Voorronde NKP 2001

Inhoudsopgave

A Factorialiseren **1**

B Rekeningrijden in Mobilia **3**

C Sleutels **5**

D Shannon-getallen **7**

E Pacman **9**

F Harmonie **11**

G Schilderijen **13**

H Quiz **15**

A Factorialiseren

Een *priemgetal* is een natuurlijk getal dat slechts deelbaar is door zichzelf en door 1. Het getal 1 is zelf echter geen priemgetal. Elk natuurlijk getal kan op precies één manier worden geschreven als het product van een aantal priemgetallen; daarbij is het mogelijk dat sommige priemgetallen meer dan ééns voorkomen. We noemen dit de ontbinding van een getal in *priemfactoren*. Zo kunnen we het getal 45 ontbinden in priemfactoren als $3 \times 3 \times 5$.

Arjen is een expert op het gebied van priemfactoren, vooral wanneer het gaat om priemfactoren van hele grote getallen. Het ontbinden van grote getallen is meestal erg moeilijk. Er zijn grote supercomputers en buitengewoon ingewikkelde algoritmen voor nodig. Voor sommige klassen van grote getallen bestaan echter snellere en eenvoudigere technieken.

Sinds kort bestudeert Arjen een nieuwe klasse van grote getallen: de *faculteiten* (of factorials in het Engels). Het getal $n!$ (n faculteit) is het product van de gehele getallen 1 tot en met n, bijvoorbeeld $6! = 1 \times 2 \times \ldots \times 6 = 720$. Deze getallen worden al snel erg groot; het getal 1000! is bijvoorbeeld al 2569 cijfers lang.

Probleem

Bedenk voor Arjen een algoritme om $n!$ in priemfactoren te ontbinden, en schrijf een computerprogramma om het algoritme te testen.

Invoer

Op de eerste regel van de invoer staat het aantal getallen a dat nog zal volgen. Daarna komen a regels, met op elke regel een geheel getal n met $2 \leq n \leq 1000$ (duizend).

Uitvoer

De uitvoer bestaat uit a regels, met op elke regel een priemontbinding. Voor elke ingelezen n geef je de priemontbinding van $n!$ in de vorm van een vergelijking. Rechts van het = teken geef je een opsomming van de priemfactoren, beginnend bij de kleinste en oplopend in grootte. Als een factor meer dan ééns (met een macht) in de ontbinding voorkomt, geef je de macht (of multipliciteit) met behulp van de ^ notatie. Bij priemfactoren die slechts éénmaal voorkomen mag je het ^ symbool **niet** gebruiken.
De = en * symbolen worden van de getallen gescheiden door spaties.

Voorbeeld

invoer	bijbehorende **uitvoer**
3 3 5 8	3! = 2 * 3 5! = 2^3 * 3 * 5 8! = 2^7 * 3^2 * 5 * 7

B Rekeningrijden in Mobilia

Het Mobiliaanse ministerie van geMotoriseerd Particulier Vervoer (MPV) heeft voorgesteld een systeem van rekeningrijden op 's lands rijkswegen in te voeren. Iedere rijksweg in Mobilia vormt altijd een direkte verbinding tussen twee verschillende Mobiliaanse steden. Verder moet nog opgemerkt worden, dat deze wegen onderverdeeld zijn in **B**-*wegen*, dit zijn de **b**elangrijke wegen en **MB**-*wegen*, dit zijn de **m**inder **b**elangrijke wegen.
Tussen twee steden kan overigens meer dan één rijksweg liggen.

De voor het rekeningrijden benodigde tolpoorten zullen in de steden van Mobilia geplaatst worden. Via deze steden zijn immers de bijna altijd verstopte rijkswegen van Mobilia te bereiken. Tot grote vreugde van de lokale overheden zullen de inkomsten van de tolheffing volledig ten goede van de steden komen.

Om de hebzucht van de lokale overheden enigszins in te tomen heeft het ministerie van MPV voor plaatsing van de tolpoorten echter de volgende voorwaarden opgesteld:

1. In iedere stad mag hoogstens één tolpoort geplaatst worden.

2. Als er tussen twee steden S_1 en S_2 minstens één **B**-weg aanwezig is, dan moet er *tenminste één* tolpoort geplaatst worden in S_1 en/of S_2.

3. Als er tussen twee steden S_1 en S_2 tenminste één **MB**-weg loopt, dan mag er *hoogstens* in één van deze twee steden een tolpoort geplaatst worden.

Probleem

De ministerraad van Mobilia wil van jou een programma hebben dat in staat is om, gegeven een verzameling van steden en rijkswegen,

- na te gaan of het mogelijk is om onder bovenstaande voorwaarden het rekeningrijden in te voeren;

- als het antwoord op de vorige vraag bevestigend is, de verzameling van steden te geven waar een tolpoort geplaatst moet worden zodat aan alle voorwaarden is voldaan.

Invoer

De invoer begint met een regel met daarop het aantal gevallen dat zal volgen. Daarna volgt per geval

- een regel met twee positieve getallen N en M, waarbij N ($2 \leq N \leq 100$) het aantal steden, en M ($M \geq 1$) het totaal aantal verbindingswegen tussen de steden aangeeft;
- M regels met elk twee positieve gehele getallen X en Y gevolgd door een string W, waarbij X en Y ($1 \leq X, Y \leq N$, $X \neq Y$) de eindpunten van een verbindingsweg zijn en W het type verbindingsweg aangeeft. Hierbij staat W = '`B`' voor een belangrijke weg en W = '`MB`' voor een minder belangrijke weg.

De invoer is zo gekozen dat er altijd maximaal één unieke manier is om de tolpoorten te plaatsen.

Uitvoer

De uitvoer dient per geval uit één regel te bestaan. Als niet aan de voorwaarden voor het rekeningrijden kan worden voldaan dient deze regel uit het antwoord '`NO`' te bestaan. Anders moet deze regel een opsomming van de steden(nummers) bevatten (in opklimmende volgorde, gescheiden door enkelvoudige spaties) waarin een tolpoort geplaatst moet worden.

Voorbeeld

invoer	bijbehorende **uitvoer**
2 3 5 1 3 B 2 3 B 1 2 B 1 3 MB 1 2 MB 4 7 1 4 B 1 3 B 2 3 B 1 2 B 1 3 MB 2 3 MB 1 2 MB	2 3 NO

C Sleutels

In het stadje Sloterdam woont een sleutelmaker, die sleutels maakt. Iedere sleutel heeft een uniek deel, wat bepaalt of een sleutel wel of niet op een bepaald slot past. Dit deel heeft een bepaalde geheeltallige breedte B en een, eveneens geheeltallige, hoogte H. Dit houdt in dat het bestaat uit B 'staafjes', die ieder maximaal H eenheden hoog kunnen zijn. Bijvoorbeeld:

Deze sleutel heeft breedte 4 en hoogte 3. De 'staafjes' hebben de hoogten 3, 1, 3 en 2.

Nu is het mogelijk om van één sleutel een andere sleutel te maken door er één of meerdere stukjes af te vijlen. Dit wil de sleutelmaker natuurlijk vermijden. Daarom zorgt hij er altijd voor, dat de sleutels uit één set zó van elkaar verschillen, dat dit niet mogelijk is. Hij heeft echter een leerling in dienst die dit idee niet helemaal begrepen heeft. Deze leerling heeft voor de sleutelmaker een groot aantal ontwerpen gemaakt voor sleutelsets, zodat de sleutelmaker aan het ontwerpen niet zoveel tijd kwijt is. Hierbij heeft hij er echter niet op gelet of ze wel aan de eis van de sleutelmaker voldoen. Hij heeft er wel voor gezorgd dat alle sleutels in zo'n set van elkaar verschillen. Voordat de sleutelmaker een sleutelset daadwerkelijk gaat maken, wil hij er natuurlijk wel van verzekerd zijn dat het een goede set is. Gelukkig heeft de leerling zijn ontwerpen in een tekstbestand op de computer opgeslagen, zodat dit makkelijk met een computerprogramma te controleren is.

Probleem

Schrijf een programma dat van een gegeven sleutelset bepaalt of dit een goede set is. Dat wil zeggen een set, waarbij het niet mogelijk is om van een sleutel uit de set een andere sleutel uit dezelfde set te maken, door er één of meerdere delen af te vijlen. Indien de set niet goed is, print dan díe sleutels waarmee de eigenaar de deur van iemand anders zou kunnen openen, indien hij er een deel af zou vijlen.

Invoer

Eerst een regel met daarop een enkele positieve integer: het aantal sets. Daarna per set:

- één regel met daarop 3 integers: de breedte ($2 \leq B \leq 20$) en de maximale hoogte ($2 \leq H \leq 9$) van de sleutels, en het aantal sleutels N in deze set ($2 \leq N \leq 100$)
- N regels met op iedere regel een integer van B cijfers, die een sleutel uit de set representeert. Ieder cijfer geeft de hoogte h van een van de staafjes weer ($0 \leq h \leq H$), waarbij deze staafjes uiteraard in de volgorde staan waarin ze op de sleutel zouden komen.

Uitvoer

Voor iedere sleutelset één regel met daarop het woord '`GOED`', indien de set goed is. Indien de set niet goed is, alle sleutels die door te vijlen omgezet zouden kunnen worden in een (of meerdere) andere sleutel(s) uit dezelfde set, gescheiden door een spatie. De sleutels dienen door hetzelfde getal te worden gerepresenteerd als bij de invoer, en ook in dezelfde volgorde te staan waarin ze bij de invoer voorkomen.

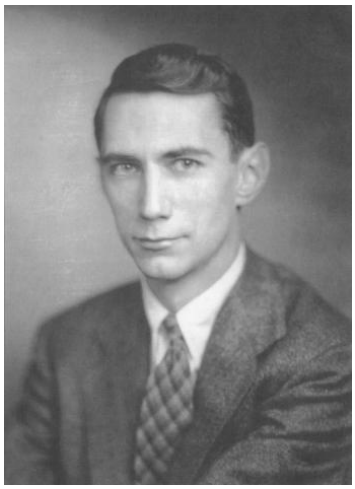
Voorbeeld

invoer	bijbehorende **uitvoer**

`2`
`3 2 3`
`112`
`121`
`211`
`2 3 3`
`22`
`23`
`21`
	`GOED`
	`22 23`

D Shannon-getallen

Begin dit jaar overleed op 84-jarige leeftijd de Amerikaanse wetenschapper en elektrotechnisch ingenieur Claude Elwood Shannon. Met enkele baanbrekende artikelen, die hij in de jaren na de Tweede Wereldoorlog publiceerde, ontstonden hele nieuwe vakgebieden zoals informatie- en coderingstheorie. Eerder was hij al de eerste die de overeenkomst opmerkte tussen schakel-elektronica en Boolse algebra, en ook schreef hij een klassiek artikel over hoe je een computer zou kunnen programmeren om schaak te spelen, waarin hij het befaamde *minimax* algoritme introduceert.



Claude Elwood Shannon
30-4-1916 – 24-2-2001

Naast dit serieuze werk was hij ook een fervent jongleur en liefhebber van vreemde apparaten. Toen het binaire stelsel al lang gemeengoed was voor elektronische computers ontwierp hij de THROBAC, een computer die intern in het Romeinse(!?) talstelsel rekent. Een ander tegendraads artikel van zijn hand gaat over een alternatieve representatie voor getallen. Deze opgave gaat over deze 'Shannon-getallen'.

Wanneer we een geheel getal G in het r-tallig stelsel uitdrukken gebruiken we daarvoor N cijfers a_i, met $0 \leq i < N$ en $a_i \in \{0,1,\ldots,r-2,r-1\}$, zodanig dat geldt:

$$G = \sum_{i=0}^{N-1} a_i r^i$$

Deze 'normale' getalrepresentatie heeft vele mooie eigenschappen. Echter, het is zeker niet de enige manier om efficiënt getallen te representeren!

Een alternatieve representatie die zeer veel lijkt op de bovenstaande krijgen we door in plaats van de cijfers $\{0,1,\ldots,r-2,r-1\}$, de cijfers $\{-\frac{r-1}{2},-\frac{r-1}{2}+1,\ldots,\frac{r-1}{2}-1,\frac{r-1}{2}\}$ te nemen. Dit werkt alleen voor een oneven grondtal r. In de rest van deze opgave nemen we het grondtal $r = 11$ constant, zodat we de cijfers $\{-5,-4,-3,-2,-1,0,+1,+2,+3,+4,+5\}$ hebben. Wanneer we nu de negatieve cijfers noteren met een accent (dus 5' in plaats van -5), krijgen we de *Shannon-representatie*.

Een voorbeeld helpt wellicht om dit te verduidelijken. Het getal (Shannon-representatie) **12'3'405'** kunnen we als volgt converteren naar decimaal:

12'3'405' =
$(-5)\cdot 11^0 + (0)\cdot 11^1 + (4)\cdot 11^2 + (-3)\cdot 11^3 + (-2)\cdot 11^4 + (1)\cdot 11^5 =$
$-5 + 0 + 484 - 3993 - 29282 + 161051 =$
128255

De Shannon-representatie biedt, vanwege de symmetrische verdeling van de cijfers, bepaalde voordelen tegenover de normale getalrepresentatie. Zo zijn de 'vermenigvuldigingstafels' dankzij de symmetrie zeer eenvoudig, hetgeen belangrijk zou kunnen zijn met het oog op machinale verwerking.

Het belangrijkste voordeel wordt ook al door Shannon opgemerkt: met de nieuwe representatie is het onmogelijk voor winkeliers om 'bedrieglijke' prijzen te hanteren zoals Fl. 4,99 of Fl.99,95!

Probleem

Gegeven een aantal gehele getallen (in standaard decimale notatie), wat is de bijbehorende Shannon-representatie?

Invoer

Op de eerste regel van de invoer staat het aantal runs R. Daarna volgen R regels met daarop een geheel getal n in decimale notatie; $-1000000 \leq n \leq 1000000$.

Uitvoer

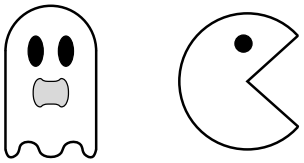
De uitvoer bestaat uit R regels met daarop in volgorde de Shannon-representatie van de in de invoer gespecificeerde getallen.

Voorbeeld

invoer	bijbehorende **uitvoer**
3 1000 -1000 0	13'31' 1'33'1 0

E Pacman

Help ! Pacman is gevangen in een doolhof, waar hij wordt achtervolgd door een spook. Hij is dringend op zoek naar een computerprogramma dat voor hem een route naar de uitgang kan bepalen.



Het doolhof bestaat uit een plat rechthoekig rooster van vakjes. Sommige van deze vakjes zijn geblokkeerd door een muur zodat niemand er kan komen. In 1 of meer vakjes bevindt zich een uitgang in de vorm van een teleporter waarmee Pacman naar de volgende level kan vluchten (spoken kunnen niet door teleporters).

Zowel Pacman als het spook beschikken over een kaart waarop de muren van het doolhof en de locatie van de uitgang zijn aangegeven. Ook zijn ze voortdurend op de hoogte van elkaars positie.

In elke beurt mag eerst Pacman en vervolgens het spook een stapje lopen. Dit houdt in dat ze een vakje naar links, rechts, boven of onder mogen bewegen, mits het vakje van bestemming niet geblokkeerd is door een muur. Het is ook toegestaan om op je plaats te blijven staan. Over de rand van het doolhof naar buiten stappen is **niet** mogelijk.

Als Pacman zich in hetzelfde vakje bevindt als het spook (doordat hij een vakje met het spook binnenloopt, of doordat het spook zijn vakje binnenloopt) wordt hij onmiddellijk opgegeten. Zodra Pacman een vakje met een teleporter binnenloopt is hij veilig ontsnapt uit het doolhof. Het spook kan **niet** in de vakjes met een teleporter komen.

Pacman wil absoluut niet opgegeten worden en probeert om in zo min mogelijk beurten veilig de uitgang te bereiken. Het spook doet juist haar uiterste best om Pacman te vangen. Ook wanneer dat niet lukt zal ze proberen om het Pacman zo lastig mogelijk te maken, zodat hij zoveel mogelijk beurten nodig heeft om bij de uitgang te komen. Zowel Pacman als het spook zijn erg slim en kunnen het hele verdere verloop van het spel overzien. Beiden volgen daardoor een foutloze strategie.

Probleem

Gegeven een kaart van het doolhof, waarop tevens de beginposities van Pacman en het spook zijn gemarkeerd. Bepaal of Pacman veilig de uitgang van het doolhof kan bereiken, en zo ja, hoeveel beurten hij daarvoor minimaal nodig heeft. Houdt er daarbij rekening mee dat het spook het hem zo lastig mogelijk zal maken.

Invoer

Op de eerste regel staat een positief geheel getal: het aantal spelletjes. Daarna volgt voor elk spelletje een beschrijving in de volgende vorm:

- Een regel met twee getallen w en h; de breedte en hoogte van het doolhof met $2 \leq w, h \leq 20$.
- Een kaart van het doolhof in de vorm van h regels met op elke regel w karakters. Een muur wordt aangegeven met de letter 'X', een uitgang met de letter 'U'. De kamer waarin Pacman het spel begint wordt aangegeven met een 'P', de kamer van het spook met een 'S'. Overige (lege) kamers worden aangegeven met een '.' (punt).

Er is altijd precies 1 kamer met een 'P' en 1 kamer met een 'S'.

Uitvoer

Voor elk spel geef je 1 regel uitvoer. Als Pacman veilig bij de uitgang kan komen, geef je het aantal beurten dat hij daarvoor nodig heeft in de volgende vorm: '`Pacman bereikt de uitgang in beurt` *n*'. Anders geef je als uitvoer de volgende tekst: '`Geen veilige route`'

Voorbeeld

invoer	bijbehorende **uitvoer**
2 3 3 P.S .X. ..U 5 3 ..P.. ..... U.S.U	Pacman bereikt de uitgang in beurt 4 Geen veilige route

F Harmonie

De afstand tussen twee tonen – het interval – wordt bepaald door de verhouding van de frequenties. Het interval tussen een toon van 400 Hz en een van 800 Hz wordt oktaaf genoemd. Een oktaaf correspondeert met een verhouding $1 : 2$. Het interval tussen 155 Hz en 310 Hz is dus ook een oktaaf, want $155 : 310 = 1 : 2$. Als de tweede toon hoger is dan de eerste (dat wil zeggen dat de tweede toon een hogere frequentie heeft), dan spreken we van een stijgend interval. Is de eerste toon hoger, dan is het interval dalend. In deze opgave beschouwen we alleen stijgende intervallen.

Een componist wil een muziekstuk maken gebaseerd op zuivere intervallen. Hij beschouwt de volgende intervallen als zuiver:

verhouding	**naam**	**meervoud**
1 : 2	oktaaf	oktaven
2 : 3	kwint	kwinten
3 : 4	kwart	kwarten
4 : 5	terts	tertsen

Hij vraagt zich af of hij een gegeven stijgend interval (bijvoorbeeld 100 : 1000) kan schrijven als een reeks van zuivere intervallen. Dat kan in dit geval:

100 : 125 = 4 : 5 (terts)
125 : 250 = 1 : 2 (oktaaf)
250 : 500 = 1 : 2 (oktaaf)
500 : 1000 = 1 : 2 (oktaaf)

De volgorde van de intervallen ligt niet vast; in het voorafgaande voorbeeld kan de terts ook aan het einde komen: 100 : 200 : 400 : 800 : 1000, of ergens middenin. Ook kan een interval soms op meer dan één manier worden verdeeld; het interval 3000 : 8000 kan worden gesplitst als:

3000 : 6000 : 8000 (een oktaaf en een kwart)
3000 : 4500 : 6000 : 8000 (een kwint en twee kwarten)

De componist streeft echter naar de eenvoudigste verdeling. Daaronder verstaat hij een verdeling in zo weinig mogelijk intervallen.

Probleem

Gegeven een interval, bepaal de eenvoudigste verdeling in zuivere intervallen, als die bestaat.

Invoer

De eerste regel van het invoerbestand bevat een getal: het aantal testgevallen. De volgende regels bevatten ieder een testgeval. Een testgeval bestaat uit twee gehele getallen l en h, de frequentie van de laagste, resp. de hoogste toon van het interval. Voor deze getallen geldt: $1 \leq l < h \leq 20000$.

Uitvoer

Voor elke regel in het testbestand een regel met daarop de eenvoudigste verdeling in intervallen (zie voorbeeld). De gebruikte intervallen moeten geordend zijn: eerst (eventuele) tertsen, vervolgens eventuele kwarten, kwinten en oktaven. Als er geen verdeling in intervallen mogelijk is, de tekst: '`ONMOGELIJK`'.

Voorbeeld

invoer	bijbehorende **uitvoer**

`4`
`155 310`
`100 1000`
`3000 8000`
`1000 1001`
	`1 OKTAAF`
	`1 TERTS 3 OKTAVEN`
	`1 KWART 1 OKTAAF`
	`ONMOGELIJK`

G Schilderijen

Een vereniging van schilderijenverzamelaars wil de collectie, die bij haar leden in bezit is, tentoonstellen aan het publiek. Hiervoor willen ze een aantal muren afhuren in een museum.

De eigenaars van de schilderijen willen echter wel een kleine vergoeding voor het uitlenen van hun schilderijen. Verder wil het museum een betaling voor het gebruik van de muren en kunnen ze maximaal maar 2 muren verhuren aan de vereniging. De prijs van een muur is afhankelijk van de ligging en breedte van de desbetreffende muur.

De vereniging heeft een niet al te groot budget tot haar beschikking en kan dus niet alle schilderijen kwijt. Deze wil daarom zorgen dat de totale publiekswaarde van de schilderijen het hoogst is.

In het museum moet tussen de zijkant van een muur en een schilderij en tussen de verschillende schilderijen onderling tenminste 10 cm zitten. Ook mogen schilderijen alleen maar naast elkaar gehangen worden. Uit esthetisch oogpunt wil het museum niet meer dan 4 schilderijen op een muur hangen.

Probleem

Bereken, gegeven een dataset, welke muren en schilderijen gehuurd moeten worden om een zo groot mogelijke publiekswaarde tentoon te stellen. Als er meerdere mogelijkheden zijn met dezelfde publiekswaarde wil men daarvan uiteraard de goedkoopste.

Invoer

Eerst één regel met daarop een enkele positieve integer: het aantal datasets. Daarna per dataset:

- 1 regel met het budgetbedrag b van de vereniging in gehele guldens ($0 < b \leq 1\,000\,000$)
- 1 regel met het aantal muren m ($0 < m \leq 5$)
- m regels met daarop twee positieve integers gescheiden door een spatie: de breedte mb in cm ($0 < mb \leq 10\,000$) en de huurprijs mh in gehele guldens ($0 < mh \leq 1\,000\,000$)
- 1 regel met het aantal schilderijen s ($0 < s \leq 10$)
- s regels met daarop drie positieve gehele getallen gescheiden door een spatie: de breedte sb in cm $0 < sb \leq 10\,000$), de huurprijs sh in gehele guldens ($0 < sh \leq 1\,000\,000$) en de publiekswaarde sp ($0 < sp \leq 1\,000\,000$)

Uitvoer

Per dataset 1 uitvoerregel met daarop twee positieve integers, gescheiden door een spatie. Daarvan is het eerste getal het bestede bedrag in gehele guldens en het tweede getal de publiekswaarde.

Voorbeeld

invoer	bijbehorende **uitvoer**
1 2000 2 400 1000 500 1300 4 100 200 40 200 300 30 150 200 20 70 400 40	1800 100

H Quiz

Bij de quiz '*Geld speelt een rol*' spelen twee teams tegen elkaar. De teams moeten een aantal vragen beantwoorden. Het team dat als eerste de vraag goed beantwoordt krijgt een aantal punten, afhankelijk van de vraag. Vorige week was de puntenverdeling als volgt:

vraag	punten
1	3
2	6
3	9
4	6
5	9
6	3

Wordt de vraag niet binnen de tijd beantwoord, dan krijgt niemand de punten. Het team dat aan het eind de meeste punten heeft, wint. Voor elke punt die zij méér hebben dan het andere team krijgen zij 100 gulden.

Vorige week sloot het B-team een weddenschap af, dat zij met precies duizend gulden naar huis zouden gaan. Dat illustreert hoe slim zij zijn, want met de bovengenoemde getallen is het onmogelijk om een verschil van 10 punten te realiseren. Ze verloren dan ook. Ze beantwoordden vraag 2, 4 en 6 goed (totaal 15 punten), maar het ITS-team beantwoordde vraag 3 en 5 goed (totaal 18 punten). Het ITS-team ging dus met 300 gulden naar huis. Het B-team verloor bovendien de weddenschap, maar dat hadden ze van te voren kunnen weten.

Probleem

Gegeven een rij getallen (de puntenaantallen voor de vragen) en een getal (het verschil). Is het mogelijk dat een team eindigt met precies dit puntenaantal voorsprong?

Invoer

De eerste regel van de invoerfile bevat een enkel getal: het aantal testgevallen. Elk testgeval wordt beschreven in twee regels. De eerste regel bevat twee niet negatieve gehele getallen: het gewenste verschil v en het aantal vragen n ($0 < n \leq 50$). De tweede regel bevat n positieve gehele getallen: voor elke vraag het aantal punten dat de vraag oplevert. Een vraag levert hooguit 50 punten op.

Uitvoer

Voor elk testgeval in de invoer bevat de uitvoer één regel, met daarop het woord ’JA’ als het mogelijk is dat de twee teams zodanig op de vragen scoren, dat het ene team aan het eind *v* punten meer heeft dan het ander team. Als dat niet het geval is moet het woordje ’NEEN’ worden gegeven.

Voorbeeld

invoer	bijbehorende **uitvoer**
2 10 6 3 6 9 6 9 3 3 6 1 2 3 4 5 6	NEEN JA