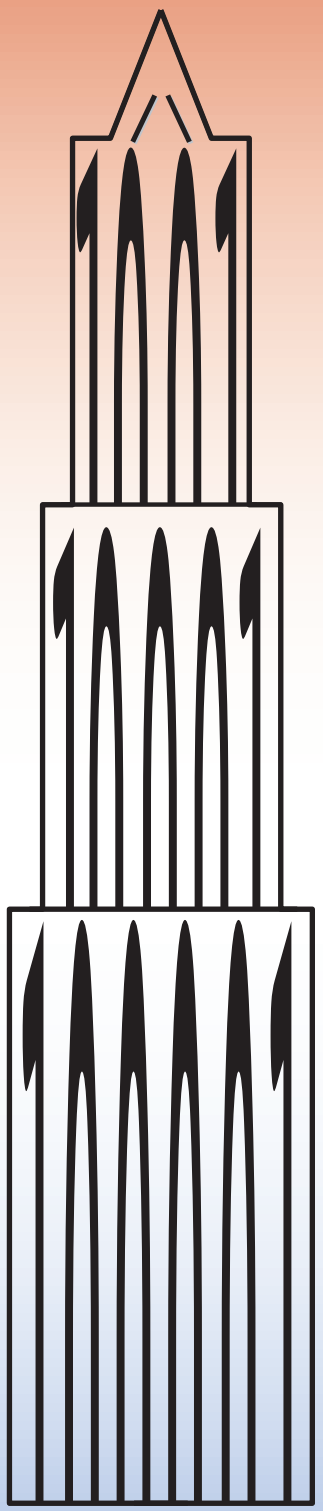
Opgaven

NKP*04*

23 oktober 2004



NKP04

UTRECHT

1001011010101101010101101010101101010110101011
0101010010001010111010101101101010110101011010

A Bustijden

Probleem

De Nederlandse Spoorwegen hebben wel eens last van problemen, maar ook de Utrechtse stadsbussen functioneren niet perfect: zoals algemeen bekend rijden bussen vaak niet op tijd. Soms ben je op tijd bij de bushalte en komt de bus veel te laat; soms kom je precies op tijd, maar blijkt de bus al net langs gekomen te zijn.

Op een dag willen we met de stadsbussen in Utrecht van de ene locatie naar de andere locatie reizen. Hiervoor hebben we de route uitgestippeld en we moeten een aantal keer overstappen. In totaal nemen we n buslijnen, waarbij we steeds opnieuw geluk of pech kunnen hebben.

We kennen de dienstregeling, maar helaas weten we niet hoeveel vroeger of later dan de dienstregeling de bussen rijden. Gelukkig gaan we als studenten zo vaak met de bus, dat we wèl weten hoeveel minuten iedere bus maximaal vroeger of later rijdt dan de dienstregeling aangeeft. Bussen doen er altijd even lang over om van de ene halte naar de andere te komen.

We vragen ons af hoe laat we aankomen in het geval dat alles mee of tegen zit. We staan om 9 uur 's ochtends bij de eerste bushalte en hoe erg alles ook tegenzit, we kunnen altijd voor middernacht aankomen. Er vertrekken geen bussen voor vier uur 's ochtends.

Invoer

Op de eerste regel één positief getal: het aantal testgevallen.

Daarna per testgeval:

- Een regel met één positief getal n met $1 \leq n \leq 100$: het aantal bussen dat we nemen op de route.
- Daarna n blokken van de volgende vorm, waarbij het i-de blok de i-de buslijn op de route beschrijft:
  - Een regel met twee positieve getallen n_i en t_i met $1 \leq n_i \leq 100$ en $1 \leq t_i \leq 100$: het aantal bussen op deze lijn en het aantal minuten dat die bussen erover doen.
  - n_i regels van het volgende formaat: "hh:mm (a)", waarbij "hh:mm" een geldige tijd weergeeft in 24 uurs-notatie[1] en a een getal met $0 \leq a \leq 100$ dat het aantal minuten is dat een bus maximaal vroeger of later kan komen.

Uitvoer

Per testgeval:

- Een regel met twee tijdstippen in 24 uurs-notatie, gescheiden door een spatie: het vroegste (alles zit mee) en laatste (alles zit tegen) tijdstip van aankomst.

[1] Onder 24 uurs-notatie verstaan we een tijdstip van de vorm "hh:mm", waarbij iedere 'h' en 'm' een cijfer (evt. 0) is en $0 \leq hh < 24$, $0 \leq mm < 60$.

Voorbeeld in- en uitvoer

Invoer	Uitvoer
2 1 2 30 08:50 (20) 13:10 (30) 2 1 31 09:00 (0) 2 15 10:30 (0) 09:30 (0)	09:30 14:10 10:45 10:45

B Mikado

Probleem

Mikado is een behendigheidsspelletje waarbij je eerst een stapel stokjes op tafel laat vallen. Vervolgens moeten de spelers steeds stokjes oppakken zonder de andere te bewegen. Omdat de stokjes willekeurig op de tafel vallen, komen ze vaak in meerdere losse stapeltjes terecht. We willen graag de groottes van de stapeltjes weten en ook hoe vaak die groottes voorkomen.

Invoer

Op de eerste regel één positief getal: het aantal testgevallen.

Daarna per testgeval:

- Een regel met één positief getal n met $1 \leq n \leq 250$: het aantal mikado stokjes.
- n regels met daarop vier gehele getallen x_1, y_1, x_2 en y_2 met $-100 \leq x_i, y_i \leq 100$: de coördinaten van de twee uiteinden van een stokje. De twee uiteinden hebben niet dezelfde coördinaten.

Uitvoer

Per testgeval:

- Een regel van het volgende formaat:

  $[n_1]$x$[a_1]$ $[n_2]$x$[a_2]$ ... $[n_r]$x$[a_r]$

  Hier is r het aantal verschillende groottes van stapeltjes, zijn de a_i de betreffende groottes en zijn de n_i hoe vaak die groottes voorkomen. De a_i moeten aflopend gesorteerd zijn.

Voorbeeld in- en uitvoer

Invoer	Uitvoer
2 3 0 0 2 0 0 1 2 3 2 1 0 3 4 0 1 0 2 1 0 2 0 1 1 3 3 2 2 4 4	1x2 1x1 1x2 2x1

C Avondvierdaagse

Probleem

Net als ieder jaar wordt in het plaatsje Regendrup de avondvierdaagse gelopen. Dit jaar heeft het in de aanloop van dit wandelevenement wederom behoorlijk hard geregend, waardoor de paden op een groot aantal plaatsen in een modderpoel zijn veranderd. De organisatie, die zich het fiasco van vorig jaar nog levendig herinnert, heeft besloten hier iets aan te doen door grote planken over de modderpoelen heen te leggen. Zij heeft hiervoor een grote hoeveelheid planken van dezelfde lengte beschikbaar. Hoeveel daarvan zijn er nodig om de hele route moddervrij te krijgen?

Invoer

Op de eerste regel één positief getal: het aantal testgevallen.

Daarna per testgeval:

- Een regel met twee getallen n en l met $1 \leq n \leq 10^4$ en $1 \leq l \leq 10^9$: het aantal modderpoelen en de lengte van de planken.
- n regels met daarop twee getallen b_i, e_i met $0 \leq b_i < e_i \leq 10^9$: het begin- en eindpunt van de modderpoel. De modderpoelen staan gesorteerd op beginpunt, zijn niet overlappend, maar kunnen wel aansluitend zijn.

Uitvoer

Per testgeval:

- Een regel met daarop één getal: het aantal planken dat nodig is om de modderpoelen te overdekken.

Voorbeeld in- en uitvoer

Invoer	Uitvoer
2 1 5 0 100 3 10 1 9 20 25 29 37	20 3

D Drugsbende

Probleem

De politie heeft een grote drugsbende opgerold, die bestaat uit k bendeleden. Deze worden allemaal in dezelfde gevangenis geplaatst, die bestaat uit een lange rechte rij cellen. Van deze cellen zijn nog n cellen vrij en deze bevinden zich op de locaties $x_1 \ldots x_n$. Er zijn minstens zoveel cellen vrij als er bendeleden zijn.

De situatie in de gevangenis wordt gevaarlijker naarmate de bendeleden dichter bij elkaar geplaatst worden en daarom willen we dat de afstand tussen de twee dichtstbijzijnde bendeleden zo groot mogelijk wordt. Wat is de grootst mogelijke afstand?

Invoer

Op de eerste regel één positief getal: het aantal testgevallen.

Daarna per testgeval:

- Een regel met twee positieve getallen n en k, met $2 \leq k \leq n \leq 10^5$: het aantal cellen en het aantal bendeleden.
- Een regel met daarop n positieve getallen $x_1, ..., x_n$ met $0 \leq x_i \leq 10^9$: de locaties van de vrije cellen. Deze zijn oplopend gesorteerd.

Uitvoer

Per testgeval:

- Een regel met daarop de grootst mogelijke afstand tussen de twee dichtbijzijnde bendeleden.

Voorbeeld in- en uitvoer

Invoer	Uitvoer
2 4 4 1 2 3 4 5 3 1 3 6 9 10	1 4

E Safari

Probleem

Safari Tours B.V. organiseert safari's in het zonnige Zuid-Afrika. Door de teruglopende economie krijgen ze echter steeds minder klanten en zelfs in die mate dat het voorbestaan van het bedrijf onzeker is. Gelukkig hebben ze hier iets op bedacht. Om extra klanten te trekken krijgen klanten de volgende garantie: als ze tijdens de safari geen leeuw zien, krijgen zij hun geld terug.

Het is natuurlijk prachtig als dit extra klanten trekt, maar het ligt voor de hand dat er niet al te vaak van deze garantieregeling gebruik moet worden gemaakt. Daarom is men op zoek naar de rondrit, waarvoor de kans dat je een leeuw ziet zo groot mogelijk is. Hiervan moet de lengte tussen bepaalde grenzen zitten. Je mag meerdere keren van dezelfde weg gebruik maken en tussentijds langs het beginpunt komen. Tijdens de rit mag niet stilgestaan worden.

Help mee en red Safari Tours B.V.!

Invoer

Op de eerste regel één positief getal: het aantal testgevallen.

Daarna per testgeval:

- Een regel met twee getallen p en q met $1 \leq p \leq q \leq 600$: de minimale en maximale lengte van de safari in minuten.
- Een regel met twee getallen k en w met $1 \leq k \leq 100$ en $0 \leq w \leq 10^4$: het aantal kruispunten en het aantal wegen. Het eerste kruispunt is het begin- en eindpunt van de safari.
- w regels met daarop vier gehele getallen b_i, e_i, l_i en p_i met $1 \leq b_i, e_i \leq k$, $1 \leq l_i \leq 600$ en $0 \leq p_i \leq 100$: de nummers van de kruispunten waar de weg begint en eindigt, de lengte van de weg in minuten en de kans om daar een leeuw te zien in procenten. Wegen kunnen alleen in de aangegeven richting doorlopen worden en er is nooit meer dan één weg tussen twee kruispunten in een gegeven richting.

Uitvoer

Per testgeval:

- Een regel met daarop "onmogelijk" als het onmogelijk is een safari te organiseren die aan de bovengenoemde eisen voldoet en anders één getal: de kans om een leeuw te zien in procenten, afgerond op twee cijfers achter de komma.

Voorbeeld in- en uitvoer

Invoer	Uitvoer
2 9 11 2 2 1 2 5 50 2 1 5 50 35 45 6 13 1 2 5 10 2 1 5 10 1 3 21 10 3 1 21 10 1 5 5 5 5 1 5 5 2 3 15 10 3 2 15 10 3 4 5 5 4 3 5 5 4 5 10 90 5 4 10 90 1 6 25 100	75.00 99.27

F Parkeren

Probleem

Zweinstein heeft een groot probleem met studenten die hun bezems illegaal parkeren. De school heeft een rechthoekige parkeerplaats gemaakt, waarop de bezems geparkeerd moeten worden. De parkeerplaats bestaat uit vakken, waarvan een deel als parkeervak toegewezen is en de rest vrij moet blijven.

Harry Potter en zijn vriendjes hebben hun bezems illegaal geparkeerd op plekken die vrij moeten blijven. Om het schoolterrein klaar te maken voor de Zwerkbalwedstrijd, wil professor Perkamentus alle bezems verplaatsen naar toegewezen parkeervakken. Om dit zo snel mogelijk te laten verlopen, wil hij dat de grootste afstand waarover een bezem verplaatst moet worden, zo klein mogelijk is.

Bezems zijn magische voorwerpen en kunnen daarom door elkaar heen vliegen. Ze moeten uiteindelijk echter wel op verschillende parkeervakken terecht komen. De afstand die een bezem aflegt, is de som van de verplaatsingen in de lengte- en breedterichting van de parkeerplaats.

Wat is de kleinst mogelijke af te leggen afstand voor de bezem die het verst moet vliegen?

Invoer

Op de eerste regel één positief getal: het aantal testgevallen.

Daarna per testgeval:

- Een regel met twee getallen l en b met $1 \leq l, b \leq 250$: de lengte en breedte van de parkeerplaats.
- l regels met b tekens: 'B' voor een bezem, 'P' voor een parkeervak en '.' voor een vak dat leeg moet blijven. Er zijn minstens evenveel parkeervakken als bezems. Verder is er minimaal 1 bezem en zijn er maximaal 50 parkeervakken.

Uitvoer

Per testgeval:

- Eén regel met daarop één positief geheel getal: de afstand afgelegd door de bezem die het verst moet vliegen.

Voorbeeld in- en uitvoer

Invoer	Uitvoer
1 4 5 ...P. ...P. .B... ..B.P	3

G Coalities

Probleem

De EU (Europese Unie) is sinds kort uitgebreid met 10 nieuwe landen tot 25 landen en voor nieuwe voorstellen binnen de EU geldt de volgende procedure.

Elk land binnen de EU mag voor of tegen het voorstel stemmen. Hierbij heeft elk land een vast aantal stemmen dat o.a. afhankelijk is van de economische macht van het land. Het voorstel wordt aangenomen als aan beide van de volgende twee condities voldaan is.

1. Het totaal aantal stemmen voor het voorstel minstens de helft ($\geq 50\%$) bedraagt van het totale aantal stemmen
2. De totale bevolking in de landen die het voorstel steunen minstens 60% ($\geq 60\%$) van de totale bevolking binnen de EU bedraagt.

Een groep landen die gezamenlijk voor een voorstel willen stemmen, wordt een *coalitie* genoemd. Wanneer deze landen door hun gezamenlijke stem een voorstel aan kunnen nemen, wordt dit een *meerderheidscoalitie* genoemd. Als er in een meerderheidscoalitie een land zit, maar deze coalitie zonder dit land geen meerderheidscoalitie meer vormt, wordt dat land *kritisch* voor de coalitie genoemd.

Voor een gegeven aantal landen met elk hun stemmen en hun bevolkingsaantal ligt het aantal meerderheidscoalities en het aantal meerderheidscoalities, waarvoor dat land kritisch is, vast. De verhouding tussen het aantal meerderheidscoalities waarvoor een land kritisch is en het totale aantal meerderheidscoalities is een maat voor de invloed van dat land binnen de EU.

Invoer

Op de eerste regel één positief getal: het aantal testgevallen.

Daarna volgt per testgeval:

- Een regel met één getal n met $1 \leq n \leq 25$: het aantal landen.
- n regels met daarop twee getallen s, b met $1 \leq s \leq 20$ en $1 \leq b \leq 100$: het aantal stemmen en het aantal inwoners van het i^{de} land.

Uitvoer

Per testgeval:

- Een regel met daarop twee getallen: het totale aantal meerderheidscoalities en het aantal meerderheidscoalities waarvoor het eerste land kritisch is.

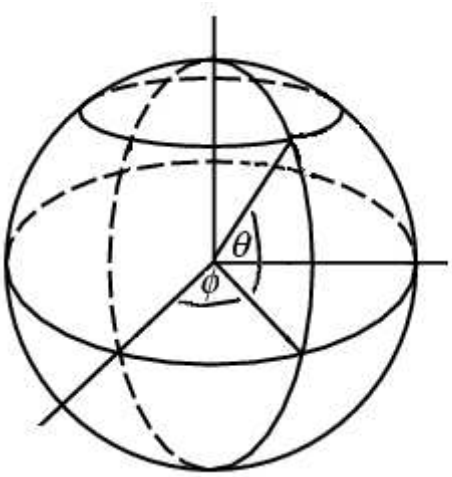
Voorbeeld in- en uitvoer

Invoer	Uitvoer
2 2 1 4 1 6 3 1 2 1 1 2 3	2 0 3 1

H Rondje om de aarde

Probleem

Een groep vrienden wil graag een rondje om de aarde vliegen. Ze hebben een aantal betaalbare vluchten geselecteerd en vragen zich af of het met deze vluchten mogelijk is een één of meerdere keren rond[2] de aarde te vliegen, als ze beginnen in de stad waar ze wonen en daar ook weer uit willen komen.



Steden op aarde worden gegeven door hun breedte- ($-90° < \theta < 90°$) en lengtegraad ($-180° < \phi \leq 180°$). Er liggen niet twee steden op dezelfde positie. Vluchten tussen twee steden lopen altijd over het aardoppervlak via de kortste weg en kunnen beide kanten op genomen worden. Er zijn geen vluchten tussen steden als er niet een unieke korste weg is, of als die kortste weg over de Noord- of Zuidpool zou lopen.

Invoer

Op de eerste regel één positief getal: het aantal testgevallen.

Daarna per testgeval:

- Een regel met één positief getal n met $n \leq 100$: het aantal steden.
- n regels met daarop één string van maximaal 25 letters (namen zijn hoofdlettergevoelig) en twee integers θ en ϕ met $-90 < \theta < 90$ en $-180 < \phi \leq 180$: de naam en de breedte- en lengtegraad van een stad. De eerste stad is de stad waar de vrienden wonen.
- Een regel met één positief getal m met $m \leq 1000$: het aantal mogelijke vluchten.
- m regels met daarop twee strings: de namen van de steden waartussen de vlucht plaatsvindt.

Uitvoer

Per testgeval:

- Een regel met daarop "ja" of "nee", afhankelijk van of het wel of niet mogelijk is een rondje om de aarde te vliegen.

[2]Bij een rondje om de aarde moet je aan het volgende denken: stel je de aardas voor als een oneindig lange stok, leg langs de route een touwtje en knoop de uiteinden aan elkaar. Dan moet het onmogelijk zijn het touwtje van de stok te scheiden (zonder hem door te knippen).

Voorbeeld in- en uitvoer

Invoer	Uitvoer
3 4 Amsterdam 52 5 Londen 51 0 Sydney -35 151 Chicago 42 -87 4 Amsterdam Londen Londen Chicago Londen Sydney Sydney Chicago 3 Amsterdam 52 5 Parijs 48 2 Londen 51 0 3 Amsterdam Parijs Parijs Londen Londen Amsterdam 5 Amsterdam 52 5 Chicago 42 -87 Sydney -35 151 Moskou 55 37 MexicoStad 19 -99 4 Amsterdam Chicago Chicago Sydney Sydney Moskou Moskou MexicoStad	ja nee nee

I Back to BASIC

Probleem

Wie herinnert zich niet die goede oude tijd, dat je je eerste programma schreef. De tijd dat Java nog niet bestond en C veel te moeilijk was, dus ging je in BASIC programmeren. Verder waren begrippen als "object georiënteerd programmeren" en zelfs "procedureel" natuurlijk volkomen onbekend, maar geen nood: het `GOTO` statement doet wonderen.

Om deze mooie tijden te doen herleven, willen we een simpele BASIC interpreter schrijven. Deze moet een (zeer) beperkte subset van de BASIC commando's implementeren, volgens de volgende grammatica:

```
variabele    := 'a' .. 'z'
integer      := '0' .. '65535'
waarde       := <variabele> | <integer>
operator     := '+' | '-' | '*' | '/'
comparator   := '<' | '=' | '>'
test         := <waarde> <comparator> <waarde>
expressie    := <waarde> | <waarde> <operator> <waarde>
statement    := 'PRINT' <waarde> |
                'GOTO' <integer> |
                'END' |
                'IF' <test> 'THEN' <statement> |
                <variabele> '=' <expressie>
```

waarbij alle eenheden gescheiden zullen zijn door een enkele spatie. Als variabelen kunnen dus alleen kleine letters gebruikt worden en integers zijn 16-bits unsigned. De operatoren en comparatoren hebben de gebruikelijke betekenis, waarbij de deling een integerdeling is (d.w.z. alles na de komma wordt afgekapt). Verder doen de commando's het volgende:

`PRINT <waarde>`	Print `waarde` op een regel (gevolgd door een return).
`GOTO <integer>`	Voert het programma verder uit op regel `integer`, waar de regels vanaf 1 genummerd zijn.
`END`	Beëindigt het programma.
`IF <test> THEN <statement>`	Als `test` waar is, wordt `statement` uitgevoerd.
`<variable> = <expressie>`	Kent aan `variabele` de waarde van `expressie` toe.

Van de programma's die uitgevoerd moeten worden, is het volgende gegeven:

- Een variabele waar nog geen waarde aan toegekend is, zal niet als waarde voorkomen.
- De waarden van expressies zullen nooit buiten de gespecificeerde waarde van een integer vallen.
- Er zal nooit door nul gedeeld worden.
- Een GOTO statement bevat altijd een geldig regelnummer.
- De laatste regel van het programma zal "END" zijn (maar er kunnen meer END statements voorkomen).
- Regels zullen niet langer zijn dan 60 tekens.

- Het programma zal eindigen en maximaal 10.000 statements uitvoeren.

Het uitvoeren van het programma begint op regel 1 en eindigt zodra een `END` statement uitgevoerd wordt.

Invoer

Op de eerste regel één positief getal: het aantal testgevallen.
Daarna per testgeval:

- Een regel met één positief getal n met $1 \leq n \leq 200$: het aantal regels programma-code.
- n regels met daarop BASIC-statements die voldoen aan de bovenstaande specificaties.

Uitvoer

Per testgeval:

- De uitvoer zoals die door het BASIC-programma gegenereerd zal worden.

Voorbeeld in- en uitvoer

Invoer	Uitvoer
1 12 p = 2 d = 2 IF d = p THEN GOTO 9 q = p / d q = q * d IF q = p THEN GOTO 10 d = d + 1 GOTO 3 PRINT p p = p + 1 IF p < 99 THEN GOTO 2 END	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

N.B. De jury looft een bonusprijs uit voor degene die het kortste (en korter dan ons) programma kan schrijven (gerekend in karakters) in deze versimpelde BASIC-syntax, dat de bovenstaande output genereert. Dit staat natuurlijk los van het NKP: er hoeft geen kostbare wedstrijdtijd aan besteed te worden.