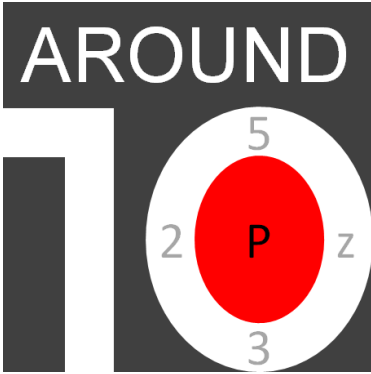


Around ten

Each position of a rectangular grid is either filled with a letter or a digit. Your task is to discover the hidden word that can be found by traversing the grid from left to right, and from top to bottom and only retaining the letters that are protected. We say that a letter is protected if the digits above, below, left and right of the letter add up to ten. Surrounding letters are ignored while computing the sum. Note that according to this definition, letters on the edge and in the corners of the grid can also be protected.



Input

The first line of the input contains a number $\$t \in \mathbb{N}_{>0}\$$ that indicates the number of rows of the rectangular grid. This is followed by another $\$t\$$ lines, each containing the same number of characters. Each character either is a lowercase letter, an uppercase letter or a digit. All these letters form the grid into which a secret word was hidden.

Output

The word that is hidden into the grid.

Example

Input:

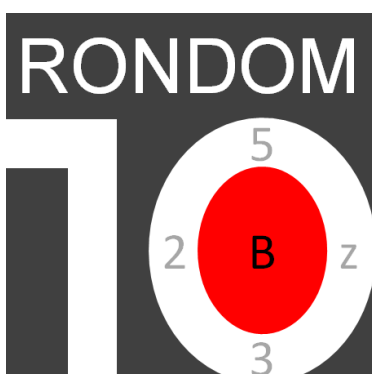
```
48
QYnwOri265ebn2w53ES1X3e3I503uU0Ve9mzD301Gh4X0K4P21sF7M5oMIN11500V2DBn
bUf2JF7n223kNMu7e3WmjyIxJ22jG1K34E0q21442eY3R31B1876B29i200EI95j4Y7iS
en250Z1zJI4N4EF4xYtA41eW2A31552195JtzS5Qcoj4365055wjsEr20Y6K5fA2w6EgS
p4Fxi5i9eY1164H3031gGI09II0j211620115i9JiE0s0qC3UOwHn11XTm2W614uQ2p1SS
H6j0U4i8ry5HY02f158VsuVA6Pv8La53VW1XpjHK1H8e40HLFv217mejb5OLH2EJ40W0u
vRm3Jd450Q55M02320Ri44bmlze8CS12ArOBT00u1Ru3NP2oJd91FQWi05gSX3r3mqMrm
sz36013904Ug7k521P4fd51Z0ZPqE4Z0jp397810214x61MXrdKt0J4635Ke5131Fr3H4
8a0441J2o1pK13Q08p92w0y2lv1AS0434MI31y2II1r2020D21LI34TEM2I19426ReD61
NGcu20314X1Wj2fX11RBnI0QK1054D026Hb2dE05b3sEQ02M6HOhaGdX9CLCuB27nAG3f
odjKO04r072JFW13RI1EW7wdq202exQWO0j9CDgiuEsH3BgUSMYx6sbFT3p3e3NmcE3VE
RKyo4kFO38WYb25112100R823063tO2rDKo8j1WAFxpjZ0Svbt2dq1Rh64g5vq39m6FFm
P413N7bUtd3elU61m072V00543DdgZn1BwNEx6N4LRXW1LPMwSr1GgRNth2n23130439Z
uT4UAYe3752225owJT7337742PGH5eN3zZGNm04Jc0ZEfhc3R1713h5OEpzGa10p15oZ
W7ARq8jYCUv1Ix1pu52Oa0XjheH2ye9r2Pnliz1F5IXh2Is4PIA0PC4qOZWQ1TZ4U4h3p
1f07SjIT1Gm1R3a2hH1tVnmqxWHMYm70zkuHC3N13d25pemI0GlbH2F2P973K1s5d89dm
3K062UEZVK2f3i1fx86b7N64o20y24A6P313TOwX4arVW040dQMY1d0D56Z06EShr1277
4040ePn21pEBEk911P10PB41U427cLUZkp4gS1JDUcZm9064uJi815c2sQR5lyDB31035
```

730A4Rp4h2E66HA0921UdQ92E2YjZAFAlYKV1K1P3230N12fLbKA590j1leaWxg12I1uO
34jfFvR01i2300x77N0x3x0P222RNe5eBLVsg512HJa3OI3vqvIOFu303s3mtxubl7VeO
0vn43fwp8072072u18Nh0D9081730IIJzSu131710EK200arrGh676f5qR5cYZ542d55i
B8rzeFk92t7zw3K2EC1CW001zTn9VkJ8vyDA5b7D06b2nngnKM0i1ev75s57p0i10259u
50WODF055KmdwCN348239Dv66ZbnYt4FdOf1zi39X94H3rWXxw75QJW42s2XPBm626my5
Gi5i24KH00cb7t4d5xeUPuoC7g2064v0S2O2978IAV2jpZBq5nfZ3KLYJx4J412WFP3V5
6S853332520QT1kms55XwgP23go230JSpuUV8r34549xG43Lc1JJRt51Z01u2G0JJcE1X
5LSE3MMH15238iOo5oeN5gjr528T27suz931786209Z003y3cF3B2g2MHI80a26oj9gER
5mtxjmL6w41f1Vld9931DoLE4i22cUG1eeu095Yx01F640K3O0iDBXXMxU1JtZZPYNfAp
31EajL12C4S86w4246033t1O06oXCO326Jkr4qHz2M263V0X5Ov8AU46wo4e27827347i
118jk0AbX61d0383ms267Em43427n37n2213i0Zk9IX12031Jph1O34P7x928xl218bSw
50aW224W7L4673UiXXe1a0lug1683E334RK3ld6m5gz415A6183f39MSAHa06e64fs1v0
430T74Va94iVZY2O2h1H7eQw1x0GCu04vLVyZ0ycE0e208S1Wp900dOopkv63n9f113K0
0600J1Gk3uB1oNH5kMmFJ90zb1h4252vNW7V0cNww4m221pkMH117cf1311030HBRy12
173guvOi1120go2f9Z4XG1tOHhyG4d82IIZ152S0dB8R1tKBrF2ZMO3n07Qf7wzF1RoI9
f86ZxeZ5VY271H01K13CVi3R67s5Vb04MURR73175w27KV5OE1167G22t2yN5sCTo4T1i
t22058CLvby7S66435227qh2O099Q068EBKR201dqvtwSh7cTo1200O7S1SlhtQWXkvO1
230cVyA17RMIG00MYLacJg361020O5RjITKA3iG88E7I2L7Y39I5218Oki451Dh3ZpvLT
3049jQ2ade5vIX25031DJ854XR10QqrPxQKU3452z1ebTs33Wrla3E00YD55Y15e1n323
7x113bU60In3v3Q92rp14431mq7XFiDCLh2002d31p3BW30fjnl6FIVxaC2E1560u3414
6Q75mG1e3mnU0W211022I2043m3aq3w4YF11534578UcM151L3zzlk4Jck14w49Y7S6n3
QFX2Z2r5TTtJHwny50470p2o6C1kUg1a2056QDi8k11nH2532jpXrfdviMr6ea0XT0nzs
1lJm141aVISW517j6w74l2b5gC5tDvF0k33zRR15015mr24f0c3Y1oQ0wdsiCX0sz52Tc
feTO61Hluz8vCg5a2g73232FMsdBnZ4bAYxnA5q1RKn1Ji30yy1mP31ozX6M1i152mAB1
1kxI4047oLpRg040T2HNGB7igDhJU3fbOdKDsN22g4RL8SA2icC30Jp7LEQ04424sb2o3
QJh222zZ2Nm5os56raS5k838634ag7812iIH4v5B6N4I91q14Te31kBZV385Z11My7r3V
LAH627sVV1401Hj4MXqqe21310ExB1dFzXu7eul31joi117461vLJtB9312924132i1k7
Wl8u1G3e1w93mj4Rv2T2g425072Vn6z2XRcz393i3vU0R1k3508R1rs6P2G5K7pi56mDs
PemqCyIm126A0z90p3822B0f280ER9k03we411f2ZK0y4F4t0fvUH5221mEn0FV52095r
1Er3v6O4D38tQSUCPvC7828bwy330F522f2k0zk1Tzk20484amz9deh86ZX200c30nR0p
TGPFUb2QZdDAKg0UdAYs523RSR971CNbAhdql1OgArc8Z74722wpkZ2019Cc76L0QkDt8

Output:

exTENSions

We geven je een rechthoekig rooster waarin op elke positie een letter of een cijfer staat. Je vindt op basis van dit rooster een verborgen woord door het rooster van links naar rechts en van boven naar onder te doorlopen, en alle beschermde letters die je daarbij tegenkomt achter elkaar te zetten. Een letter is beschermd als de som van de cijfers die boven, onder, links en rechts van de letter staan samen som tien hebben. Omliggende letters worden genegeerd bij het berekenen van deze som. Merk op dat volgens deze definitie letters op de rand van het rooster dus evengoed beschermd kunnen zijn.



Invoer

De eerste regel van de invoer bevat een getal $\$t \in \mathbb{N}_0$ dat het aantal rijen van het rechthoekig rooster aangeeft. Daarna volgen t regels die telkens een vast aantal karakters bevatten. Deze karakters kunnen zowel hoofdletters, kleine letters en cijfers zijn. Alle deze karakters vormen samen een rooster waarin een woord verborgen zit dat kan samengesteld worden uit de beschermde letters.

Uitvoer

Het woord dat in het rooster verborgen zit.

Voorbeeld

Invoer:

48

QYnwOri265ebn2w53ES1X3e3I503uU0Ve9mzD301Gh4X0K4P21sF7M5oMIN11500V2DBn
bUf2JF7n223kNMu7e3WmjlxJ22jG1K34E0q21442eY3R31B1876B29i200EI95j4Y7iS
en250Z1zJl4N4EF4xYtA41eW2A31552195JtzS5Qcoj4365055wjsEr20Y6K5fA2w6EgS
p4F5i9eY1164H3031gGl09lI0j211620115i9JiE0s0qC3UOwHn11XTm2W614uQ2p1SS
H6j0U4i8ry5HY02f158VsuVA6Pv8La53VW1XpjHK1H8e40HLFv217mejb5OLH2EJ40W0u
vRm3Jd450Q55M02320Rl44bmlze8CS12ArOBT00u1Ru3NP2oJd91FQWi05gSX3r3mqMrm
sz36013904Ug7k521P4fd51Z0ZPqE4Z0jp397810214x61MXrdKt0J4635Ke5131Fr3H4
8a0441J2o1pK13Q08p92w0y2lv1AS0434MI31y2lI1r2020D21LI34TEM2119426ReD61
NGcu20314X1Wj2fX11RBnl0QK1054D026Hb2dE05b3sEQ02M6HOhaGdX9CLCuB27nAG3f
odjKO04r072JFW13Rl1EW7wdq202exQWO0j9CDgiuEsH3BgUSMYx6sbFT3p3e3NmcE3VE
RKyo4kFO38WYb25112100R823063tO2rDKo8j1WAFxpjZ0Svbt2dq1Rh64g5vq39m6FFm
P413N7bUtd3elU61m072V00543DdgZn1BwNEx6N4LRXW1LPMwSr1GgRNth2n23130439Z
uT4UAYe3752225owJT7337742PGH5eN3zZGNm04Jc0ZEfh3R1713h5OEpzGa10p15oZ
W7ARq8jYCUv11x1pu52Oa0XjheH2ye9r2Pnliz1F5IXh2Is4PIA0PC4qOZWQ1TZ4U4h3p
1f07SjIT1Gm1R3a2hH1tVnmqxWHMYm70zkuHC3N13d25peml0GlbH2F2P973K1s5d89dm
3K062UEZVK2f3i1fx86b7N64o20y24A6P313TOwX4arVW040dQMY1d0D56Z06EShr1277
4040ePn21pEBEk911P10PB41U427cLUZkp4gS1JDUcZm9064uJi815c2sQR5lyDB31035
730A4Rp4h2E66HA0921UdQ92E2YjZAFAlYKV1K1P3230N12fLbKA590j1leaWxg12l1uO
34jFvR01i2300x77N0x3x0P222RNe5eBLVsg512HJa3OI3vqvIOFu303s3mtxubl7VeO
0vn43fwp8072072u18Nh0D9081730lIzSu131710EK200arrGh676f5qR5cYZ542d55i
B8rzeFk92t7zw3K2EC1CW001zTn9VkJ8vyDA5b7D06b2nngnKM0i1ev75s57p0i10259u
50WODF055KmdwCN348239Dv66ZbnYt4FdOf1zi39X94H3rWXxw75QJW42s2XPBm626my5
Gi5i24KH00cb7t4d5xeUPuoC7g2064v0S2O2978IAV2jpZBq5nfZ3KLYJx4J412WFP3V5
6S853332520QT1kms55XwgP23go230JSpuUV8r34549xG43Lc1JJRt51Z01u2G0JJcE1X
5LSE3MMH15238iOo5oeN5gjr528T27suz931786209Z003y3cF3B2g2MHI80a26oj9gER
5mtxjmL6w41f1Vld9931DoLE4i22cUG1eeu095Yx01F640K300iDBXXMxU1JtZZPYNfAp
31EajL12C4S86w4246033t1O06oXCO326Jkr4qHz2M263VOX5Ov8AU46wo4e27827347i
118jk0AbX61d0383ms267Em43427n37n2213i0Zk9IX12031Jph1O34P7x928xI218bSw
50aW224W7L4673UiXXe1a0lug1683E334RK3ld6m5gz415A6183f39MSAHa06e64fs1v0
430T74Va94iVZY2O2h1H7eQw1x0GCu04vLVyZ0ycE0e208S1Wp900dOopkv63n9f1I3K0
0600J1Gk3uB1oNH5kMmFJ90zb1h4252vNW7V0cNww4m221pkMH117cf1311030HBRy12
173guvOi1I20go2f9Z4XG1tOHhyG4d82lIZ152S0dB8R1tKBrF2ZMO3n07Qf7wzF1RoI9
f86ZxeZ5VY271H01K13CVi3R67s5Vb04MURR73175w27KV5OE1167G22t2yN5sCTo4T1i
t22058CLvby7S66435227qh2O099Q068EBKR201dqvtwSh7cTo1200O7S1SIhtQWXkvO1
230cVyA17RMIG00MYLacJg361020O5RjITKA3iG88E7l2L7Y39I52180ki451Dh3ZpvLT
3049jQ2ade5vIX25031DJ854XR10QqrPxQKU3452z1ebTs33Wrla3E00YD55Y15e1n323
7x113bU60ln3v3Q92rp14431mq7XFiDCLh2002d31p3BW30fjnl6FIVxaC2E1560u3414
6Q75mG1e3mnU0W21102212043m3aq3w4YF11534578UcM151L3zzlk4Jck14w49Y7S6n3
QFX2Z2r5TTtJHwny50470p2o6C1kUg1a2056QDi8k11nH2532jpXrfdviMr6ea0XT0nzs
1lJm141aVISW517j6w74l2b5gC5tDvF0k33zRR15015mr24f0c3Y1oQ0wdsiCX0sz52Tc
feTO61Hluz8vCg5a2g73232FMsdbnZ4bAYxnA5q1RKn1Ji30yy1mP31ozX6M1i152mAB1
1kxI4047oLpRg040T2HNGB7igDhJU3fbOdKDsN22g4RL8SA2icC30Jp7LEQ04424sb2o3

QJh222zZ2Nm5os56raS5k838634ag7812iIH4v5B6N4I91q14Te31kBZV385Z11My7r3V
LAH627sVV1401Hj4MXqqe21310ExB1dFzXu7eul31jol117461vLJtB9312924132i1k7
Wl8u1G3e1w93mj4Rv2T2g425072Vn6z2XRcz393i3vU0R1k3508R1rs6P2G5K7pi56mDs
PemqCylm126A0z90p3822B0f280ER9k03we411f2ZK0y4F4t0fvUH5221mEn0FV52095r
1Er3v6O4D38tQSUCPvC7828bwy330F522f2k0zk1Tzk20484amz9deh86ZX200c30nR0p
TGPFUb2QZdDAKg0UdAYs523RSR971CNbAhdql1OgArc8Z74722wpkZ2019Cc76L0QkDt8

Uitvoer:

exTENSions