hoe oud Joep precies zou worden, hij had wel een idee van een ondergrens M ($M \leq 20$). Om aan een pijnlijke situatie te ontkomen bedacht Flip een plan.

De leeftijd van Joep zou hem wel weer te binnen schieten wanneer hij bij hem langs zou gaan (daar ging hij tenminste wel van uit). Daar zou hij Joep een aantal aanliggende punten (dit om de structuur van de taart niet teveel te verstoren) aanbieden, zodanig dat de som van het aantal kaarsjes op deze punten gelijk zou zijn aan de leeftijd van Joep.

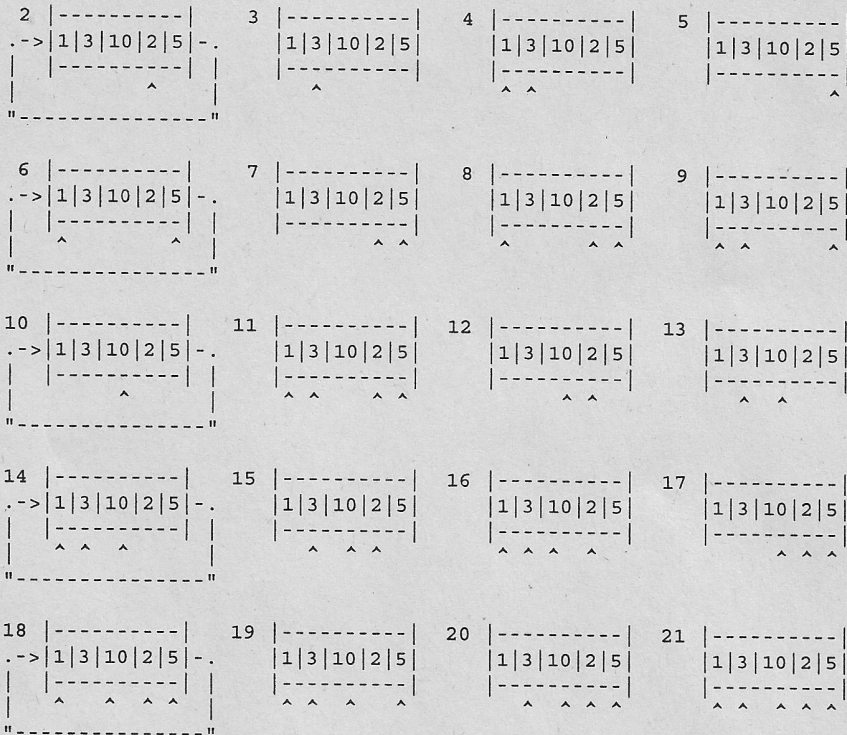
Om nu meer kans te hebben dat dit zou lukken plaatst Flippie nul of meer extra kaarsen op de verschillende punten van de taart, zodanig dat hij met het aantal kaarsen in een punt, of het totaal aantal kaarsen in aangrenzende punten, een ononderbroken reeks van leeftijden tussen M en I kan maken ($M, M+1, M+2, \dots, I$), dit om er zeker van te zijn dat hij niet toevallig precies de leeftijd van Joep zou missen, als deze $\leq I$ blijkt te zijn.

Natuurlijk wil hij ook het risico dat zijn beste vriend Joep ouder blijkt te zijn dan I minimaliseren. Dit zou immers het einde van een jarenlange hechte vriendschap kunnen zijn. Daarom wil Flip de aantallen kaarsen in de verschillende punten zodanig kiezen, dat I zo groot mogelijk wordt.

Net op het moment dat Flippie het een en ander uit aan het puzzelen is, speelt zijn hersenbeschadiging hem weer parten. Kun jij de radeloze Flip uit de brand helpen door een programma te schrijven dat voor hem uitrekent wat de maximale waarde van I is?

Opgave D - Flip's verjaardagstaart (extra)

Mogelijkheden waarop leeftijden gevormd kunnen worden met aantallen kaarsjes (1 3 10 2 5):



Opgave E - Roeien

Een roeivereniging houdt elk half jaar een tournooi voor haar leden. De roeivereniging bestaat uit een aantal teams, die verdeeld zijn over verschillende categorieën. De nationale roeibond heeft echter de beperking gesteld dat een team binnen 45 minuten niet meer dan één keer aan de start mag verschijnen.

De roeivereniging wil weten hoeveel dagen het tournooi minimaal gaat duren als per wedstrijd twee teams kunnen roeien en bij een oneven aantal teams in een categorie een baan leeg mag blijven. Bij een start mogen geen teams uit verschillende categorieën starten.

Bereken hoeveel dagen het tournooi minimaal gaat duren als gegeven is hoeveel minuten een dag mag duren. Verder is gegeven dat tussen twee starts tenminste 10 minuten zit.

Opgave E - Roeien (I/O)

Source

E.PAS
E.C

Invoer

Op de eerste regel van de invoerfile staat een getal a dat aangeeft van hoeveel tournooien de maximale lengte berekend moet worden.

Hierna volgt per tournooi de volgende informatie:

Een regel waarop aangegeven is hoeveel minuten t een wedstrijddag ten hoogste mag duren.

Een regel met het aantal teams n van de roeivereniging.

n regels met een voor het betreffende tournooi uniek teamnummer gevolgd door op dezelfde regel de nummers van de categorien (in oplopende volgorde) waarin het team roeit. Al deze nummers worden gescheiden door een spatie.

Uitvoer

De uitvoerfile bevat voor elk tournooi een regel met een getal dat aangeeft hoeveel dagen een tournooi minimaal moet duren.

Voorbeeld

invoer

```
2
40
4
1 1 2
2 1 2
3 1
4 1
100
2
1 1 2 3 4 5 6 7 8 9 10
2 1 2 3 4 5 6 7 8
```

uitvoer

```
2
4
```

<- links alligned, geen extra spaties

Opgave F - Rad van Fortuin

Jaren geleden, in het prille begin van de commerciële televisie in Nederland, had men een leuk idee voor een spelprogramma bedacht. Dit idee was niet helemaal nieuw, het was namelijk afgeleid van het bekende spel 'galgje'. Dit idee zou uiteindelijk een voorloper van het spel 'Rad van Fortuin' blijken te zijn.

Onder deskundige leiding van Hans van der Togt zouden een aantal kandidaten een onbekend woord moeten raden door middel van het gokken van de letters in het onbekende woord. Als een kandidaat een letter gokt die voorkomt in het woord zou deze nog een poging krijgen. Bij keuze van een letter die niet in het woord voorkomt zou de beurt naar de volgende kandidaat gaan.

Om het geheel attractiever te maken voor de kijkers thuis, werden twee elementen aan het spel toegevoegd:

1. De kandidaten zouden voor het gokken van een letter aan een felgekleurd rad draaien, met daarop te winnen geldbedragen.
2. Een liefallige jongedame zou de geraden letters omdraaien op een groot bord.

Er was echter een groot probleem en dit had te maken met de de jongedame. Zij kwam namelijk altijd te laat in de studio. Uiteindelijk kwam ze uiteindelijk altijd wel opdagen, alleen nooit op tijd. Omdat studiotijd kostbaar is werd er een slimme oplossing voor dit probleem bedacht:

De kandidaten zouden beginnen met het raden van letters, ook al was de assistente nog niet aangekomen of was ze nog voor de spiegel in de kleedkamer bezig.

Nu moest er alleen voorkomen worden dat de kandidaten geen letters zouden raden die voorkomen in het woord. Dan zouden er immers letters omgedraaid moeten worden, en dat kon nog niet.

De keuze van het te raden woord werd dus zo lang mogelijk uitgesteld. De kandidaten zouden wél te horen krijgen uit hoeveel letters het woord bestond, maar Hans zou een rij alternatieve woorden van dezelfde lengte achter de hand houden. Pas wanneer hij geen andere mogelijkheid meer had zou er een nieuwe letter omgedraaid worden, dit allemaal dus in de hoop dat zijn liefallige assistente dan wel in de studio aanwezig zou zijn.

Het is aan jou om als er een rij woorden van dezelfde lengte gegeven is, alsmede de volgorde waarin de kandidaten de letters raden, te bepalen wat het maximale aantal letters is dat geraden kan worden voordat de assistente nodig is.

Opgave F - Rad van Fortuin (I/O)

Source

F.PAS
F.C

Invoer

De eerste regel van de invoer bevat een getal r . Hierna volgen r runs, die de volgende vorm hebben:

De eerste regel van een run bevat twee getallen x ($1 \leq x \leq 100$) en y ($1 \leq y \leq 200$). x is het aantal woorden waaruit gekozen kan worden en y geeft aan hoe lang de woorden zijn. Het is duidelijk dat alle woorden in een run dezelfde lengte hebben.

De volgende x regels bevatten steeds een woord van lengte Y .

De laatste regel van een run bevat een rij letters in de volgorde waarin deze gekozen worden. Alle woorden bestaan slechts uit kleine letters en de rij met gekozen letters is altijd lang genoeg om tot een keuze te moeten komen.

Uit jarenlange televisieervaring is gebleken dat kandidaten vaak niet slim zijn, en dat het dus voor kan komen dat een letter gekozen wordt die al aan de beurt is geweest.

Uitvoer

Geef per run een getal dat aangeeft wat het maximale aantal keren is dat een letter geraden kan worden die niet voorkomt in het uiteindelijk te kiezen woord.

Voorbeeld

Invoer

```
1
5 5
beest
woord
tocht
einde
getal
oraxdnby
```

Uitvoer

```
5          <- links alligned, geen extra spaties
```

Opgave G - Rokkers

Je hebt zojuist de rechten geerfd van de nog niet eerder uitgebrachte liedjes van de populaire popgroep de Razende Rokkers. Om uit deze erfenis nog wat winsten te behalen heb je besloten om een verzameling CD's uit te geven met de nummers van deze groep.

Eigenlijk ken je de popgroep helemaal niet, en na één van hun nummer beluisterd te hebben had je wel weer genoeg van de vreselijke muziek. Om toch tot een redelijke indeling van de liedjes te komen heb je besloten de liedjes op volgorde van uitbrengen op de cd's te zetten.

Om de fans van de groep een plezier te doen wil je zoveel mogelijk liedjes op de cd-set kwijt, waarbij er natuurlijk geen liedjes half op de ene cd en half op de volgende mogen staan.

Opgave G - Rokkers (I/O)

Source

G.PAS
G.C

Invoer

Op de eerste regel van de invoer vind je een getal a dat aangeeft hoeveel maal deze test uitgevoerd gaat worden.

Vervolgens vind je per test twee regels met invoer getallen.

Op de eerste regel vind je drie getallen gescheiden door spaties. Allereerst het aantal cd's dat gevuld dient te worden, vervolgens een getal dat aangeeft hoeveel minuten muziek er op 1 cd kan, en als laatste een getal n dat aangeeft hoeveel liedjes er zijn.

Op de tweede regel vind je vervolgens n getallen die aangeven hoe lang alle liedjes duren. De liedjes zijn gesorteerd op volgorde van uitbrengen.

Je zult nooit alle liedjes op de cd's kunnen passen. Ook komen er geen liedjes voor die langer duren dan de lengte van 1 cd.

Uitvoer

Geef per testgeval een getal dat aangeeft hoeveel liedjes er maximaal op de cd-set te plaatsen zijn.

Voorbeeld

invoer

```
3
3 5 10
5 5 5 5 5 5 5 5 5
4 6 5
4 3 4 4 5
3 5 10
3 5 1 2 3 5 4 1 1 5
```

uitvoer

```
3
4
6
```

<- links alligned, geen extra spaties

Opgave H - Kerning

In het lange donker verleden van het computertijdperk, gebruikte iedereen fixed-width lettertypen. Het was gemakkelijk uit te rekenen hoeveel karakters er op één regel passen. Omdat de 'M' en de 'i' net als alle andere karakters dezelfde breedte hadden.

Proportionele lettertypes deden kort daarna hun intrede. Om uit te rekenen hoeveel karakters er nu op één regel passen, moet van elk karakter de breedte bekend zijn. Een 'M' had bijvoorbeeld breedte 10 en 'i' breedte 4.

De gebruikers waren echter nog steeds niet tevreden en verbeterden het uiterlijk van proportionele lettertypen door gebruik te maken van 'kerning'. Wanneer een 'V' onmiddellijk gevolgd wordt door een 'A', dan kan de 'A' naar links worden geschoven om disproportioneel grote ruimte tussen de twee karakters te voorkomen. In het algemeen kan kerning worden toegepast wanneer door de vorm van twee karakters er een relatief grote ruimte tussen de karakters komt. Afhankelijk van de precieze vorm van de karakters zijn voorbeelden van zulke paren: "AV ", "A", "LT", "ij", etc..

Het is nu de bedoeling dat een programma te schrijven dat de lengte van een bepaalde string uitreken, gegeven de vormen van alle afzonderlijke karakters. Kerning moet worden toegepast om de karakters zo dicht mogelijk op elkaar te zetten, zonder dat pixels van twee opeenvolgende karakters elkaar raken danwel overlappen. (Niet horizontaal, verticaal en diagonaal). Verder geldt dat de meest rechtse plaats van een karakter i altijd minstens één hoger is dan de meest rechtse plaats van karakter $i-1$. Dit voorkomt dat de string "....." er uit gaat zien als ";;;".

Opgave H - Kerning (I/O)

Source

H. PAS

H. C

Invoer

Op de invoer is zoals gewoonlijk allereerst een getal a te vinden dat aangeeft hoeveel verschillende problemen er nog volgen. Hierna vind je per probleem de volgende informatie:

Allereerst een regel met een getal n dat aangeeft hoeveel karakters er gedefinieerd zijn. ($1 \leq n \leq 94$). Hierna worden de n karakters gedefinieerd.

Een karakter definitie ziet er als volgt uit:

Op de eerste regel

- het gedefinieerde karakter (ASCII '!'.. '~'). Geen enkel karakter is dubbel gedefinieerd.
- de breedte w van het karakter ($1 \leq w \leq 20$). De breedte van de zogenaamde 'bounding box'.
- de hoogte h van het karakter ($1 \leq h \leq 20$). De hoogte van de zogenaamde 'bounding box'.
- de y -offset y van de data ($0 \leq y \leq 20-h$).

Vervolgens h regels van w karakters ('.' of '*'). Tenminste één regel begint met een '*' en tenminste één regel eindigt met een '*'. Tevens bevatten de eerste en de laatste van de h regels een '*'.

Na alle karakterdefinities volgt nog één regel met daarop een woord dat geheel bestaat uit gedefinieerde karakters.

Opgave H - Kerning (I/O)

Uitvoer

Eén regel die de breedte van het woord in pixels weergeeft (breedte < 80).

Voorbeeld

invoer

1

2

A 7 6 1

...*...

..*.*..

.*.***.

.*****.

.....

.....

' 2 2 5

.*

*.

A''A

uitvoer

18 <- links alligned, geen overbodige spaties