1 *Opgave A:* Kubisme

Inleiding

Pablo Incasso maakt kubistische kunstvoorwerpen door uit een grote kubus kleinere kubussen te zagen. Hij doet dit op een zeer systematische manier.

Hij schrijft namelijk altijd eerst op welke kubussen hij gaat verwijderen. De lengte van de uit te zagen kubussen t.o.v. de grote kubus is altijd de 1/2 of 1/4, 1/8 enz. Hij gebruikt daarvoor de volgende codes.

l	=	links
r	=	rechts
v	=	voor
a	=	achter
b	=	boven
o	=	onder

Hij geeft eerst l/r dan v/a en dan b/o van de sub-kubussen aan. Hij kan nu dus uitgaande van de oorspronkelijke kubus zeggen welke kleine kubus hij daaruit wil hebben. Bijvoorbeeld: "lvb rab" wil zeggen dat hij van de originele kubus het stuk (links voor boven) neemt en daar weer van het kubusje (rechts achter boven) verwijderd. (Dus alleen die laatste sub-kubus wordt uit de oorspronkelijke kubus verwijderd.) Door meer letterkombinaties achter elkaar te zetten kan hij nog kleinere kubussen verwijderen. Iets als "lvb rab lab" geeft weer een nog kleinere sub-kubus uit het vorige voorbeeld weer.

Pablo wil de oppervlakte van zijn kunstwerken weten om ze te kunnen verven. Bij de simpele objecten ging het nog wel maar bij de laaste ingenieuze super kunstwerken werd het hem teveel.

Opgave

Schrijf een programma dat berekent hoeveel oppervlakte Pablo zal moeten verven. Het blok waarmee hij begint is 512 bij 512 bij 512 cm. Pablo heeft zulk gereedschap dat het kleinste blokje dat hij uit kan zagen 1 x 1 x 1 cm groot is. Pablo werkt op opdracht en moet een kunstwerk voor een bepaalde tijd afhebben hierdoor kan hij niet meer dan 16384 blokjes uitzagen. De heer Incasso is een tovenaar hij kan dus blokjes uitzagen die in het binnenste van de kubus zitten, en omdat hij een artiest is verft hij deze holtes ook nog eens helemaal van binnen. De oppervlakte moet worden gegeven in vierkante cm.

Invoer

Op de eerste regel staat n = het aantal kunstwerken Voor elk kunstwerk komt daarna een getal m = het aantal uitgehakte kubussen. Dan komen er m regels met een uitgezaagde kubus dwz. steeds een aantal keer een drietal letters gescheiden door spaties.

Uitvoer

De uitvoer bestaat uit n regels met daarop de opervlakte in vierkante centimeters dat moet worden geverfd.

Voorbeeld

invoer:

```
2
1
lvo
1
lvo rab
```

uitvoer:

```
1572864
1671168
```

Het eerste voorbeeld is 3*(512*512) + 3*(512*512 - 256*256) + 3*(256*256)
Het tweede voorbeeld is 6*(512*512) + 6*(128*128)

1 *Opgave B:* De volgende combinatie

Inleiding

Twee natuurlijke getallen zijn cijfer-gelijk als elk van de cijfers 1 tot en met 9 even vaak in de decimale representatie voorkomen. 3173 en 1337 zijn bijvoorbeeld cijfer-gelijk, 911 en 1091 ook, maar 123 en 921 niet.

De volgende combinatie van een positief natuurlijk getal is het kleinste cijfer-gelijke getal dat groter is. De volgende combinatie van 123 is 132 en van 911 1019.

Opgave

Uw programma berekent de volgende combinaties voor een rij getallen.

Invoer

Op de eerste regel van de invoer staat het aantal problemen n. Daarna volgen n regels met daarop getallen van maximaal 75 cijfers.

Uitvoer

Voor elk van de n problemen staat in de uitvoer een regel met daarop de decimale representatie (zonder "leading zeroes") van de volgende combinatie van het getal uit de invoer.

Voorbeeld

invoer:

```
3
3173
0123
911
```

uitvoer:

```
3317
132
1019
```

1 *Opgave C:* The color of money

Inleiding

Beleggingsfondsen leveren meestal meer rente op dan bankrekeningen. Ze hebben echter een nadeel; je moet transactie kosten betalen om geld op je fondsrekening te zetten en ook om je geld er weer vanaf te halen.

De transactie kosten zijn evenredig met de hoeveelheid geld dat van rekening naar rekening wordt overgebracht. En de waarde van het beleggingsfonds varieert van dag tot dag.

Indien men een bedrag x wil storten op de rekening van een fonds dan wordt slechts een bedrag van $(1-t)*x$ gebruikt om aandelen te kopen en het bedrag ter grootte van $t*x$ vormt de transactie kosten.

Als men geld wil opnemen uit het fonds dan moet men aandelen terugverkopen aan het beleggingsfonds. Voor de verkoop van x aandelen ontvangt men een bedrag van $x * koers * (1-v)$. Het getal v geeft hier dus de transactiekosten aan en $koers$ is de koers van de aandelen op die dag.

Opgave

Gevraagd wordt om het maximale vermogen dat —met zicht op de toekomst— bereikt kan worden te bepalen. Het vermogen aan het einde van de rit wordt als volgt bepaald: $x + y * koers$, waarbij x het eindsaldo op de bankrekening is y het aantal aandelen in het fonds, en $koers$ de koers op de laatste dag.
Gegeven wordt:

- Begin saldo gewone rekening, $0 \leq s \leq 1000000$. Met twee cijfers achter de komma.

- Aantal aandelen op fonds, $0 \leq a \leq 1000000$. Met twee cijfers achter de komma.

- De transactie-kosten om een aantal aandelen te kopen in het beleggingsfonds, t. Dit is een getal van minimaal 0.0 en maximaal 1.0. Altijd gegeven met èèn getal voor en èèn getal achter de komma.

- De transactie-kosten om geld op te nemen uit het beleggingsfonds, v. Dit is een getal van minimaal 0.0 en maximaal 1.0 Altijd gegeven met èèn getal voor en èèn getal achter de komma.

U kunt in de toekomst kijken en dus de koers van de aandelen voor alle dagen zien. De bank is zo aardig alle tussenresultaten te bewaren, u hoeft dus pas op het laatst als u het kapitaal afdrukt af te ronden op twee decimalen achter de komma. (Deze moet u printen ook al zijn ze nul.) Het is niet nodig een geheel aantal aandelen te kopen/verkopen. U krijgt geen rente op de bankrekening.

Invoer

Op de eerste regel van de input staat één getal met het aantal problemen dat opgelost moet worden. Daarna wordt ieder probleem als volgt beschreven:

Op de eerste regel van een probleem staan de getallen s, a, t, v en n, in die volgorde en gescheiden door spaties. De betekenis voor de getallen s, a, t en v staan in het probleem beschreven. De volgende n regels bestaan dan uit een beschrijving van het koersverloop van het fonds; iedere regel geeft aan hoeveel een aandeel van het fonds waard is per dag. Dit is een geheel getal $1 \leq x \leq 100$.

Uitvoer

Per probleem produceerd u één regel waarop het maximale vermogen op de laatste dag staat, op twee decimalen nauwkeurig.

Voorbeeld

invoer:

hoev. geld
aantal aandele
transactie te verkopen
transactie kosten te kopen

```
2
100.00 0.00 0.6 0.5 3
1
4
3
100.00 50.00 0.4 0.4 4
10
50
5
```

uitvoer:

```
120.00
330.00
```

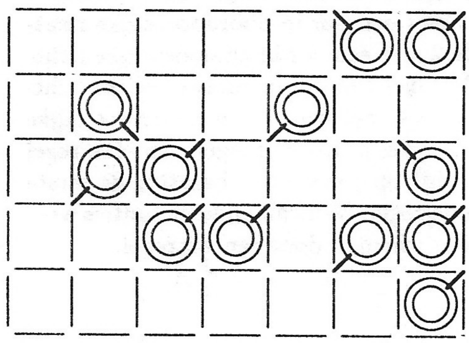
1 *Opgave D:* Cups

Inleiding

Een multinational beschikt in een van zijn vestigingen over een goed uitgeruste kantine annex bedrijfsrestaurant. Daarin hebben de werknemers tussen 12 en 2 uur 's middags de gelegenheid om tegen een schappelijke vergoeding een warme maaltijd te gebruiken. Onmiddellijk na de lunch praten velen van hen onder het genot van een kopje thee of koffie nog wat bij over uiteenlopende onderwerpen.

De thee- en koffiekopjes worden na gebruik afgewassen door het kantinepersoneel. Voor het zover is, worden ze echter eerst door de werknemers zelf op hun kop op een speciaal, rechthoekig rek gezet om uit te druipen. Op dat rek is met behulp van wandjes een rooster van $m \times n$ vierkantjes gecreëerd. In elk vakje past precies één kopje. De oren van de kopjes passen evenwel niet binnen de vierkantjes. Om die reden zijn de hoeken van de vakjes open gelaten, dat wil zeggen: daar lopen de wandjes niet door. In de opening die daardoor ontstaat, is precies ruimte voor het oor van één kopje (zie de onderstaande figuur).

Waar elke opening (behalve die aan de rand) dus grenst aan vier vakjes, kan hoogstens één kopje van de ruimte gebruik maken. Vanzelfsprekend grenzen de openingen op de hoeken van het rek elk aan maar één vakje, en de andere openingen aan de rand van het rek aan maar twee vakjes.



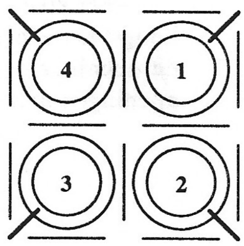
Het is duidelijk dat men, door alle kopjes met het oor in dezelfde richting te zetten, alle vakjes op het rek kan benutten om er een kopje op uit te laten druipen. Helaas gaan de werknemers niet allemaal zo systematisch te werk. Hierdoor kan het voorkomen dat een nog leeg vakje niet zonder meer gevuld kan worden, doordat in alle openingen in de hoeken van dat vakje al een oor van een ander kopje steekt. Dit geldt bijvoorbeeld voor het middelste vakje in het hierboven weergegeven uitdruiprek. Omdat geen enkele werknemer op het idee komt om de kopjes die al op het rek staan zó te draaien of te verplaatsten dat er een opening vrijkomt in een hoek van het lege vakje, zal het betreffende vakje onbezet blijven.

Opgave

Bij een gegeven rek en een gegeven opstelling van al eerder geplaatste kopjes, is het nu de grote vraag of, door een juiste plaatsing van andere kopjes, de nog lege vakjes allemaal benut kunnen worden. Men mag hierbij aannemen dat het aantal kopjes dat nog moet uitdruipen geen beperking vormt; er komen er in principe genoeg om het rek te vullen.

Invoer

Op de eerste regel van de invoer staat een getal x met het aantal instanties van een rek met een opstelling van eerder geplaatste kopjes. Een instantie wordt beschreven door een regel met daarop twee integers: de waardes voor respectievelijk m en n ($0 \leq m, n \leq 15$), gescheiden door spaties; verder door m regels die elk n integers bevatten. Deze n getallen kunnen de waardes 0, 1, 2, 3 en 4 aannemen. Het j-de getal op de i-de regel geeft aan of vakje (i, j) al bezet is door een kopje, en zo ja: in welke richting het oor van het kopje wijst ($1 \leq i \leq m$, $1 \leq j \leq n$).



Een waarde 0 komt overeen met een leeg vakje, en een hogere waarde met een bezet vakje. Een waarde 1 correspondeert met een oor in noordoostelijke richting, een waarde 2 met een oor in zuidoostelijke richting, een waarde 3 met een oor in zuidwestelijke richting en een waarde 4 met een oor in noordwestelijke richting (zie de figuur). De n integers op elke regel volgen onmiddellijk op elkaar en ze bezetten de eerste n plaatsen van de regel. Verschillende instanties worden van elkaar gescheiden door een witregel.

Uitvoer

De uitvoer bestaat uit x regels met indien er een volledige benutting van het uitdruiprek mogelijk is, JA (hoofdletters) en als het rek niet helemaal vol gezet kan worden met kopjes, NEE (hoofdletters) als enige woord op die regel.

Voorbeeld

invoer:

```
3
4 4
```

```
0004
0444
2000
2120
2 2
02
10
5 7
0000041
0200300
0310004
0011031
0000001
```

uitvoer:

```
JA
JA
NEE
```

NB: Het laatste voorbeeld komt overeen met het plaatje in de inleiding.

1 *Opgave E:* Klanken verwisselen

William A. Spooner, een Engelse priester en leraar uit het begin van deze eeuw, had een rare gewoonte om de beginklanken van een woord om te draaien. Hij zei bijvoorbeel "tons of soil" in plaats van "sons of toil". Misschien begon het als een grap, maar na een tijdje kon Spooner er niet meer mee stoppen. Het verwisselen van klanken was een onbewuste dwang voor William geworden. Deze psychologische afwijking staat tegenwoordig bekend als *spoonerism*. En er zijn inmiddels veel verschillende vormen gerapporteerd, het verwisselen van medeklinkers binnen een woord is een van die vormen.

Uw programma zal werken als een dwangmatige *spoonerist* van het laatste type: van elk woord zal het de eerste medeklinker groepen van de eerste twee lettergrepen die met een medeklinker beginnen verwisselen.

Om woorden in lettergrepen te verdelen zullen we gewoon de Nederlandse regels gebruiken, maar we zullen ze hier nog even opnoemen.

De letterparen `ch`, `ij`, `qu` worden behandeld als enkele letters. De letters `a`, `e`, `i`, `ij`, `o`, `u` en `y` zijn klinkers. De anderen, `ch` en `qu` inclusief, zijn medeklinkers. De tabel hieronder definieert de medeklinker paren die uitgesproken kunnen worden. Deze paren worden gevormd door een medeklinker uit de eerste kolom te nemen en die te combineren met een medeklinker uit de tweede kolom op dezelfde regel. Een medeklinkerrij is uitspreekbaar als alle burenparen uitspreekbaar zijn. Voorbeelden: `j`, `gl` en `schr`.

b, c, ch, v	l, r
d	r, w
f	j, l, n, r
g, p	l, n, r
k	l, n, r, w
s	c, ch, f, h, j, k, l, m, n, p, qu, t
t	h, j, r, w
w	h, r
z	w

In een woord definieert elke niet-lege medeklinkerrij w die wordt ingesloten door klinkers een breekpunt tussen twee lettergrepen als volgt. Als w bestaat uit een enkele medeklinker dan ligt het breekpunt tussen de opvolgende klinker en w. Als w uit twee medeklinkers bestaat, dan ligt het breekpunt daartussen. Als w uit drie of meer medeklinkers bestaat, dan wordt w zo verdeeld dat het tweede deel een maximaal "uitspreekbare rij is", dat wil zeggen dat de rij niet meer uitspreekbaar is als er een medeklinker aan wordt toegevoegd. Voorbeelden van deze regels zijn "kat-ten" en "op-schrij-ven".

In- en uitvoer

De invoer van uw programma bestaat uit een text file. Elke invoer regel bevat een woord in kleine letters (minstens 1, hoogstens 79), behalve de laatste regel, die uit een enkele asterisk ('*') bestaat. De uitvoer van uw programma is ook een text file. Het bevat de "spoonerized" versie van de invoer woorden (in dezelfde volgorde), elk woord op een nieuwe regel.

Bijvoorbeeld, de invoer:

```
katten
opschrijven
*
```

produceert de volgende uitvoer:

```
tatken
opvijschren
```

1 *Opgave F:* Making Stupid Windows

Inleiding

Vroeger werden muren in bergrijke omgevingen niet van bakstenen gemaakt, maar van natuursteen. Natuursteen wordt gehouwen uit bergen en dit zorgt ervoor dat de stenen niet allemaal dezelfde vorm hebben. Geen van allen zijn rechthoekig en ze kunnen dus het best beschreven worden door een polynoom. Een muur bestaat dus uit een aantal polynomen, die zo tegen elkaar geplaatst zijn dat ze een redelijk sluitend geheel vormen.

Een boer die nog in een oud huis wordt, wil een extra raam in zijn huis. Daarvoor moet hij dus een aantal natuurstenen uit een muur halen. Hij heeft een rechthoek getekend op de muur tussen de coordinaten (a,b) - (c,d).

De stenen die (gedeeltelijk) in dit rechthoek vallen moeten er dus uit. Dit houdt in dat er ook open ruimte buiten de rechthoek komt.

De opgave

Gevraagd wordt om te bepalen welke stenen er uit de muur gehaald moeten worden.

Invoer

Op de eerste regel van de invoer staat het aantal muren waarin ramen moeten worden gemaakt. Daarna volgt voor iedere muur het volgende:
Op de eerste regel per muur staan de getallen a, b, c, d en n, in die volgorde waarbij n het aantal stenen in de muur is. Daarna volgen n regels met daarop de stenen in de muur. Een steen word gedefineerd door één invoerregel met daarop als eerste het aantal coördinaten van de hoekpunten en vervolgens de coördinaten in de volgorde $x_1 y_1 x_2 y_2$ etcetera. Het beginpunt komt twee maal voor.

Uitvoer

U wordt gevraagd om aan te geven welke stenen er uit de muur worden gehaald. Dit doet u door door per muur één regel weg te schrijven met daarop het nummers van de stenen die weggeschreven worden. De eerste steen die gedefineerd wordt in de invoer file is steen 1. Deze getallen dienen in dezelfde oplopende volgorde te verschijnen.

1 *Opgave H:* Roman dudes...

Inleiding

De oude romeinen waren een ijverig volkje. Ze bouwden van aquaducten tot villa's en van badhuizen tot bruggen. Om deze bouwwerken verantwoord te bouwen moesten de bouwmeesters ontzettend veel rekenwerk verrichten, en daar waren de oude romeinen, met hun onhandige rekensysteem niet zo dol op. Daarom vragen zij aan jullie of jullie misschien een handig programmaatje kunnen schrijven wat dit voor hen uitrekent. De antwoorden moeten weer in romeinse cijfers gegeven worden, anders begrijpen de bouwmeesters het antwoord niet.

Romeinse getallen

Regels:

1. Je gebruikt de volgende basis-letters:

```
I - 1
V - 5
X - 10
L - 50
C - 100
D - 500
M - 1000
```

2. Je schrijft grote letters links en kleinere aflopend naar rechts.
   (B.V. CI = 100 + 1 = 101)

3. Je mag een kleinere letter voor (links van) een grotere zetten, dan wordt hij van zijn onmiddellijke rechter buurman afgetrokken.

   (a) Dit mag echter alleen met letters die 1 of 2 stappen kleiner zijn dan de letter waar ze voor staan.
   (B.V. IX = 10 - 1 = 9, maar 99 is niet IC.)

   (b) Een aftrekking heeft altijd slechts betrekking op twee letters. Dit wil zeggen dat je een aftrekking niet weer van een andere letter mag aftrekken. Je loopt een getal van rechts naar links af. Als je een aftrekking tegenkomt reken je die uit en gaat door naar de volgende letter, vanaf daar begin je als het ware opnieuw met rekenen.
   (B.V. MCMXCV = 1000 + (1000 - 100) + (100 - 10) + 5 = 1995)

   (c) Een aftrekking moet gezien worden als nieuwe letter, waarop wel regel 2 maar geen van de regels van 3 op van toepassing zijn. Zie 3(b).

(B.V. 2995 is NIET MCMMXCV = (1000 - 100) + 1000 + (100 - 10) + 5, want 1000 is groter dan (1000 - 100) en moet daar dus voor komen.)

4. Je gebruikt altijd de kleinste mogelijkheid, dat wil zeggen zo min mogelijk en dan ook nog eens zo klein mogelijke basis-letters.
   (B.V. 1995 is NIET MCMLVL = 1000 + (1000 - 100) + 50 + (50 - 5), want het eerste voorbeeld is "kleiner" - zit geen L in.)

De romeinen kunnen alleen vermenigvuldigen $*$, optellen $+$ en aftrekken $-$. Vermenigvuldigen gaat voor optellen en aftrekken. Het eindresultaat van hun berekeningen is altijd minstens I en maximaal MMMMMMMMMM. Dit zijn ook de maximale getallen die u in de gegeven rekenkundige expressies zult tegenkomen. Er zullen niet meer dan 10 getallen in een expressie staan

Invoer

Op de eerste regel staat het aantal i rekenkundige romeinse expressies die u moet oplossen (natuurlijk weergegeven door een romeins getal). Daarna komen i regels met daarop de expressies. De expressies worden gegeven in romeinse getallen met daartussen de operatoren $*$, $+$ en $-$. Ze worden afgesloten met een punt (.).

Uitvoer

De uitvoer bestaat uit i regels met correcte uitkomsten van de i rekenkundige romeinse expressies gegeven in romeinse getallen.

Voorbeeld

invoer:

```
II
I+I.
MCCXXXIV+DLXVII*II.
```

uitvoer:

```
II
MMCCCLXVIII
```