Tabla de contenido

[1. Chutes and Ladders 7](#_Toc11675054)

[2. Eight 8](#_Toc11675055)

[3. The Die is Cast 10](#_Toc11675056)

[4. Blowing Fuses 11](#_Toc11675057)

[5. Sorting Slides 12](#_Toc11675058)

[6. The dog Task 15](#_Toc11675059)

[7. Parentheses Balance 16](#_Toc11675060)

[8. Coin Change 17](#_Toc11675061)

[9. All Walks of length n from the first node 17](#_Toc11675062)

[10. Dropping Balls 18](#_Toc11675063)

[11. Convex Hull Finding 19](#_Toc11675064)

[Goldbach's Conjecture (II 21](#_Toc11675065)

[12. The Collatz Sequence 21](#_Toc11675066)

[13. How Many Knights 22](#_Toc11675067)

[14. The falling leaves 22](#_Toc11675068)

[15. Date Bugs 23](#_Toc11675069)

[16. Triple Ties: The Organizer's Nightmare 24](#_Toc11675070)

[17. Dividing 25](#_Toc11675071)

[18. S-tree 26](#_Toc11675072)

[19. Adding Reversed Numbers 27](#_Toc11675073)

[20. Copying Books 33](#_Toc11675074)

[21. Glass Beads 34](#_Toc11675075)

[22. Division 34](#_Toc11675076)

[23. The Hamming Distance Problem 36](#_Toc11675077)

[24. Anagrams by Stack 36](#_Toc11675078)

[25. Soundex Indexing 37](#_Toc11675079)

[26. Burrows Wheeler Decoder 38](#_Toc11675080)

[27. The MTM Machine 39](#_Toc11675081)

[28. Numeric Puzzles Again! 40](#_Toc11675082)

[29. 8 Queen Chess Problem 45](#_Toc11675083)

[30. A plug for UNIX 46](#_Toc11675084)

[31. 487-3279 48](#_Toc11675085)

[32. Biorhythms 49](#_Toc11675086)

[33. The Same Game 50](#_Toc11675087)

[34. The Return of the Roman Empire 53](#_Toc11675088)

[35. DNA Sequencing 54](#_Toc11675089)

[36. Fibinary Numbers 56](#_Toc11675090)

[37. Hamiltonian Cycle 62](#_Toc11675091)

[38. Monkeys in a Regular Forest 63](#_Toc11675092)

[39. Indexing 65](#_Toc11675093)

[40. Network Connections 65](#_Toc11675094)

[41. Critical Links 67](#_Toc11675095)

[42. Internet Bandwidth 68](#_Toc11675096)

[43. Page Hopping 70](#_Toc11675097)

[44. Walking on the Safe Side 71](#_Toc11675098)

[45. Water Falls 72](#_Toc11675099)

[46. Continued Fractions 74](#_Toc11675100)

[47. Largest Submatrix 75](#_Toc11675101)

[48. Light and Transparencies 76](#_Toc11675102)

[49. A Multiplication Game 77](#_Toc11675103)

[50. Deciding victory in Go 78](#_Toc11675104)

[51. Lunch in Grid City 78](#_Toc11675105)

[52. Substitution Cipher 79](#_Toc11675106)

[53. Numerical maze 80](#_Toc11675107)

[54. Airline Comparison 81](#_Toc11675108)

[55. Counting Cells in a Blob 82](#_Toc11675109)

[56. Ordering 83](#_Toc11675110)

[57. Factorial Factors 85](#_Toc11675111)

[58. Word Problem 85](#_Toc11675112)

[59. Brick Wall Patterns 86](#_Toc11675113)

[60. Password search 87](#_Toc11675114)

[61. Winterim Backpacking Trip 87](#_Toc11675115)

[62. Re-connecting Computer Sites 88](#_Toc11675116)

[63. TV game 89](#_Toc11675117)

[64. Joana and the Odd Numbers 90](#_Toc11675118)

[65. Jumping Champion 90](#_Toc11675119)

[66. Sunny Mountains 91](#_Toc11675120)

[67. Spreading the News 94](#_Toc11675121)

[68. No more prerequisites, please! 95](#_Toc11675122)

[69. Walking Around Wisely 97](#_Toc11675123)

[70. Integer Sequences from Addition of Terms 98](#_Toc11675124)

[71. Eternal Truths 99](#_Toc11675125)

[72. Number Maze 100](#_Toc11675126)

[73. Checking the N-Queens Problem 101](#_Toc11675127)

[74. Genes 103](#_Toc11675128)

[75. Permutations 104](#_Toc11675129)

[76. Happy Numbers 105](#_Toc11675130)

[77. Popes 106](#_Toc11675131)

[78. Taxicab Numbers 107](#_Toc11675132)

[79. Circular 107](#_Toc11675133)

[80. Kaprekar Numbers 108](#_Toc11675134)

[81. Lemmings Battle! 109](#_Toc11675135)

[82. Round and Round Maze 111](#_Toc11675136)

[83. Many paths, one destination 112](#_Toc11675137)

[84. Su Doku 113](#_Toc11675138)

[85. Diving for gold 115](#_Toc11675139)

[86. Safe Salutations 116](#_Toc11675140)

[87. Product of digits 117](#_Toc11675141)

[88. Celebrity jeopardy 118](#_Toc11675142)

[89. 4 values whose sum is 0 118](#_Toc11675143)

[90. Air Raid 119](#_Toc11675144)

[91. Machine Schedule 120](#_Toc11675145)

[92. Calling Extraterrestrial Intelligence Again 121](#_Toc11675146)

[93. Tiling Up Blocks 122](#_Toc11675147)

[94. The Suspects 122](#_Toc11675148)

[95. The Geodetic Set Problem 123](#_Toc11675149)

[96. A DP Problem 125](#_Toc11675150)

[97. Taxi Cab Scheme 126](#_Toc11675151)

[98. Argus 127](#_Toc11675152)

[99. AGTC 128](#_Toc11675153)

[100. WordFish 129](#_Toc11675154)

[101. Sum of Consecutive Prime Numbers 130](#_Toc11675155)

[102. Sum of different primes 130](#_Toc11675156)

[103. The Bug Sensor Problem 131](#_Toc11675157)

[104. Editor 133](#_Toc11675158)

[105. Digit Counting 134](#_Toc11675159)

[106. The longest constant gene 135](#_Toc11675160)

[107. MODEX 137](#_Toc11675161)

[108. Skyline 138](#_Toc11675162)

[109. RACING 140](#_Toc11675163)

[110. Anti Brute Force Lock 140](#_Toc11675164)

[111. Expert Enough? 142](#_Toc11675165)

[112. Jollybee Tournament 143](#_Toc11675166)

[113. Find Terrorists 143](#_Toc11675167)

[114. Interstar Transport 145](#_Toc11675168)

[115. Sales 146](#_Toc11675169)

[116. Longest Path 147](#_Toc11675170)

[117. Bicoloring 148](#_Toc11675171)

[118. Packing polygons 149](#_Toc11675172)

[119. Carmichael Numbers 150](#_Toc11675173)

[120. Count the Trees 151](#_Toc11675174)

[121. What's Cryptanalysis? 157](#_Toc11675175)

[122. All Roads Lead Where 157](#_Toc11675176)

[123. Where's Waldorf? 159](#_Toc11675177)

[124. Joseph's Cousin 160](#_Toc11675178)

[125. Flip-Flop the Squarelotron 160](#_Toc11675179)

[126. The Never Ending Towers of Hanoi 162](#_Toc11675180)

[127. Reverse and Add 164](#_Toc11675181)

[128. Funny Encryption Method 164](#_Toc11675182)

[129. Shoemaker's Problem 165](#_Toc11675183)

[130. Edit Step Ladders 166](#_Toc11675184)

[131. Divisibility 168](#_Toc11675185)

[132. Jolly Jumpers 169](#_Toc11675186)

[133. Vito's family 169](#_Toc11675187)

[134. The Monocycle 170](#_Toc11675188)

[135. Hartals 172](#_Toc11675189)

[136. The Necklace 173](#_Toc11675190)

[137. Jimmy's Riddles 175](#_Toc11675191)

[138. How many zeros and how many digits 177](#_Toc11675192)

[139. Tell me the frequencies 177](#_Toc11675193)

[140. Knuth's Permutation 178](#_Toc11675194)

[141. Distinct Subsequences 179](#_Toc11675195)

[142. Take the land 183](#_Toc11675196)

[143. Airlines 184](#_Toc11675197)

[144. Gopher II 185](#_Toc11675198)

[145. Marbles 187](#_Toc11675199)

[146. The Problem with the Problem Setter 188](#_Toc11675200)

[147. Generating Fast Sorted Permutation 190](#_Toc11675201)

[148. The Tourist Guide 190](#_Toc11675202)

[149. Longest Match 191](#_Toc11675203)

[150. Bangla Numbers 192](#_Toc11675204)

[151. What is the Median 193](#_Toc11675205)

[152. Light , more light 194](#_Toc11675206)

[153. Loansome Car Buyer 194](#_Toc11675207)

[154. Automatic Editing 195](#_Toc11675208)

[155. Robot Motion 196](#_Toc11675209)

[156. Play on Words 197](#_Toc11675210)

[157. Play on Words 198](#_Toc11675211)

[158. CDVII 199](#_Toc11675212)

[159. Factovisors 200](#_Toc11675213)

[160. Prime Distance 201](#_Toc11675214)

[161. ShellSort 202](#_Toc11675215)

[162. War 203](#_Toc11675216)

[163. Ant on a Chessboard 205](#_Toc11675217)

[164. Meeting Prof. Miguel 205](#_Toc11675218)

[165. Ocean Deep! Make it shallow!! 207](#_Toc11675219)

[166. Count the faces 208](#_Toc11675220)

[167. Bee Maja 208](#_Toc11675221)

[168. Automated Judge Script 209](#_Toc11675222)

[169. Check the Check 211](#_Toc11675223)

[170. Adventures in Moving 214](#_Toc11675224)

[171. Snow Clearing 215](#_Toc11675225)

[172. Stack 'em Up 216](#_Toc11675226)

[173. Is This Integration ? 217](#_Toc11675227)

[174. The Last Non-zero Digit 217](#_Toc11675228)

[175. Satellites 218](#_Toc11675229)

[176. Hardwood Species 219](#_Toc11675230)

[177. Forest 219](#_Toc11675231)

[178. Throw the Dice 220](#_Toc11675232)

[179. Fire! Fire!! Fire!!! 226](#_Toc11675233)

[180. The Grand Dinner 227](#_Toc11675234)

[181. The Other Two Trees 228](#_Toc11675235)

[182. Common Permutation 229](#_Toc11675236)

[183. Adventure of Super Mario 230](#_Toc11675237)

[184. Hanoi Tower Troubles Again! 231](#_Toc11675238)

[185. Boastin' Red Socks 232](#_Toc11675239)

[186. BabelFish 233](#_Toc11675240)

[187. The Kissing Circles 235](#_Toc11675241)

[188. Chessboard in FEN 235](#_Toc11675242)

[189. Longest Run on a Snowboard 238](#_Toc11675243)

[190. Hay Points 239](#_Toc11675244)

[191. Jogging Trails 239](#_Toc11675245)

[192. Power Strings 241](#_Toc11675246)

[193. Relatives 242](#_Toc11675247)

[194. Rings and Glue 242](#_Toc11675248)

[195. e-Coins 243](#_Toc11675249)

[196. Killing Aliens in Borg Maze 244](#_Toc11675250)

[197. Roads in the North 246](#_Toc11675251)

[198. Goldbatch and Euler 247](#_Toc11675252)

[199. Pay the Price 248](#_Toc11675253)

[200. Power Transmision 250](#_Toc11675254)

[201. Mischievous Children 251](#_Toc11675255)

[202. 23 Out of 5 252](#_Toc11675256)

[203. Antenna Placement 253](#_Toc11675257)

[204. Liftless EME 254](#_Toc11675258)

[205. Rat Attack 255](#_Toc11675259)

[206. Tic Tac Toe 256](#_Toc11675260)

[207. Square 257](#_Toc11675261)

[208. Blocks 258](#_Toc11675262)

[209. Election 259](#_Toc11675263)

[210. Maze Traversal 260](#_Toc11675264)

[211. Watering Grass 261](#_Toc11675265)

[212. The One-Handed Typist 262](#_Toc11675266)

[213. Game Show Math 262](#_Toc11675267)

[214. Injured Queen Problem 264](#_Toc11675268)

[215. Simple Division 266](#_Toc11675269)

[216. Die Game 267](#_Toc11675270)

[217. Knights in FEN 268](#_Toc11675271)

[218. Naughty Sleepy Boys 270](#_Toc11675272)

[219. Polygon Inside A Circle 271](#_Toc11675273)

[220. Ferry Loading 271](#_Toc11675274)

[221. Catenyms 272](#_Toc11675275)

[222. Rock, Scissors, Paper 274](#_Toc11675276)

[223. Is There A Second Way Left 275](#_Toc11675277)

[224. Homer Simpson 276](#_Toc11675278)

[225. To Carry or not to Carry 278](#_Toc11675279)

[226. Help the Leaders 278](#_Toc11675280)

[227. Sabotage 279](#_Toc11675281)

[228. Divisibility of Factors 281](#_Toc11675282)

[229. Boxes of Chocolates 282](#_Toc11675283)

[230. Mr. Azad and His Son 283](#_Toc11675284)

[231. Cows and Cars 283](#_Toc11675285)

[232. Collecting Beepers 284](#_Toc11675286)

[233. The dominoes solitaire 285](#_Toc11675287)

[234. Montesco vs Capuleto 285](#_Toc11675288)

[235. Waking up brain 287](#_Toc11675289)

[236. Wavio Sequence 288](#_Toc11675290)

[237. Almost Prime Numbers 289](#_Toc11675291)

[238. Traveling Politician 290](#_Toc11675292)

[239. Combination Lock 291](#_Toc11675293)

[240. XYZZY 292](#_Toc11675294)

[241. Geometry Paradox 293](#_Toc11675295)

[242. Y2K Accounting Bug 294](#_Toc11675296)

[243. Center of symmetry 295](#_Toc11675297)

[244. Area 296](#_Toc11675298)

[245. Happy Number 297](#_Toc11675299)

[246. Morning Walk 298](#_Toc11675300)

[247. ACM contest and Blackout 298](#_Toc11675301)

[248. Friends 300](#_Toc11675302)

[249. Gopher and Hawks 301](#_Toc11675303)

[250. The Playboy Chimp 302](#_Toc11675304)

[251. Divisible Group Sums 303](#_Toc11675305)

[252. Prince and Princess 304](#_Toc11675306)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Chutes and Ladders

\*\*\*ID: 647

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int dice[ 1005 ] , a[ 105 ], pos[ 7 ] , turn[ 7 ];

int loss[ 105 ];

int len = 0 , players , u , v , win;

while( scanf("%d" , &dice[ len ] ) , dice[ len ]){ len++; }

while( scanf("%d" , &players ) , players ){

memset( a , 0 , sizeof( a ) );

for( int i = 0 ; i < 7 ; ++i ){ pos[ i ] = 0; turn[ i ] = 1; }//turn 1 si tiene un turno 2 si tiene 2, 0 si no tiene

while( scanf("%d %d" , &u , &v ) , u | v ){ a[ u ] = v; }

memset( loss , 0 , sizeof( loss ) );

while( scanf("%d" , &u ) , u ){ if( u < 0 )loss[ -u ] = -1; else loss[ u ] = 1; }

win = 0;

for( int i = 0 , p = 0; i < len ; ++i ){

if( !turn[ p ] ){ turn[ p ] = 1; p = ( p + 1 ) % players; } // si no tiene turno continua el otro jugador

if( pos[ p ] >= 95 && ( dice[ i ] + pos[ p ] ) != 100 ){ p = ( p + 1 ) % players; continue;}

pos[ p ] += dice[ i ];

if( a[ pos[ p ] ] ) pos[ p ] = a[ pos[ p ] ];

else{

if( loss[ pos[ p ] ] < 0 ) turn[ p ] = 0;

else if( loss[ pos[ p ] ] )turn[ p ] = 2;

}

if( pos[ p ] == 100 ){ win = p + 1; break;}

if( turn[ p ] != 2 ) p = ( p + 1 ) % players;

else turn[ p ] = 1;

}

printf("%d\n" , win );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Eight

\*\*\*ID: 652

\*\*\*Juez: UVA

\*\*\*Tipo: BFS | IDA\* | A\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//BFS

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 400000

#define SIZE 9

int N = 3;

char direcciones[ 5 ] = { 'l' , 'r' , 'u' , 'd' };

int dx[ 4 ] = { 0 , 0 , 1 , -1 };

int dy[ 4 ] = { 1 , -1 , 0 , 0 };

int final[ SIZE ] = { 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 };

int seen[ MAX ], previous[ MAX ];

string path[ MAX ];

struct Estado{

int d , posVacio , puzzle[ SIZE ] , hash;

string path;

Estado(){}

};

int getId( int x , int y ){ return x \* N + y; }

void print( int hash ){

int i;

for( i = path[ hash ].size() - 1 ; i >= 0 ; --i ) printf("%c" , path[ hash ][ i ] );

printf("\n");

}

int fact[ 10 ] = { 1 , 1 , 2 , 6 , 24 , 120 , 720 , 5040 , 40320 };

//Permutation inversion

int HashValue( int puzzle[] ){

int i , j , inversions, hash = 0;

for( i = 0 ; i < SIZE ; ++i ){

inversions = 0;

for( j = i + 1 ; j < SIZE ; ++j ){

if( puzzle[ i ] > puzzle[ j ] ) inversions++;

}

hash += fact[ SIZE - i - 1 ] \* inversions;

}

return hash;

}

void bfs(){

int i , nx , ny , x , y , nuevoPos , hash , posVacio , hashAct;

memset( seen , 0 , sizeof( seen ) );

queue< Estado > Q;

Estado inicial;

memcpy( inicial.puzzle , final , sizeof( final ) );

inicial.d = 0;

inicial.posVacio = 8;

inicial.hash = 0;

inicial.path = "";

Q.push( inicial );

seen[ 0 ] = 1;

previous[ 0 ] = -1;

while( !Q.empty() ){

Estado act = Q.front(); Q.pop();

posVacio = act.posVacio;

hashAct = act.hash;

for( i = 0 ; i < 4 ; ++i ){

y = posVacio % N;

x = posVacio / N;

nx = dx[ i ] + x;

ny = dy[ i ] + y;

Estado nuevo = act;

if( nx >= 0 && ny >= 0 && nx < N && ny < N ){

nuevoPos = getId( nx , ny );

swap( nuevo.puzzle[ posVacio ] , nuevo.puzzle[ nuevoPos ] );

hash = HashValue( nuevo.puzzle );

if( !seen[ hash ] ){

seen[ hash ] = 1;

previous[ hash ] = hashAct;

nuevo.path += direcciones[ i ];

path[ hash ] = nuevo.path;

nuevo.hash = hash;

nuevo.posVacio = nuevoPos;

Q.push( nuevo );

}

}

}

}

}

int main(){

bfs( );

int casos , i , hash , q;

char c;

scanf("%d" , &casos );

while( casos-- ){

Estado inicial;

for( i = 0 ; i < SIZE ; ++i ){

scanf(" %c" , &c );

if( c == 'x' ){

inicial.puzzle[ i ] = 9;

}

else inicial.puzzle[ i ] = c - '0';

}

inicial.d = 0;

hash = HashValue( inicial.puzzle );

if( seen[ hash ] ){

print( hash );

}

else puts("unsolvable");

if( casos ) printf("\n");

}

return 0;

}

/\*

12

x 8 7 6 5 4 3 2 1

6 8 7 x 5 4 3 2 1

8 x 7 6 5 4 3 2 1

6 8 7 3 5 4 x 2 1

6 8 7 5 x 4 3 2 1

8 5 7 6 x 4 3 2 1

8 7 x 6 5 4 3 2 1

2 1 8 5 4 3 7 6 x

2 1 x 5 6 8 7 3 4

2 1 8 5 x 6 7 3 4

2 4 1 5 6 8 x 7 3

1 7 8 2 5 4 x 6 3

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Die is Cast

\*\*\*ID: 657

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Flood Fill, DFS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <vector>

using namespace std;

vector<string> t;

int a, b, c = 0;

int ffill(int x, int y, char i) {

if(x < 0 || x >= a || y < 0 || y >= b) return 0;

if(t[y][x] == '.') return 0;

if(t[y][x] == i) {

t[y][x] = i == '\*'?'.':'\*';

return ffill(x+1,y,i) + ffill(x-1,y,i) + ffill(x,y+1,i) + ffill(x,y-1,i);

}

if(i == 'X') return 0;

ffill(x,y,'X');

return ffill(x,y,'\*') + 1;

}

void number() {

vector<int> s;

int n[6];

for(int i = 0; i < 6; i++) {

n[i] = 0;

}

for(int i = 0; i < a; i++) {

for(int j = 0; j < b; j++) {

if(t[j][i] == '\*') {

n[ffill(i,j,'\*')-1]++;

}

}

}

cout<<"Throw "<<++c<<endl;

for(int k = 0; k < 6; k++) {

for(int l = 0; l < n[k]; l++) {

s.push\_back(k+1);

}

}

for(int i = 0; i < s.size()-1; i++) cout<<s[i]<<" ";

cout<<s[s.size()-1];

cout<<endl<<endl;

}

int main() {

string line;

while(true) {

cin>>a>>b;

if(!a && !b) break;

for(int i = 0; i < b; i++) {

cin>>line;

t.push\_back(line);

}

number();

t.clear();

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Blowing Fuses

\*\*\*ID: 661

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

using namespace std;

int main(){

int n,m,c,num,cont=1;

bool flag;

while(scanf("%d %d %d",&n,&m,&c)){

if(!n && !m && !c)break;

int a[n+1],maxi=0,sum=0;

bool visitado[n+1];

for(int i=0;i<n;i++){

scanf("%d",&a[i+1]);

visitado[i+1]=false;

}

flag=true;

while(m--){

scanf("%d",&num);

if(!visitado[num]){

sum+=a[num];

visitado[num]=true;

}

else{

sum-=a[num];

visitado[num]=false;

}

maxi=max(maxi,sum);

if(sum>c){flag=false;}

}

printf("Sequence %d\n",cont++);

if(!flag){

printf("Fuse was blown.\n\n");

}

else{

printf("Fuse was not blown.\n");

printf("Maximal power consumption was %d amperes.\n\n", maxi);

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Sorting Slides

\*\*\*ID: 663

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Maximum bipartite matching

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define M 128

#define N 10005

bool graph[M][N];

bool seen[N];

int matchL[M], matchR[N];

int n, m;

bool bpm( int u )

{

for( int v = 0; v < m; v++ ) if( graph[u][v] )

{

if( seen[v] ) continue;

seen[v] = true;

if( matchR[v] < 0 || bpm( matchR[v] ) )

{

matchL[u] = v;

matchR[v] = u;

return true;

}

}

return false;

}

struct Rect{

int xmin , xmax , ymin , ymax;

Rect( int x , int xx ,int y , int yy ): xmin(x), xmax(xx), ymin(y), ymax(yy){}

};

struct Point{

int x, y;

Point( int xx , int yy ): x(xx), y(yy){}

};

bool insideRect(Point p,Point L, Point R){

if(p.x<=R.x && L.x<=p.x && p.y<=R.y && L.y<=p.y){

return true;

}

return false;

}

bool correct;

bool cycle;

void dfs( int x , int origen ){

seen[ x ] = 1 ;

for( int i = 0 ; i < m ; ++i ){

if( graph[ x ][ i ] ){

if( i == origen ){ cycle = true; return;}

if( !seen[ i ] ){

dfs( i , origen );

}

if( cycle ) return;

}

}

}

int main(){

int V, xmin , xmax, ymin , ymax, x ,y, t = 1;

while( scanf("%d" , &V) && V ){

vector<Rect> r;

vector<Point> p;

///letras

for( int i = 0 ; i < V ; ++i ){

scanf("%d %d %d %d",&xmin ,&xmax, &ymin, &ymax );

r.push\_back( Rect( xmin, xmax , ymin , ymax ) );

}

///numeros

for( int i = 0 ; i < V ; ++i ){

scanf("%d %d", &x, &y );

p.push\_back( Point( x , y ) );

}

m = V \* 2;

n = 0;

///formamos grafo para cada slide unimos con cada numero posible

for( int i = 0 ; i < V ; ++i ){

for( int j = 0 ; j < V ; ++j ){

///verificamos si el numero actual esta dentro del rectangulo y creamos enlace

if( insideRect( p[ j ] , Point(r[ i ].xmin , r[ i ].ymin ) , Point( r[ i ].xmax , r[ i ].ymax) ) ){

graph[ i ][ j + V ] = 1;

n++;

}

}

}

memset( matchL, -1, sizeof( matchL ) );

memset( matchR, -1, sizeof( matchR ) );

int cnt = 0;

for( int i = 0; i < m; i++ ){

memset( seen, 0, sizeof( seen ) );

if( bpm( i ) ) cnt++;

}

correct = false;

bool first = true;

printf("Heap %d\n", t++ );

if( cnt == V ){

for( int i = 0 ; i < V ;++i ){

graph[ i ][ matchL[ i ] ] = 0;

graph[ matchL[ i ] ] [ i ] = 1;

}

///Recorremos con dfs y vemos posible ciclos

for( int i = 0 ; i < V ;++i ){

memset( seen, 0, sizeof( seen ) );

cycle = false;

dfs( i , i );

if( !cycle ){

correct = true;

if( !first ) printf(" ");

printf("(%c,%d)", i + 'A' , matchL[ i ] - V + 1 );

first = false;

}

}

}

if( !correct )printf("none");

printf("\n\n");

memset( graph, 0 ,sizeof( graph ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The dog Task

\*\*\*ID: 670

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Maximum bipartite matching

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cmath>

#include <cstring>

#define MAX 105

int bob[ MAX ][ 2 ], dog[ MAX ][ 2 ];

#define M 105

#define N 105

bool graph[M][N];

bool seen[N];

int matchL[M], matchR[N];

int b,d;

bool bpm( int u )

{

for( int v = 0; v < d; v++ ) if( graph[u][v] )

{

if( seen[v] ) continue;

seen[v] = true;

if( matchR[v] < 0 || bpm( matchR[v] ) )

{

matchL[u] = v;

matchR[v] = u;

return true;

}

}

return false;

}

double dist( int x , int y , int x1 , int y1 ){

return sqrt( (x - x1) \* ( x - x1 ) + ( y - y1 ) \* ( y - y1 ) );

}

int main(){

int t , g;

double len , da , db;

scanf( "%d" ,&t );

for( int q = 0 ; q < t ; ++q ){

if( q )printf("\n");

scanf("%d %d" , &b , &d );

for( int i = 0 ; i < b ; ++i ){

scanf("%d %d" , &bob[ i ][ 0 ] , &bob[ i ][ 1 ] );

}

for( int i = 0 ; i < d ; ++i ){

scanf("%d %d" , &dog[ i ][ 0 ] , &dog[ i ][ 1 ] );

}

for( int i = 0 ; i < b - 1 ; ++i ){

for( int j = 0 ; j < d ; ++j ){

len = dist( bob[ i ][ 0 ] , bob[ i ][ 1 ] , bob[ i + 1 ][ 0 ] , bob[ i + 1 ][ 1 ] );

da = dist( bob[ i ][ 0 ] , bob[ i ][ 1 ] , dog[ j ][ 0 ] , dog[ j ][ 1 ] );

db = dist( bob[ i + 1 ][ 0 ] , bob[ i + 1 ][ 1 ] , dog[ j ][ 0 ] , dog[ j ][ 1 ] );

if( da + db < 2 \* len ) {

graph[ i ][ j ] = 1;

}

}

}

memset( matchL, -1, sizeof( matchL ) );

memset( matchR, -1, sizeof( matchR ) );

int cnt = 0;

for( int i = 0; i < b; i++ )

{

memset( seen, 0, sizeof( seen ) );

if( bpm( i ) ) cnt++;

}

printf("%d\n" , cnt + b );

for( int i = 0 ; i < b ; ++i ){

if( i > 0 )printf(" ");

printf("%d %d" , bob[ i ][ 0 ] , bob[ i ][ 1 ]);

if( matchL[ i ] != -1 )printf(" %d %d" , dog[ matchL[ i ] ][ 0 ] , dog[ matchL[ i ] ][ 1 ] );

}

printf("\n");

memset( graph , 0 , sizeof( graph ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Parentheses Balance

\*\*\*ID: 673

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, Stacks

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stack>

using namespace std;

#define MAX 130

char s[ MAX ];

bool possible(){

stack< char > S;

for( int i = 0 ; s[ i ] ; ++i ){

if( s[ i ] == '(' || s[ i ] == '[' ) S.push( s[ i ] );

else{

if( !S.empty() ){

char c = S.top(); S.pop();

if( ( s[ i ] == ')' && c != '(' )|| ( s[ i ] == ']' && c != '[' ) ) return false;

}

else return false;

}

}

if( S.size() ) return false;

return true;

}

int main(){

int t;

scanf("%d" , &t );

gets( s );

while( t-- ){

gets( s );

if( possible() ) puts("Yes");

else puts("No");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Coin Change

\*\*\*ID: 674

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

using namespace std;

int tabla[7500];

int coin[6]={1,5,10,25,50};

void Coin(){

tabla[0]=1;

int i,j,c;

for(i = 0; i<=4; i++) {

c = coin[i];

for(j = c; j<=7500; j++) {

tabla[j] += tabla[j-c];

}

}

}

int main(){

int n;

Coin();

while(scanf("%d",&n)!=EOF){

printf("%d\n",tabla[n]);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# All Walks of length n from the first node

\*\*\*ID: 677

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 16

int ady[ MAX ][ MAX ];

bool seen[ MAX ];

int n, sz;

int ans[ MAX ];

bool found;

void dfs( int len , int x ){

ans[ len ] = x;

if( len == sz ){

printf("(");

for( int i = 0 ; i <= sz ; ++i ){

if( i ) printf(",");

printf("%d" , ans[ i ] );

}

printf(")\n");

found = true;

return;

}

seen[ x ] = 1;

for( int i = 1 ; i <= n ; ++i ){

if( ady[ x ][ i ] && !seen[ i ] ){

dfs( len + 1 , i );

}

}

seen[ x ] = 0;

}

int main(){

int x;

int b = 0;

while( scanf("%d %d" , &n , &sz ) != EOF ){

for( int i = 1 ; i <= n ; ++i )

for( int j = 1 ; j <= n ; ++j )

scanf("%d", &ady[ i ][ j ] );

found = false;

memset( seen , 0 , sizeof( seen ) );

if( b != 0 )printf("\n");

dfs( 0 , 1 );

if(!found )printf("no walk of length %d\n" , sz );

memset( ady , 0 , sizeof( ady ) );

scanf("%d" , &b );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Dropping Balls

\*\*\*ID: 679

\*\*\*Juez: UVA

\*\*\*Tipo: Tree

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 524290

int n;

bool tree[ 4 \* MAX ];

int ans[ 25 ][ 1<<20 ];

int solve( int ball , int left = 0 , int right = n - 1 , int node = 0 , int depth = 1 ){

ans[ depth ][ ball ] = node + 1;

if( left == right ){

return node;

}

int mid = (left + right)>>1, ans;

if( tree[ node ] ){

tree[ node ] = 0;

ans = solve( ball , mid + 1 , right , 2 \* node + 2 , depth + 1 );

}

else{

tree[ node ] = 1;

ans = solve( ball , left , mid , 2 \* node + 1 , depth + 1 );

}

return ans;

}

void init( ){

for( int i = 0 ; i < 4 \* MAX ; ++i ) tree[ i ] = 0;

}

int main(){

int t , d , ball;

init();

n = 1<<19;

for( int i = 1 ; i <= 1<<19 ; ++i ) solve( i );

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &d , &ball );

printf("%d\n" , ans[ d ][ ball ] );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Convex Hull Finding

\*\*\*ID: 681

\*\*\*Juez: UVA

\*\*\*Tipo: Geometry, Convex Hull

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <algorithm>

#include <vector>

#include<iostream>

using namespace std;

struct Point{

Point(int x1,int y1){

x=x1;

y=y1;

}

Point(){

}

int x;

int y;

///PRODUCTO PUNTO

int operator \*(const Point &p1)const{

return x\*p1.x+y\*p1.y;

}

///PRODUCTO CRUZ

int operator ^(const Point &p1)const{

return x\*p1.y-y\*p1.x;

}

Point operator-(const Point &p1)const{

Point a(x-p1.x,y-p1.y);

return a;

}

bool operator<(const Point &p1) const{

return (y<p1.y || (y==p1.y && x<p1.x));

}

};

///Direccion de vectores

///1 Sentido Antihorario

///-1 Sentido Horario

///Puntos Colineales(si todos los puntos forman una recta)

int ccw(const Point &O, const Point &A, const Point &B)

{

return (A.x - O.x) \* (B.y - O.y) - (A.y - O.y) \* (B.x - O.x);

}

///CONVER HULL CON ALGORITMO MONOTONE CHAIN

///ORDENA PUNTOS Y RECORRE PARTE SUPERIOR EH INFERIOR

vector<Point> monotoneChain(vector<Point> P){

int n=P.size(),k=0;

vector<Point> H(2\*n);

//Ordenamos los puntos lexicograficamente

sort(P.begin(),P.end());

//Construimos convex hull limite inferior

for(int i=0;i<n;i++){

while(k>=2 && ccw(H[k-2],H[k-1],P[i])<=0)k--;

H[k++]=P[i];

}

//Construimos convex hull limite superior

for(int i=n-2, t=k+1;i>=0;i--){

while(k>=t && ccw(H[k-2],H[k-1],P[i])<=0)k--;

H[k++]=P[i];

}

H.resize(k);

return H;

}

int main(){

int casos,puntos,x,y;

cin>>casos;

cout<<casos<<endl;

while(casos--){

cin>>puntos;

vector<Point> v;

for(int i=0;i<puntos;i++){

cin>>x>>y;

Point p(x,y);

v.push\_back(p);

}

if(casos)cin>>puntos;

vector<Point> pp=monotoneChain(v);

cout<<pp.size()<<endl;

for(int i=0;i<pp.size();i++){

cout<<pp[i].x<<" "<<pp[i].y<<endl;

}

if(casos)cout<<-1<<endl;

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Goldbach's Conjecture (II)

\*\*\*ID: 686

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory, sieve

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

using namespace std;

#define MAX 35000

bool primes[MAX];

void sieve(){

primes[0]=false;

for(int i=1;i<MAX;i++) primes[i]=true;

for(int i=2;i\*i<=MAX;i++){

if(primes[i]){

for(int j=i;j\*i<=MAX;j++){

primes[j\*i]=false;

}

}

}

}

int main(){

int n,i,j,resp;

sieve();

while(scanf("%d",&n) && n!=0){

resp=0;

for(i=2;i<=n/2;i++){

if(!primes[i])continue;

if(primes[i] && primes[n-i]){resp++;}

}

printf("%d\n",resp);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Collatz Sequence

\*\*\*ID: 694

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

using namespace std;

int main(){

long long a,b,i;

int cont,caso=1;

while(cin>>a>>b && a>=0 && b>=0){

cont=1;

i=a;

while(i!=1){

if(i%2==0){i=i/2;}

else {i=3\*i+1;}

if(i>b)break;

cont++;

}

cout<<"Case "<<caso++<<": A = "<<a<<", limit = "<<b<<", number of terms = "<<cont<<"\n";

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# How Many Knights

\*\*\*ID: 696

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main(){

int n , m, ans, aux;

while( scanf("%d %d" , &m , &n) , m | n ){

if( m > 2 && n > 2 ) ans = ( ( m \* n ) + 1 ) /2 ;

else if( m == 1 || n == 1 ) ans = m \* n;

else if( m == 2 ){

aux = n - 3;

if( aux < 0 ) aux = 0;

else aux = ( aux / 4 ) + 1;

ans = 2 \* ( n - ( n/4 ) - aux );

}

else if( n == 2 ){

aux = m - 3;

if( aux < 0 ) aux = 0;

else aux = ( aux / 4 ) + 1;

ans = 2 \* ( m - ( m/4 ) - aux );

}

printf("%d knights may be placed on a %d row %d column board.\n" , ans, m , n );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The falling leaves

\*\*\*ID: 699

\*\*\*Juez: UVA

\*\*\*Tipo: Tree traversal, recursion

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 1005

int tree[ MAX ];

void Solve( int pos , int value ){

tree[ pos ] += value;

int n;

///hijo izquierdo

scanf("%d",&n);

if( n != -1 ) Solve( pos - 1 , n );

///hijo derecho

scanf("%d",&n);

if( n != -1 ) Solve( pos + 1 , n );

return;

}

int main(){

int n, t = 1;

bool first;

while( scanf("%d" ,&n) ){

if( n == -1 )break;

memset( tree, 0 , sizeof( tree ) );

Solve( 500 , n );

first = true;

printf("Case %d:\n", t++);

for( int i = 0 ; i < MAX; ++i ){

if( tree[ i ] ){

if( first ){

printf( "%d" , tree[ i ] );

first = false;

}

else{

printf(" %d", tree[ i ] );

}

}

}

printf("\n\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Date Bugs

\*\*\*ID: 700

\*\*\*Juez: UVA

\*\*\*Tipo: Simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

#include <cstring>

#include <stdlib.h>

#include <cctype>

#include <cmath>

#include <cstdio>

#include <cstdlib>

#include <ctime>

#include <functional>

#include <numeric>

#include <utility>

#include <deque>

#include <stack>

#include <bitset>

#include <map>

#include <set>

#include <string>

#include <vector>

#include <queue>

#include <limits>

#include <fstream>

#include <list>

#include <sstream>

#include <iostream>

#include <iomanip>

using namespace std;

#define MAX 10000

int main(){

int q = 1 , n, C[ MAX ] , a[ MAX ] , b[ MAX ], maxi ;

int seen[ MAX ];

while( scanf("%d" , &n ) , n ){

maxi = 0;

for( int i = 0 ; i < n ; ++i ){

scanf("%d %d %d" , &C[ i ] , &a[ i ] , &b[ i ]);

if( a[ i ] > maxi ){

maxi = a[ i ];

}

}

int op = maxi, cnt;

bool bb = false;

printf("Case #%d:\n" , q++);

while( op < MAX ){

cnt = 0;

for( int i = 0 ; i < n ; ++i ){

int op2 = ( ( op - a[ i ] ) % ( b[ i ] - a[ i ] ) );

if( ( op2 == ( C[ i ] - a[ i ] ) ) ){

cnt++;

}

else break;

}

if( cnt == n ){

bb = true;

printf("The actual year is %d.\n" , op );

break;

}

op++;

}

if( !bb )puts("Unknown bugs detected.");

putchar('\n');

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Triple Ties: The Organizer's Nightmare

\*\*\*ID: 703

\*\*\*Juez: UVA

\*\*\*Tipo: Complete Search

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

#define MAXN 50005

int n, ady[ MAX ][ MAX ], cnt;

int a[ MAXN ] , b[ MAXN ], c[ MAXN ];

int main(){

while( scanf("%d" , &n ) == 1 ){

for( int i = 0 ; i < n ; ++i )

for( int j = 0 ; j < n ;++j )

scanf("%d" , &ady[ i ][ j ] );

cnt = 0;

for( int i = 0 ; i < n ; ++i ){

for( int j = 0 ; j < n ; ++j ){

for( int k = 0 ; k < n ; ++k ){

//cycle

if( ady[ i ][ j ] && ady[ j ][ k ] && ady[ k ][ i ] ){

if( (i > j && j > k) || ( i < j && j < k ) ){

a[ cnt ] = i + 1, b[ cnt ] = j + 1, c[ cnt++ ] = k + 1;

}

}

else if( !ady[ i ][ j ] && !ady[ j ][ k ] && !ady[ k ][ i ] && !ady[ j ][ i ] && !ady[ k ][ j ] && !ady[ i ][ k ]){

if( ( i < j && j < k ) ){

a[ cnt ] = i + 1, b[ cnt ] = j + 1, c[ cnt++ ] = k + 1;

}

}

}

}

}

printf("%d\n" , cnt );

for( int i = 0 ; i < cnt ; ++i )

printf("%d %d %d\n" , a[ i ] , b[ i ] , c[ i ] );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Dividing

\*\*\*ID: 711

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

bool dp[120000/2+1];

int main(){

int a[ 7 ], sum , q = 1 , j;

while( 1 ){

scanf("%d %d %d %d %d %d", &a[ 1 ] , &a[ 2 ], &a[ 3 ] , &a[ 4 ], &a[ 5 ] , &a[ 6 ] );

sum = 0;

memset( dp , 0 , sizeof( dp ) );

for( int i = 1 ; i < 7 ; ++i ) sum += ( a[ i ] \* i );

if( sum == 0 )break;

printf("Collection #%d:\n" , q++ );

if( sum & 1 ) printf("Can't be divided.\n\n");

else{

sum /= 2;

for( int i = 0 ; i < 7 ; ++i ){

dp[ i ] = 0;

}

dp[ 0 ] = 1;

for( int k = 1 ; k < 7 ; ++k ){

for( j = 1 ; j \* 2 - 1 <= a[ k ] ; j <<=1 )

for( int i = sum ; i >= j \* k ; --i ) dp[ i ] |= dp[ i - j \* k ];

j = a[ k ] - ( j - 1 );

if( j > 0 ){

for( int i = sum ; i >= j \* k ; --i ){

dp[ i ] |= dp[ i - j \* k ];

}

}

}

if( dp[ sum ] == 1 ) printf("Can be divided.\n\n");

else printf("Can't be divided.\n\n");

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# S-tree

\*\*\*ID: 712

\*\*\*Juez: UVA

\*\*\*Tipo: Tree, Tree traversal

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAXN 2000

#define MAX 2001

char tree[ MAX ], leafs[ MAX ];

int depth , pos;

void buildTree( int dp , int nodo ){

if( dp == depth ){

tree[ nodo \* 2 + 1 ] = leafs[ pos++ ];

tree[ nodo \* 2 + 2 ] = leafs[ pos++ ];

}

else{

//hijo izq

tree[ nodo \* 2 + 1 ] = '0';

buildTree( dp + 1 , nodo \* 2 + 1 );

//hijo der

tree[ nodo \* 2 + 2 ] = '0';

buildTree( dp + 1 , nodo \* 2 + 2 );

}

}

char dfs( int nodo , char \*s ){

int l = strlen( s );

for( int i = 0 ; i < l ; ++i ){

if( s[ i ] == '0' ){

nodo = 2 \* nodo + 1;

}

else nodo = 2 \* nodo + 2 ;

}

return tree[ nodo ];

}

int main(){

char s[ MAX ];

int Q, q = 1;

while( scanf("%d" ,&depth ) && depth ){

for( int i = 0 ; i < depth ; ++i )scanf("%s", &s );

scanf("%s" , &leafs );

tree[ 0 ] = '0';

pos = 0;

buildTree( 1 , 0 );

printf("S-Tree #%d:\n" , q++ );

scanf("%d" , &Q );

while( Q-- ){

scanf("%s" , &s );

printf("%c" , dfs( 0 , s ) );

}

printf("\n\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Adding Reversed Numbers

\*\*\*ID: 713

\*\*\*Juez: UVA

\*\*\*Tipo: ad hoc, bignum

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

const int MAXD = 500, DIG = 9, BASE = 1000000000;

const unsigned long long BOUND = numeric\_limits <unsigned long long> :: max () - (unsigned long long) BASE \* BASE;

struct bignum

{

int D, digits [MAXD / DIG + 2];

inline void trim ()

{

while (D > 1 && digits [D - 1] == 0)

D--;

}

inline void init (long long x)

{

memset (digits, 0, sizeof (digits));

D = 0;

do

{

digits [D++] = x % BASE;

x /= BASE;

}

while (x > 0);

}

inline bignum (long long x)

{

init (x);

}

inline bignum (int x = 0)

{

init (x);

}

inline bignum (char \*s)

{

memset (digits, 0, sizeof (digits));

int len = strlen (s), first = (len + DIG - 1) % DIG + 1;

D = (len + DIG - 1) / DIG;

for (int i = 0; i < first; i++)

digits [D - 1] = digits [D - 1] \* 10 + s [i] - '0';

for (int i = first, d = D - 2; i < len; i += DIG, d--)

for (int j = i; j < i + DIG; j++)

digits [d] = digits [d] \* 10 + s [j] - '0';

trim ();

}

inline char \*str ()

{

trim ();

char \*buf = new char [DIG \* D + 1];

int pos = 0, d = digits [D - 1];

do

{

buf [pos++] = d % 10 + '0';

d /= 10;

}

while (d > 0);

reverse (buf, buf + pos);

for (int i = D - 2; i >= 0; i--, pos += DIG)

for (int j = DIG - 1, t = digits [i]; j >= 0; j--)

{

buf [pos + j] = t % 10 + '0';

t /= 10;

}

buf [pos] = '\0';

return buf;

}

inline bool operator < (const bignum &o) const

{

if (D != o.D)

return D < o.D;

for (int i = D - 1; i >= 0; i--)

if (digits [i] != o.digits [i])

return digits [i] < o.digits [i];

return false;

}

inline bool operator == (const bignum &o) const

{

if (D != o.D)

return false;

for (int i = 0; i < D; i++)

if (digits [i] != o.digits [i])

return false;

return true;

}

inline bignum operator << (int p) const

{

bignum temp;

temp.D = D + p;

for (int i = 0; i < D; i++)

temp.digits [i + p] = digits [i];

for (int i = 0; i < p; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum operator >> (int p) const

{

bignum temp;

temp.D = D - p;

for (int i = 0; i < D - p; i++)

temp.digits [i] = digits [i + p];

for (int i = D - p; i < D; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum range (int a, int b) const

{

bignum temp = 0;

temp.D = b - a;

for (int i = 0; i < temp.D; i++)

temp.digits [i] = digits [i + a];

return temp;

}

inline bignum operator + (const bignum &o) const

{

bignum sum = o;

int carry = 0;

for (sum.D = 0; sum.D < D || carry > 0; sum.D++)

{

sum.digits [sum.D] += (sum.D < D ? digits [sum.D] : 0) + carry;

if (sum.digits [sum.D] >= BASE)

{

sum.digits [sum.D] -= BASE;

carry = 1;

}

else

carry = 0;

}

sum.D = max (sum.D, o.D);

sum.trim ();

return sum;

}

inline bignum operator - (const bignum &o) const

{

bignum diff = \*this;

for (int i = 0, carry = 0; i < o.D || carry > 0; i++)

{

diff.digits [i] -= (i < o.D ? o.digits [i] : 0) + carry;

if (diff.digits [i] < 0)

{

diff.digits [i] += BASE;

carry = 1;

}

else

carry = 0;

}

diff.trim ();

return diff;

}

inline bignum operator \* (const bignum &o) const

{

bignum prod = 0;

unsigned long long sum = 0, carry = 0;

for (prod.D = 0; prod.D < D + o.D - 1 || carry > 0; prod.D++)

{

sum = carry % BASE;

carry /= BASE;

for (int j = max (prod.D - o.D + 1, 0); j <= min (D - 1, prod.D); j++)

{

sum += (unsigned long long) digits [j] \* o.digits [prod.D - j];

if (sum >= BOUND)

{

carry += sum / BASE;

sum %= BASE;

}

}

carry += sum / BASE;

prod.digits [prod.D] = sum % BASE;

}

prod.trim ();

return prod;

}

inline double double\_div (const bignum &o) const

{

double val = 0, oval = 0;

int num = 0, onum = 0;

for (int i = D - 1; i >= max (D - 3, 0); i--, num++)

val = val \* BASE + digits [i];

for (int i = o.D - 1; i >= max (o.D - 3, 0); i--, onum++)

oval = oval \* BASE + o.digits [i];

return val / oval \* (D - num > o.D - onum ? BASE : 1);

}

inline pair <bignum, bignum> divmod (const bignum &o) const

{

bignum quot = 0, rem = \*this, temp;

for (int i = D - o.D; i >= 0; i--)

{

temp = rem.range (i, rem.D);

int div = (int) temp.double\_div (o);

bignum mult = o \* div;

while (div > 0 && temp < mult)

{

mult = mult - o;

div--;

}

while (div + 1 < BASE && !(temp < mult + o))

{

mult = mult + o;

div++;

}

rem = rem - (o \* div << i);

if (div > 0)

{

quot.digits [i] = div;

quot.D = max (quot.D, i + 1);

}

}

quot.trim ();

rem.trim ();

return make\_pair (quot, rem);

}

inline bignum operator / (const bignum &o) const

{

return divmod (o).first;

}

inline bignum operator % (const bignum &o) const

{

return divmod (o).second;

}

inline bignum power (int exp) const

{

bignum p = 1, temp = \*this;

while (exp > 0)

{

if (exp & 1) p = p \* temp;

if (exp > 1) temp = temp \* temp;

exp >>= 1;

}

return p;

}

};

char\* strrev(char \*s)

{

int len = strlen(s);

for (int i = 0; i < len / 2; ++i) {

std::swap(s[i], s[len - i - 1]);

}

return s;

}

int main(){

int t;

scanf("%d",&t);

char n1[MAXD],n2[MAXD];

while( t -- ){

scanf("%s %s",&n1,&n2);

strcpy( n1 , strrev(n1) );

strcpy( n2 , strrev(n2) );

strcpy( n1 , (bignum(n1) + bignum(n2)).str() );

printf("%s\n",bignum(strrev(n1)).str());

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Copying Books

\*\*\*ID: 714

\*\*\*Juez: UVA

\*\*\*Tipo: Binary Search + Greedy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 505

int N , k , slash[ MAX ];

long long a[ MAX ];

long long maximo( long long a , long long b ){ return ( a > b )? a : b ; }

long long Partition( long long mid ){

int j;

long long max = 0 , sum = 0;

for( int i = N - 1, j = 0 ; i >= 0 ; --i ){

if( j + 1 < k && ( i + 1 < k - j || sum + a[ i ] > mid) ){

max = maximo( max, sum );

sum = a[ i ];

slash[ i ] = 1;

j++;

}

else sum += a[ i ];

}

return maximo( max , sum );

}

int main(){

int t ;

long long left , right , mid;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &N , &k );

left = right = 0;

for( int i = 0 ; i < N ; ++i ) {

scanf("%lld" , &a[ i ] );

right += a[ i ];

slash[ i ] = 0;

}

while( left < right ){

mid = ( left + right )>>1;

if( Partition( mid ) > mid ) left = mid + 1;

else right = mid;

}

memset( slash , 0 , sizeof( slash ) );

Partition( left );

printf("%lld" , a[ 0 ] );

if( slash[ 0 ] )printf(" /");

for( int i = 1 ; i < N ; ++i ){

printf(" %lld" , a[ i ] );

if( slash[ i ] )printf(" /");

}

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Glass Beads

\*\*\*ID: 719

\*\*\*Juez: UVA

\*\*\*Tipo: Minimum Lexicographic Rotation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

char s[10001];

int main(){

int t;

scanf("%d" , &t );

while( t-- > 0 && scanf("%s",s ) ){

int N = strlen( s ) , ans = 0 , p = 1 , l = 0;

while( p < N && ans + l + 1 < N ){

if( s[ ans + l ] == s[ ( p + l ) % N ] ) ++l;

else if( s[ ans + l ] < s[ ( p + l ) % N ] ){

p = p + l + 1;

l = 0;

}else{

ans = max( ans + l + 1 , p );

p = ans + 1;

l = 0;

}

}

//minimo numero de movimientos

printf("%d\n",ans + 1 );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Division

\*\*\*ID: 725

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<bits/stdc++.h>

#include<cstdlib>

using namespace std;

string toStr(int n){

stringstream ss;

ss<<n;

string s;

ss>>s;

return s;

}

string llenar(string s){

int l=s.length();

for(int i=0;i<5-l;i++)s="0"+s;

return s;

}

bool Dif(string n,string m){

int a[10]={0};

for(int i=0;i<n.length();i++){

a[n[i]-'0']++;

}

for(int i=0;i<m.length();i++){

a[m[i]-'0']++;

}

for(int i=0;i<10;i++){

if(a[i]>1)return false;

}

return true;

}

int main(){

int n;

bool f=false;

while(cin>>n && n){

if(f)cout<<endl;

bool b=false;

for(int i=0;i<10;i++){

for(int j=0;j<10;j++){

if(i==j)continue;

for(int k=0;k<10;k++){

if(k==j || k==i)continue;

for(int q=0;q<10;q++){

if(q==k || q==j || q==i)continue;

for(int w=0;w<10;w++){

if(w==q || w==k || w==j || w==i)continue;

int num=i\*10000+j\*1000+k\*100+q\*10+w;

if(num%n == 0){

string n1=llenar(toStr(num));

string n2=llenar(toStr(num/n));

if(!Dif(n1,n2))continue;

cout<<n1<<" / "<<n2<<" = "<<n<<endl;

b=true;

}

}

}

}

}

}

if(!b)cout<<"There are no solutions for "<<n<<"."<<endl;

f=true;

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Hamming Distance Problem

\*\*\*ID: 729

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking, brute force

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

int sz, n;

int bitcount( int n ){

int cnt = 0;

while( n ){

n = n & ( n - 1 );

cnt++;

}

return cnt;

}

void print( int x ){

int bits[ 32 ] = { 0 };

for( int i = 0 ; i < 32 ; ++i ) if( x & 1<<i ) bits[ i ] = 1;

for( int i = n - 1 ; i >= 0 ; --i ) printf("%d" , bits[ i ] );

printf("\n");

}

int main(){

int t;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &n , &sz );

for( int i = 1 ; i < 1<<n ; ++i )

if( bitcount( i ) == sz ) print( i );

if( t ) printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Anagrams by Stack

\*\*\*ID: 732

\*\*\*Juez: UVA

\*\*\*Tipo: Stack

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

char s1[ 3005 ], s2[ 3005 ];

int l1 , l2;

vector<string> v;

void solve( int idx , string s, stack<char> S , int act ){

if( act == l2 ){

v.push\_back( s );

return;

}

stack<char> aux = S;

if( idx < l1 ){

aux.push( s1[ idx ] );

solve( idx + 1 , s + "i" , aux , act );

aux = S;

}

if( !S.empty() && S.top() == s2[ act ]){

aux.pop();

solve( idx , s + "o" , aux , act + 1 );

}

aux = S;

}

int main(){

while( gets( s1 ) ){

gets( s2 );

l1 = strlen( s1 );

l2 = strlen( s2 );

if( l1 != l2 ){

printf("[\n]\n");

continue;

}

stack<char> S;

v.clear();

solve( 0 , "" , S , 0 );

puts("[");

for( int i = 0 ; i < v.size() ; ++i ){

printf("%c" , v[ i ][ 0 ] );

for( int j = 1 ; j < v[ i ].length() ; ++j )printf(" %c" , v[ i ][ j ] );

printf("\n");

}

puts("]");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Soundex Indexing

\*\*\*ID: 739

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int value[26] = {0 , 1 , 2 , 3 ,0 , 1 , 2 , 0 , 0 , 2 ,2 , 4 , 5, 5 , 0 , 1 , 2 , 6 , 2 , 3 , 0 , 1 , 0 , 2 , 0 , 2 };

int main(){

char s[30];

int l, ant, cont;

printf(" NAME SOUNDEX CODE\n");

while( scanf("%s", s) !=EOF){

l = strlen( s );

printf(" %s",s);

for(int i = l ; i < 25 ;++i)putchar(' ');

putchar(s[0]);

cont = 1;

ant = value[ s[0] - 'A'];

for( int i = 1; i < l ; ++i ){

if( ant != value[ s[i] - 'A' ] && value[ s[i]- 'A'] != 0){

printf("%d",value [ s[i] - 'A'] );

cont++;

}

ant = value[ s[i] - 'A' ];

if( cont > 3)break;

}

for( int i = cont ; i < 4; ++i)putchar('0');

putchar('\n');

}

printf(" END OF OUTPUT\n");

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Burrows Wheeler Decoder

\*\*\*ID: 741

\*\*\*Juez: UVA

\*\*\*Tipo: Sorting

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

void concat( string s , vector< string> &v){

int i , sz = v.size();

for( i = 0 ; i < sz; ++i ){

v[ i ] = s[ i ] + v[ i ];

}

}

void print( vector< string > v ){

for( int i = 0 ; i < v.size() ; ++i ) cout<<v[ i ]<<endl;

}

int main(){

string s , aux;

int n , len , q = 0;

while( cin>>s>>n , n){

if( q ) printf("\n");

q = 1;

aux = s;

len = s.length();

vector< string > v( len , "");

concat( s , v );

while( len-- ){

sort( v.begin() , v.end() );

concat( aux , v );

}

sort( v.begin() , v.end() );

cout<<v[ n - 1 ].substr( 0 , v.size())<<endl;

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The MTM Machine

\*\*\*ID: 743

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, Simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

string s;

int len;

bool zeros(){

for( int i = 0 ; i < len ; ++i ) if( s[ i ] == '0' ) return true;

return false;

}

/\*

- 3X -> si X es de la forma 2X y produce Y entonces devolvemos Y2Y

- 2X -> devolvemos solamente X

\*/

bool possible;

string solve( int i ){

if( i == len ){

return "";

}

string ans = "" , tipo2;

if( s[ i ] == '3' ){

if( i + 1 < len ){

tipo2 = solve( i + 1 );

if( tipo2.length() > 0 ){

ans = tipo2 + "2" + tipo2;

}

else possible = false;

}

else possible = false;

}

else if( s[ i ] == '2' ){

if( i + 1 < len ){

ans = s.substr( i + 1 );

}

else possible = false;

}

else{

possible = false;

return "";

}

return ans;

}

int main(){

while( cin>>s ){

len = s.length();

if( len == 1 && s[ 0 ] == '0' ) break;

if( zeros() ) puts("NOT ACCEPTABLE");

else{

possible = true;

string ans = solve( 0 );

if( !possible ) puts("NOT ACCEPTABLE");

else printf("%s\n" , ans.c\_str() );

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Numeric Puzzles Again!

\*\*\*ID: 745

\*\*\*Juez: UVA

\*\*\*Tipo: Simulation, Brute Force , BigNum

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 30

char line[ MAX ];

int n , sz;

struct Data{

int digit , row , col;

int s[ MAX ][ MAX ];

};

vector< Data > ady[ MAX ];

//extraemos el digito de la linea

int digito(){

if( line[ 0 ] == '#' ) return -1;

for( int i = 0 ; line[ i ] ; ++i ){

if( line[ i ] >= '0' && line[ i ] <= '9' ){

return line[ i ] - '0';

}

}

}

//agregamos como fila cada linea correcta

void actualizar( Data &table , int fila ){

for( int i = 0 ; line[ i ] ; ++i ){

if( isdigit( line[ i ] ) )

table.s[ fila ][ i ] = line[ i ] - '0';

}

}

//Rotamos 90 grados una tabla

Data rotar( Data table ){

Data novo;

int i , j , row = table.row, col = table.col;

for( i = 0 ; i < row ; ++i ){

for( j = 0 ; j < col ; ++j ){

novo.s[ j ][ row - i - 1 ] = table.s[ i ][ j ];

}

}

novo.digit = table.digit;

novo.row = col;

novo.col = row;

return novo;

}

//Imprimirmos resultado

void imprimir( Data act ){

for( int i = 0 ; i < act.row ; ++i ){

for( int j = 0 ; j < act.col ; ++j ){

printf("%d" , act.s[ i ][ j ] );

}

printf("\n");

}

}

Data ans;

bool finish;

//Backtracking probando todas las posibilidades con las figuras rotadas

void solve( Data resp , int len ){

if( len == sz ){

ans = resp;

finish = true;

return;

}

int i , j , k , a , b;

for( i = 0 ; i < ady[ len ].size() ; ++i ){

Data act = ady[ len ][ i ];

for( j = 0 ; j + act.row <= n ; ++j ){

for( k = 0 ; k + act.col <= n ; ++k ){

Data novo = resp;

bool bad = false;

//Verificamos si la posicion donde deseamos colocar ya fue ocupada por otra pieza

for( a = 0 ; a < act.row ; ++a ){

for( b = 0 ; b < act.col ; ++b ){

if( act.s[ a ][ b ] == 0 ) continue;

if( novo.s[ a + j ][ b + k ] != 0 ){

a = act.row;

bad = true;

break;

}

}

}

if( bad ) continue;

for( a = 0 ; a < act.row ; ++a ){

for( b = 0 ; b < act.col ; ++b ){

if( act.s[ a ][ b ] == 0 ) continue;

novo.s[ a + j ][ b + k ] = act.s[ a ][ b ];

}

}

solve( novo , len + 1 );

if( finish ) return;

}

}

}

}

//BIGNUM

const int MAXD = 105, DIG = 9, BASE = 1000000000;

const unsigned long long BOUND = numeric\_limits <unsigned long long> :: max () - (unsigned long long) BASE \* BASE;

struct bignum

{

int D, digits [MAXD / DIG + 2];

inline void trim ()

{

while (D > 1 && digits [D - 1] == 0)

D--;

}

inline void init (long long x)

{

memset (digits, 0, sizeof (digits));

D = 0;

do

{

digits [D++] = x % BASE;

x /= BASE;

}

while (x > 0);

}

inline bignum (long long x)

{

init (x);

}

inline bignum (int x = 0)

{

init (x);

}

inline bignum (char \*s)

{

memset (digits, 0, sizeof (digits));

int len = strlen (s), first = (len + DIG - 1) % DIG + 1;

D = (len + DIG - 1) / DIG;

for (int i = 0; i < first; i++)

digits [D - 1] = digits [D - 1] \* 10 + s [i] - '0';

for (int i = first, d = D - 2; i < len; i += DIG, d--)

for (int j = i; j < i + DIG; j++)

digits [d] = digits [d] \* 10 + s [j] - '0';

trim ();

}

inline char \*str ()

{

trim ();

char \*buf = new char [DIG \* D + 1];

int pos = 0, d = digits [D - 1];

do

{

buf [pos++] = d % 10 + '0';

d /= 10;

}

while (d > 0);

reverse (buf, buf + pos);

for (int i = D - 2; i >= 0; i--, pos += DIG)

for (int j = DIG - 1, t = digits [i]; j >= 0; j--)

{

buf [pos + j] = t % 10 + '0';

t /= 10;

}

buf [pos] = '\0';

return buf;

}

inline bool operator < (const bignum &o) const

{

if (D != o.D)

return D < o.D;

for (int i = D - 1; i >= 0; i--)

if (digits [i] != o.digits [i])

return digits [i] < o.digits [i];

return false;

}

inline bignum operator + (const bignum &o) const

{

bignum sum = o;

int carry = 0;

for (sum.D = 0; sum.D < D || carry > 0; sum.D++)

{

sum.digits [sum.D] += (sum.D < D ? digits [sum.D] : 0) + carry;

if (sum.digits [sum.D] >= BASE)

{

sum.digits [sum.D] -= BASE;

carry = 1;

}

else

carry = 0;

}

sum.D = max (sum.D, o.D);

sum.trim ();

return sum;

}

inline bignum operator \* (const bignum &o) const

{

bignum prod = 0;

unsigned long long sum = 0, carry = 0;

for (prod.D = 0; prod.D < D + o.D - 1 || carry > 0; prod.D++)

{

sum = carry % BASE;

carry /= BASE;

for (int j = max (prod.D - o.D + 1, 0); j <= min (D - 1, prod.D); j++)

{

sum += (unsigned long long) digits [j] \* o.digits [prod.D - j];

if (sum >= BOUND)

{

carry += sum / BASE;

sum %= BASE;

}

}

carry += sum / BASE;

prod.digits [prod.D] = sum % BASE;

}

prod.trim ();

return prod;

}

};

//Obtenemos maximo rotando la figura y sumando valores, recordar son 20 digitos a lo mucho, por ello BigInteger

void getMax(){

int i , j , k;

Data aux = ans;

bignum sum , maxi = bignum( 0 ) , num;

for( k = 0 ; k < 4 ; ++k ){

sum = bignum( 0 );

for( i = 0 ; i < n ; ++i ){

num = bignum( 0 );

for( j = 0 ; j < n ; ++j ){

num = num \* 10 + bignum( aux.s[ i ][ j ] );

}

sum = sum + num;

}

if( maxi < sum ){

maxi = sum;

ans = aux;

}

aux = rotar( aux );

}

imprimir( ans );

}

int main(){

int digit , aux , i , fila , col;

while( scanf("%d" , &n ) , n ){

scanf("%d" , &sz );

gets( line );

gets( line );

for( i = 0 ; i < sz ; ++i ){

Data table;

memset( table.s , 0, sizeof( table.s ) );

fila = col = 1;

col = max( col , ( int )strlen( line ) );

digit = digito();

if( digit == -1 ) break;

actualizar( table , 0 );

while( gets( line ) ){

aux = digito();

if( aux != digit ) break;

actualizar( table , fila++ );

col = max( col , ( int )strlen( line ) );

}

table.row = fila; table.col = col; table.digit = digit;

ady[ i ].push\_back( table );

}

Data resp;

resp.row = resp.col = n;

memset( resp.s , 0 , sizeof( resp.s ) );

finish = false;

solve( resp , 0 );

getMax();

printf("\n");

for( i = 0 ; i <= sz ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 8 Queen Chess Problem

\*\*\*ID: 750

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

int idx, num;

int row[ 10 ];

int rightDiag[ 20 ], leftDiag[ 20 ];

int resp[ 100 ][ 10 ] , tmp[ 8 ];

void generateSolution( int len ){

if( len == 8 ){

for( int i = 0 ; i < 8 ; ++i ){

resp[ idx ][ i ] = tmp[ i ];

}

idx++;

return;

}

for( int i = 0 ; i < 8 ; ++i ){ //fila

if( row[ i ] && leftDiag[ len + i ] && rightDiag[ len - i + 8 ] ){

row[ i ] = leftDiag[ len + i ] = rightDiag[ len - i + 8 ] = 0;

tmp[ len ] = i + 1;

generateSolution( len + 1 );

row[ i ] = leftDiag[ len + i ] = rightDiag[ len - i + 8 ] = 1;

}

}

}

int main(){

int t , x , y;

scanf("%d" , &t );

idx = 0;

memset ( rightDiag , 1, sizeof (rightDiag));

memset ( leftDiag , 1 , sizeof (leftDiag));

memset( row , 1 , sizeof( row ) );

generateSolution( 0 );

for( int q = 0 ; q < t ; ++q ){

if( q )putchar('\n');

scanf("%d %d" , &x , &y );

y--;

printf("SOLN COLUMN\n");

printf(" # 1 2 3 4 5 6 7 8\n\n");

num = 0;

for( int i = 0 ; i < idx ; ++i ){

if( resp[ i ][ y ] == x ){

printf ("%2d ", ++num);

printf("%d" , resp[ i ][ 0 ] );

for( int j = 1 ; j < 8 ; ++j ){

printf(" %d" , resp[ i ][ j ] );

}

printf("\n");

}

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# A plug for UNIX

\*\*\*ID: 753

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Max Flow

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define INF 1<<30

#define MAX 705

char plug[ MAX ][ 25 ] , plugOnly[ MAX ][ 25 ];

int num\_plug;

struct Edge{

int u;

int v;

int c;

int r;

Edge( int uu, int vv, int cc, int rr): u(uu), v(vv), c(cc), r(rr){}

};

vector< Edge > E;

vector< vector<int> > ady; //posee indices de aristas

int maxflow( int source , int sink ){

for( int f = 0; ;){

///vector de enlaces previos para ver la ruta recorrida del argumenting path

///tiene indices de aristas

vector<int> prev( ady.size() , -1 );

queue<int> Q;

Q.push( source );

while( !Q.empty() ){

int u = Q.front(); Q.pop();

for( int i = 0 ; i < ady[ u ].size(); ++i ){

///indice de arista actual

int e = ady[ u ][ i ];

int v = E[ e ].v;

///si no puede pasar o si ya fue visitado

if( E[ e ].c == 0 || prev[ v ] >= 0 )continue;

Q.push( v );

prev[ v ] = e;

}

}

///si ya no hay argumenting paths

if( prev[ sink ] < 0 )return f;

///hallamos minima capacidad de argumentin path

int c = 0x7fffffff;

for( int x = sink ; x != source ; x = E[ prev[ x ] ].u ){

c <?= E[ prev[ x ] ].c;

}

///para nodos de ida 1->2 restamos capacidad minima

///para nodos de vuevla 2<-1 sumamos capacidad minima

for( int x = sink ; x != source ; x = E[ prev[ x ] ].u ){

E[ prev[ x ] ].c -= c;

E[ E[ prev[ x ] ].r ].c += c;

}

f += c;

}

}

///indices de aristas comienzan x cero

///si tengo 1->2 su indice es 0, si 1<-2 su indice es 1

///x tanto r de primero es 1, de segundo es 0

void add( int u , int v , int c , bool dirigido = true ){

Edge e1( u , v , c , E.size() + 1 );

ady[ u ].push\_back( E.size() );

E.push\_back( e1 );

///capacidad 0 si es dirigido

Edge e2( v , u , ( dirigido )? 0 : c , E.size() - 1 );

ady[ v ].push\_back( E.size() );

E.push\_back( e2 );

}

int getId( char \*s){

for( int i = 1 ; i < num\_plug ; ++i ) if( strcmp( s , plug[ i ] ) == 0 ) return i;

strcpy( plug[ num\_plug ] , s );

return num\_plug++;

}

int main(){

int t , n , m , k;

char u[ 25 ] , v[ 25 ];

scanf("%d" ,&t );

for( int q = 0 ; q < t ; ++q ){

if( q )putchar('\n');

E.clear();

ady = vector< vector< int > > ( MAX , vector<int>() );

scanf("%d" , &n);

for( int i = 1 ; i <= n ; ++i ){

scanf("%s" ,&plug[ i ] );

}

num\_plug = n + 1;

scanf("%d" , &m );

for( int i = 0 ; i < m ; ++i ){

scanf("%s %s", &u , &v );

int uu = getId( u );

add( 0 , uu , 1 );

add( uu , getId( v ) , 1 );

}

scanf("%d" , &k );

while( k-- ){

scanf("%s %s" , &u , &v );

add( getId( u ) , getId( v ) , INF );

}

for( int i = 1 ; i <= n ; ++i ){

add( getId( plug[ i ] ) , MAX - 1 , 1 );

}

int flow = maxflow( 0 , MAX - 1 );

printf("%d\n" , m - flow );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 487-3279

\*\*\*ID: 755

\*\*\*Juez: UVA

\*\*\*Tipo: Data structures

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

#include<vector>

#include<map>

using namespace std;

string value[]={"2","2","2","3","3","3","4","4","4","5","5","5","6","6","6","7","","7","7","8","8","8","9","9","9",""};

string convert(string s){

string ss="";

bool h=false;

for(int i=0;i<s.length();i++){

if(s[i]=='-'){

ss+="";

}

else if(isdigit(s[i]))ss+=s[i];

else ss+=value[s[i]-'A'];

}

ss=ss.substr(0,3)+"-"+ss.substr(3);

return ss;

}

int main(){

int casos,n;

scanf("%d",&casos);

string s;

cin.get();

while(casos--){

getline(cin,s);

scanf("%d",&n);

map<string,int> mp;

map<string,int>::iterator it;

while(n--){

cin>>s;

s=convert(s);

mp[s]++;

}

bool b=false;

for(it=mp.begin();it!=mp.end();it++){

if((\*it).second>1){

cout<<(\*it).first<<" "<<(\*it).second<<endl;

b=true;

}

}

if(!b)printf("No duplicates.\n");

if(casos)putchar('\n');

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Biorhythms

\*\*\*ID: 756

\*\*\*Juez: UVA

\*\*\*Tipo: Chinese Remainder Theorem

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int extended\_euclid( int a , int b , int &x , int &y ){

if( b == 0 ){ x = 1; y = 0; return a; }

int d = extended\_euclid( b , a % b , y , x );

y -= a/b \* x;

return d;

}

int mod( int a , int b ){

return ( (a % b) + b ) % b;

}

#define pii pair< int , int >

#define mp make\_pair

pii chinese\_remainder\_theorem( int a , int n1 , int b , int n2 ){

int x , y , d;

d = extended\_euclid( n1 , n2 , x , y );

if( a % d != b % d ) return mp( 0 , -1 );

return mp( mod( a \* y \* n2 + b \* x \* n1 , n1 \* n2 )/d , ( n1 \* n2 )/d );

}

//Consideramos mas de 2 ecuaciones

pii chinese\_remainder\_theorem( vector< int > A , vector< int > N ){

int i , sz = A.size();

pii ans = mp( A[ 0 ] , N[ 0 ] );

for( i = 1 ; i < sz ; ++i ){

ans = chinese\_remainder\_theorem( ans.first , ans.second , A[ i ] , N[ i ] );

if( ans.second == -1 ) break;

}

return ans;

}

int A[ 3 ] , N[ 3 ] = { 23 , 28 , 33 };

int main(){

int day , q = 1 , resp;

while( scanf("%d %d %d %d" , &A[ 0 ] , &A[ 1 ] , &A[ 2 ] , &day ) != EOF ){

if( A[ 0 ] == -1 && A[ 1 ] == -1 && A[ 2 ] == -1 )break;

pii ans = chinese\_remainder\_theorem( vector< int >( A , A + 3 ) , vector< int >( N , N + 3 ) );

resp = mod( ans.first - day , 21252 );

if( resp == 0 ) resp = 21252;

printf("Case %d: the next triple peak occurs in %d days.\n" , q++ , resp );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Same Game

\*\*\*ID: 758

\*\*\*Juez: UVA

\*\*\*Tipo: Simulation, flood fill

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAXR 10

#define MAXC 15

#define pii pair<int,int>

#define mp make\_pair

char s[ MAXR + 5 ][ MAXC + 5 ] , aux[ MAXR + 5 ][ MAXC + 5];

int dx[] = { 0 , 0 , 1 , -1 };

int dy[] = { 1 , -1 , 0 , 0 };

bool seen[ MAXR ][ MAXC ];

void rearrange(){

for( int j = 0 ; j < MAXC ; ++j ){

for( int i = MAXR - 1 ; i >= 0 ; --i ){

if( s[i][j] == '.' ){

for( int k = i - 1 ; k >= 0 ; --k ){

if( s[k][j] != '.' ){

s[i][j] = s[k][j];

s[k][j] = '.';

break;

}

}

}

}

}

for( int j = 0 ; j < MAXC ; ++j ){

int i = 0;

for( ; i < MAXR ; ++i ){

if( s[i][j] != '.')

break;

}

if( i == MAXR ){

int k;

for( k = j + 1 ; k < MAXC ; ++k ){

int w;

for( w = 0 ; w < MAXR ; ++w ){

if( s[w][k] != '.'){

break;

}

}

if( w != MAXR ){

for( int v = 0 ; v < MAXR ; ++v ){

s[v][j] = s[v][k];

s[v][k] = '.';

}

break;

}

}

}

}

}

vector<pii> coord;

int flood( int x , int y , char color ){

aux[ x ][ y ] = '.';

coord.push\_back( mp( x ,y ) );

int ans = 1;

for( int i = 0 ; i < 4 ; ++i ){

int nx = dx[i] + x;

int ny = dy[i] + y;

if( nx >= 0 && ny >= 0 && nx < MAXR && ny < MAXC && color == aux[ nx ][ ny ] ){

ans += flood( nx , ny , color );

}

}

return ans;

}

void printArray(){

cout<<endl;

for( int i = 0 ; i < MAXR; ++i){

for( int j = 0 ; j < MAXC ; ++j ){

cout<<s[i][j];

}

cout<<endl;

}

}

struct Data{

int x , y , cnt;

Data(){};

Data( int xx , int yy , int c ): x(xx) , y(yy) , cnt(c){}

};

bool cmp( Data d1 , Data d2 ){

return (d1.cnt > d2.cnt) || (d1.cnt == d2.cnt && d1.y < d2.y )

|| ( d1.cnt == d2.cnt && d1.y == d2.y && d1.x > d2.x );

}

vector<Data> countBalls( int xini ){

vector<Data> countMap;

memcpy( aux , s , sizeof( s ) );

for( int i = xini ; i < MAXR ; ++i ){

for( int j = 0 ; j < MAXC ; ++j ){

if( aux[i][j] != '.' ){

coord.clear();

int cnt = flood( i , j , aux[i][j] );

for( int k = 0 ; k < coord.size() ; ++k ){

int x = coord[k].first , y = coord[k].second;

countMap.push\_back( Data(x , y , cnt ));

}

}

}

}

sort( countMap.begin() , countMap.end() , cmp );

return countMap;

}

void removeBalls( int x , int y , char color ){

s[x][y] = '.';

for( int i = 0 ; i < 4 ; ++i ){

int nx = dx[i] + x;

int ny = dy[i] + y;

if( nx >= 0 && ny >= 0 && nx < MAXR && ny < MAXC && color == s[ nx ][ ny ] ){

removeBalls( nx , ny , color );

}

}

}

int main(){

int t, moves , points , ans , xini, remain;

scanf("%d" , &t );

for( int q = 1 ; q <= t ; ++q ){

if( q - 1 )

printf("\n");

for( int i = 0 ; i < MAXR && scanf("%s" , s[i] ) ; ++i );

ans = 0;

printf("Game %d:\n\n" , q );

xini = 0;

moves = 0;

remain = 0;

while( true ){

vector<Data> countMap = countBalls(xini);

if( countMap.size() == 0 ){ ans+=1000; break;}

int x = countMap[0].x, y = countMap[0].y, cnt = countMap[0].cnt;

if( cnt <= 1 ){

remain += countMap.size();

break;

}

char color = s[ x ][ y ];

removeBalls( x , y , s[ x ][ y ] );

rearrange();

moves++;

points = ( cnt - 2 ) \* ( cnt - 2 );

x = MAXR - x; y++;

printf("Move %d at (%d,%d): removed %d balls of color %c, got %d points.\n" , moves , x , y , cnt , color , points );

ans += points;

}

printf("Final score: %d, with %d balls remaining.\n" , ans , remain );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Return of the Roman Empire

\*\*\*ID: 759

\*\*\*Juez: UVA

\*\*\*Tipo: Roman Cnoversion

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int romanToDecimal( const char \*s ) {

static int val[ 256 ] = { -1 };

if ( val[ 0 ] < 0) {

int a[] = { 1 , 5 , 10 , 50 , 100 , 500 , 1000 };

for (int i = 0; i < 256; ++i ) val[ i ] = 0;

for (int i = 0; i < 7; ++i ) val[ "IVXLCDM"[ i ] ] = a[ i ];

}

int res = 0;

for ( ; \*s ; ++s ) {

res += val[ \*s ];

for ( const char \*t = s + 1 ; \*t ; ++t ) {

if ( val[ \*t ] > val[ \*s ] ) {

res -= 2 \* val[ \*s ];

break;

}

}

}

return res;

}

#define MAX 104

char s[ MAX ];

string decimal2roman( int input)

{

const string roman[13] = { "M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"};

const int decimal[13] = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};

string romanvalue = "";

for(int i = 0 ; i < 13 ; ++i )

{

while( input >= decimal[ i ] )

{

input -= decimal[ i ];

romanvalue += roman[ i ];

}

}

return romanvalue;

}

bool valid(){

string roman= "IVXLCDM";

for( int i = 0 ; s[ i ] ; ++i ){

if( roman.find( s[ i ] ) == -1 ) return false;

}

return true;

}

int main(){

int res;

while( gets( s ) ){

if( valid() ){

res = romanToDecimal( s );

if( res < 4000 && strcmp( decimal2roman( res ).c\_str() , s ) == 0 ){

printf("%d\n" , res );

}

else puts("This is not a valid number");

}

else puts("This is not a valid number");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# DNA Sequencing

\*\*\*ID: 760

\*\*\*Juez: UVA

\*\*\*Tipo: Suffix Array

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<cmath>

#include<cstring>

#include<stdio.h>

#define MAXN 1000

#define Clear(a) memset(a,0,sizeof(a))

#define Max(a,b) a>b?a:b

#define Min(a,b) a<b?a:b

using namespace std;

#define F(x) ((x)/3+((x)%3==1?0:tb))

#define G(x) ((x)<tb?(x)\*3+1:((x)-tb)\*3+2)

int wa[ MAXN ] , wb[ MAXN ],wv[ MAXN ], wss[ MAXN ];

int LCP[ MAXN ], pos[ MAXN \* 3 ], rank[ MAXN ], r[ MAXN \* 3 ];

char s[ MAXN ];

int c0(int r[],int a,int b) {return r[a]==r[b]&&r[a+1]==r[b+1]&&r[a+2]==r[b+2];}

int c12(int k,int r[],int a,int b) {

if( k == 2 )

return r[a]<r[b]||r[a]==r[b]&&c12(1,r,a+1,b+1);

else return r[a]<r[b]||r[a]==r[b]&&wv[a+1]<wv[b+1];

}

void sort(int r[],int a[],int b[],int n,int m)

{

int i;

for( i = 0 ; i < n ; ++i ) wv[ i ] = r[ a[ i ] ];

for( i = 0 ; i < m ; ++i ) wss[ i ] = 0;

for( i = 0 ; i < n ; ++i ) wss[wv[i]]++;

for( i = 1 ; i < m ; ++i ) wss[ i ] += wss[i-1];

for( i = n - 1 ; i >= 0 ; --i ) b[--wss[wv[i]]]=a[i];

return;

}

void dc3(int r[],int pos[],int n,int m)

{

int i , j , \*rn = r + n ,\*san=pos+n,ta=0,tb=(n+1)/3,tbc=0,p;

r[ n ] = r[ n + 1 ] = 0;

for( i = 0 ; i < n ; ++i ) if( i % 3 != 0 ) wa[ tbc++ ]=i;

sort( r + 2 , wa , wb , tbc , m );

sort( r + 1 , wb , wa , tbc , m );

sort( r , wa , wb , tbc , m );

for( p = 1 , rn[ F( wb[ 0 ] ) ] = 0 ,i=1; i<tbc; ++i)

rn[ F( wb[ i ] ) ] = c0( r , wb[i-1] , wb[i] )?p-1:p++;

if( p < tbc ) dc3( rn , san , tbc , p );

else for( i = 0 ; i < tbc ; ++i ) san[rn[i]] = i;

for( i = 0 ; i < tbc; ++i )if(san[i]<tb) wb[ta++]=san[i]\*3;

if( n % 3 == 1 ) wb[ta++] = n - 1;

sort( r , wb , wa , ta , m );

for( i = 0 ; i < tbc ; ++i ) wv[wb[i]=G(san[i])]=i;

for( i = 0 ,j=0 , p=0 ; i<ta && j < tbc ; ++p )

pos[ p ] =c12(wb[j]%3,r,wa[i],wb[j])?wa[i++]:wb[j++];

for( ; i < ta ; ++p ) pos[ p ] = wa[ i++ ];

for( ; j < tbc ; ++p ) pos[ p ] = wb[ j++ ];

return;

}

void getLCP( int r[], int pos[], int n, int rank[], int LCP[])

{

int i , j , k = 0 ;

for( i = 1 ; i <= n ;++i) rank[ pos[ i ] ] = i;

for( i = 0 ; i < n ; LCP[ rank[ i++ ] ] = k )

for( k ? k--:0, j = pos[rank[i]-1]; r[i+k]==r[j+k] ; ++k );

return;

}

// sa[i] i [1..n] (0..n-1)

// rank[i] i [0..n-1]

// height[i] i [1..n]

void get\_suffix(char s[], int rank[], int pos[], int LCP[])

{

int n = strlen( s );

for( int i = 0 ; i <= n ; ++i ) r[ i ] = s[ i ];

dc3( r , pos, n + 1 , 256);

getLCP( r, pos , n , rank , LCP );

}

char str[ MAXN];

int main(){

bool b = false;

while( gets( str ) ){

s[ 0 ] = '\0';

gets( s );

if( b ) printf("\n");

b = true;

int len = strlen( str ) , i , j , l = len , sub = 0;

str[ len++ ] = '.';

for( i = 0 ; s[ i ] ; ++i ) str[ len++ ] = s[ i ];

str[ len ] = '\0';

Clear( rank );Clear( pos ); Clear( LCP );

get\_suffix( str , rank, pos , LCP );

for( i = 1 ; i <= len ; ++i ){

if( ( pos[ i ] >= l && pos[ i - 1 ] < l ) || ( pos[ i ] < l && pos[ i - 1 ] >= l ) ){

sub = max( sub , LCP[ i ] );

}

}

string ant = "-1";

for( i = 1 ; i <= len ; ++i ){

if( ( ( pos[ i ] >= l && pos[ i - 1 ] < l ) || ( pos[ i ] < l && pos[ i - 1 ] >= l ) ) && sub > 0 && LCP[ i ] == sub ){

string aux = "";

for( j = 0 ; j < LCP[ i ] ; ++j ) aux += str[ pos[ i ] + j ];

if( aux != ant ) printf("%s\n" , aux.c\_str() );

ant = aux;

}

}

if( sub == 0 ) puts("No common sequence.");

gets( s );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Fibinary Numbers

\*\*\*ID: 763

\*\*\*Juez: UVA

\*\*\*Tipo: Fibonacci Numbers, BigInteger

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

const int MAXD = 105, DIG = 9, BASE = 1000000000;

const unsigned long long BOUND = numeric\_limits <unsigned long long> :: max () - (unsigned long long) BASE \* BASE;

struct bignum

{

int D, digits [MAXD / DIG + 2];

inline void trim ()

{

while (D > 1 && digits [D - 1] == 0)

D--;

}

inline void init (long long x)

{

memset (digits, 0, sizeof (digits));

D = 0;

do

{

digits [D++] = x % BASE;

x /= BASE;

}

while (x > 0);

}

inline bignum (long long x)

{

init (x);

}

inline bignum (int x = 0)

{

init (x);

}

inline bignum (char \*s)

{

memset (digits, 0, sizeof (digits));

int len = strlen (s), first = (len + DIG - 1) % DIG + 1;

D = (len + DIG - 1) / DIG;

for (int i = 0; i < first; i++)

digits [D - 1] = digits [D - 1] \* 10 + s [i] - '0';

for (int i = first, d = D - 2; i < len; i += DIG, d--)

for (int j = i; j < i + DIG; j++)

digits [d] = digits [d] \* 10 + s [j] - '0';

trim ();

}

inline char \*str ()

{

trim ();

char \*buf = new char [DIG \* D + 1];

int pos = 0, d = digits [D - 1];

do

{

buf [pos++] = d % 10 + '0';

d /= 10;

}

while (d > 0);

reverse (buf, buf + pos);

for (int i = D - 2; i >= 0; i--, pos += DIG)

for (int j = DIG - 1, t = digits [i]; j >= 0; j--)

{

buf [pos + j] = t % 10 + '0';

t /= 10;

}

buf [pos] = '\0';

return buf;

}

inline bool operator < (const bignum &o) const

{

if (D != o.D)

return D < o.D;

for (int i = D - 1; i >= 0; i--)

if (digits [i] != o.digits [i])

return digits [i] < o.digits [i];

return false;

}

inline bool operator == (const bignum &o) const

{

if (D != o.D)

return false;

for (int i = 0; i < D; i++)

if (digits [i] != o.digits [i])

return false;

return true;

}

inline bignum operator << (int p) const

{

bignum temp;

temp.D = D + p;

for (int i = 0; i < D; i++)

temp.digits [i + p] = digits [i];

for (int i = 0; i < p; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum operator >> (int p) const

{

bignum temp;

temp.D = D - p;

for (int i = 0; i < D - p; i++)

temp.digits [i] = digits [i + p];

for (int i = D - p; i < D; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum range (int a, int b) const

{

bignum temp = 0;

temp.D = b - a;

for (int i = 0; i < temp.D; i++)

temp.digits [i] = digits [i + a];

return temp;

}

inline bignum operator + (const bignum &o) const

{

bignum sum = o;

int carry = 0;

for (sum.D = 0; sum.D < D || carry > 0; sum.D++)

{

sum.digits [sum.D] += (sum.D < D ? digits [sum.D] : 0) + carry;

if (sum.digits [sum.D] >= BASE)

{

sum.digits [sum.D] -= BASE;

carry = 1;

}

else

carry = 0;

}

sum.D = max (sum.D, o.D);

sum.trim ();

return sum;

}

inline bignum operator - (const bignum &o) const

{

bignum diff = \*this;

for (int i = 0, carry = 0; i < o.D || carry > 0; i++)

{

diff.digits [i] -= (i < o.D ? o.digits [i] : 0) + carry;

if (diff.digits [i] < 0)

{

diff.digits [i] += BASE;

carry = 1;

}

else

carry = 0;

}

diff.trim ();

return diff;

}

inline bignum operator \* (const bignum &o) const

{

bignum prod = 0;

unsigned long long sum = 0, carry = 0;

for (prod.D = 0; prod.D < D + o.D - 1 || carry > 0; prod.D++)

{

sum = carry % BASE;

carry /= BASE;

for (int j = max (prod.D - o.D + 1, 0); j <= min (D - 1, prod.D); j++)

{

sum += (unsigned long long) digits [j] \* o.digits [prod.D - j];

if (sum >= BOUND)

{

carry += sum / BASE;

sum %= BASE;

}

}

carry += sum / BASE;

prod.digits [prod.D] = sum % BASE;

}

prod.trim ();

return prod;

}

inline double double\_div (const bignum &o) const

{

double val = 0, oval = 0;

int num = 0, onum = 0;

for (int i = D - 1; i >= max (D - 3, 0); i--, num++)

val = val \* BASE + digits [i];

for (int i = o.D - 1; i >= max (o.D - 3, 0); i--, onum++)

oval = oval \* BASE + o.digits [i];

return val / oval \* (D - num > o.D - onum ? BASE : 1);

}

inline pair <bignum, bignum> divmod (const bignum &o) const

{

bignum quot = 0, rem = \*this, temp;

for (int i = D - o.D; i >= 0; i--)

{

temp = rem.range (i, rem.D);

int div = (int) temp.double\_div (o);

bignum mult = o \* div;

while (div > 0 && temp < mult)

{

mult = mult - o;

div--;

}

while (div + 1 < BASE && !(temp < mult + o))

{

mult = mult + o;

div++;

}

rem = rem - (o \* div << i);

if (div > 0)

{

quot.digits [i] = div;

quot.D = max (quot.D, i + 1);

}

}

quot.trim ();

rem.trim ();

return make\_pair (quot, rem);

}

inline bignum operator / (const bignum &o) const

{

return divmod (o).first;

}

inline bignum operator % (const bignum &o) const

{

return divmod (o).second;

}

inline bignum power (int exp) const

{

bignum p = 1, temp = \*this;

while (exp > 0)

{

if (exp & 1) p = p \* temp;

if (exp > 1) temp = temp \* temp;

exp >>= 1;

}

return p;

}

};

inline bignum gcd (bignum a, bignum b)

{

bignum t;

while (!(b == 0))

{

t = a % b;

a = b;

b = t;

}

return a;

}

#define MAX 150

char n1[ MAX ] , n2[ MAX ];

bignum fib[ MAX ];

bignum convert( char \*s ){

int i , j , len = strlen( s );

bignum n = 0;

for( i = len - 1 , j = 1 ; i >= 0 ; --i , ++j ){

if( s[ i ] == '1' ){

n = n + fib[ j ];

}

}

return n;

}

char ans[ MAX ];

bool b[ MAX ];

void solve( bignum n ){

int i;

memset( b , 0 , sizeof( b ) );

for( i = MAX - 1 ; i >= 1 ; --i ){

if( n == fib[ i ] || fib[ i ] < n ){

n = n - fib[ i ];

b[ i ] = 1;

if( n == 0 ) break;

}

}

for( i = MAX - 1 ; i >= 1 ; --i ){

if( b[ i ] ) break;

}

int j = 0;

for( ; i >= 1 ; --i ){

ans[ j++ ] = ( b[ i ] == 1 ? '1' : '0');

}

ans[ j ] = '\0';

}

int main(){

int i;

fib[ 1 ] = 1;

fib[ 2 ] = 2;

for( i = 3 ; i < MAX ; ++i ) fib[ i ] = fib[ i - 1 ] + fib[ i - 2 ];

bignum num1 , num2;

bool b = false;

while( scanf("%s %s" , &n1 , &n2 ) != EOF ){

if( b ) printf("\n");

b = true;

num1 = convert( n1 );

num2 = convert( n2 );

if( num1 + num2 == 0 ) {

puts("0");

continue;

}

solve( num1 + num2 );

printf("%s\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Hamiltonian Cycle

\*\*\*ID: 775

\*\*\*Juez: ICPC - Ores Theorem

\*\*\*Tipo: DFS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

//http://en.wikipedia.org/wiki/Ore%27s\_theorem

#define MAX 300

int V;

char s[ MAX ];

vector< int > ady[ MAX ];

int seen[ MAX ] , ini = 0;

int path[ MAX ];

bool end;

void solve( int x , int parent , int len ){

path[ len ] = x;

int i , v;

for( i = 0 ; i < ady[ x ].size() ; ++i ){

v = ady[ x ][ i ];

if( !seen[ v ] ){

seen[ v ] = 1;

solve( v , x , len + 1 );

if( end ) return;

seen[ v ] = 0;

}

else if( v != parent && v == ini && len == V ){

for( int j = 1 ; j <= len ; ++j ) printf("%d " , path[ j ] + 1 );

puts("1");

end = true;

return;

}

}

}

int main(){

int i , u , v;

while( scanf("%d" , &V ) != EOF ){

gets( s );

while( gets( s ) , s[ 0 ] != '%' ){

stringstream ss( s );

ss>>u>>v;

u--; v--;

ady[ u ].push\_back( v );

ady[ v ].push\_back( u );

}

end = false;

seen[ 0 ] = 1;

solve( 0 , -1 , 1 );

for( i = 0 ; i <= V ; ++i ) ady[ i ].clear() , seen[ i ] = 0;

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Monkeys in a Regular Forest

\*\*\*ID: 776

\*\*\*Juez: UVA

\*\*\*Tipo: Flood Fill

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int h, w;

#define MAX 130

char ady[ MAX ][ MAX ];

int resp[ MAX ][ MAX ];

int digits[ MAX ][ MAX ];

bool seen[ MAX ][ MAX ];

int maxD[ MAX ]; //maxima cantidad de digitos x columna para alinear;

int dx[ 8 ] = { 1 , -1 , 0 , 0 , 1 , 1 , -1 , -1 };

int dy[ 8 ] = { 0 , 0 , 1 , -1 , 1 , -1 , 1 , -1};

int numDigits( int n ){

int c = 0;

while( n > 0 ){

n /= 10;

c++;

}

return c;

}

void dfs( int x , int y , char c, int val ){

if( x < 0 || y < 0 || x >= h || y >= w || seen[ x ][ y ] || ady[ x ][ y ] != c )return;

seen[ x ][ y ] = true;

resp[ x ][ y ] = val;

if( val > maxD[ y ] ) maxD[ y ] = val;

for( int i = 0 ; i < 8 ; ++i )dfs( dx[ i ] + x , dy[ i ] + y , c , val );

}

int main(){

int l, ancho ;

char C;

while( 1 ){

memset( ady , 0 , sizeof( ady ) );

h = w = ancho = 0;

while( ( C = getchar() ) != EOF ){

if( isalpha( C ) ){

ady[ h ][ w++ ] = C;

if( ancho < w ) ancho = w;

}

else if( C == '\n'){

h++; w = 0;

}

else if( C == '%'){ C = getchar(); break;}

}

w = ancho;

if( w == 0 || h == 0 ) break;

int idx = 1;

memset( seen , false , sizeof( seen ) );

memset( maxD , 0 , sizeof( maxD ) );

for( int i = 0 ; i < h ; ++i ){

for( int j = 0 ; j < w ; ++j ){

if( !seen[ i ][ j ] ){

dfs( i , j , ady[ i ][ j ] , idx++ );

}

}

}

for( int i = 0 ; i < w ; ++i ) maxD[ i ] = numDigits(( maxD[ i ] ) );

for( int i = 0 ; i < h ; ++i ){

for( int j = 0 ; j < w ; ++j ){

printf((j + 1) >= w ? "%\*d\n" : "%\*d " , maxD[ j ], resp[ i ][ j ] );

}

}

printf("%%\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Indexing

\*\*\*ID: 789

\*\*\*Juez: UVA

\*\*\*Tipo: Map

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

char s[ MAX ];

bool isLetter(char c){return c>='A' && c<='Z' || c>='a' && c<='z';}

map< string , set< int > > mp;

int main(){

int line = 1 , i; char c;

scanf(" %c" , &c );

gets( s );

while( gets( s ) ){

for( i = 0 ; s[ i ] ; ++i ) if( !isLetter( s[ i ] ) ) s[ i ] = ' ';

stringstream ss( s );

while( ss>>s )

if( s[ 0 ] == c ) mp[ s ].insert( line );

line++;

}

for( map< string , set< int > > :: iterator it = mp.begin(); it != mp.end() ; ++it ){

printf("%s" , (it -> first).c\_str() );

for( set< int > :: iterator it2 = ( it ->second ).begin() ; it2 != ( it ->second ).end() ; ++it2 )

printf(" %d" , \*it2 );

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Network Connections

\*\*\*ID: 793

\*\*\*Juez: UVA

\*\*\*Tipo: Union Find

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define pb push\_back

#define mp make\_pair

#define x first

#define y second

#define pii pair<int , int >

#define psi pair<string , int>

#define all( x ) x.begin(),x.end()

int MIN( int a , int b ){ return ( a < b ) ? a : b; }

int MAX( int a , int b ){ return ( a < b ) ? b : a; }

string toStr(int n){string s;ostringstream buffer;buffer<<n;s=buffer.str();return s;}

int toInt(string str){int n;istringstream buffer(str);buffer>>n;return n;}

long long pot(long long a,long long b){if(!b)return 1;if(b&1)return a\*pot(a\*a,b>>1); else return pot(a\*a,b>>1);}

#define MAX\_SIZE 100005

int parent[ MAX\_SIZE ] , rank[ MAX\_SIZE ];

void Make\_Set( const int x ){

for( int i = 0 ; i < x ; ++i ){

parent[ i ] = i;

rank[ i ] = 0;

}

}

int Find( const int x ){

if( parent[ x ] != x ) parent[ x ] = Find( parent[ x ] );

return parent[ x ];

}

bool sameSet( int x , int y ){

return ( Find( x ) == Find( y ) );

}

void Union( const int x, const int y ){

int PX = Find( x ) , PY = Find( y );

if( rank[ PX ] > rank[ PY ] ) parent[ PY ] = PX;

else{

parent[ PX ] = PY;

if( rank[ PX ] == rank[ PY ] ) rank[ PY ]++;

}

}

char c[1000];

int main(){

string s;

int V , u , v , t;

scanf("%d" , &t );

for( int q = 0 ; q < t && scanf("%d" , &V) ; ++q ){

if( q )

printf("\n");

Make\_Set( V );

int pos , neg;

pos = neg = 0;

getline( cin , s );

while( getline( cin , s ) && s.length() > 0 ){

stringstream ss( s );

ss>>c>>u>>v;

if( c[0] == 'c' ){

Union( u - 1 , v - 1 );

}else if( c[0] == 'q'){

if( sameSet( u - 1 , v - 1) ){

pos++;

}else

neg++;

}

}

printf("%d,%d\n" , pos , neg );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Critical Links

\*\*\*ID: 796

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Bridges, Biconnected Components

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

int min( int a , int b ){ return ( a < b )? a : b ; }

int seen[ MAX ] , dfsNum[ MAX ] , low[ MAX ] , num , numBridges , V;

struct Bridges{

int x , y;

Bridges( int xx , int yy ): x( xx ), y( yy ){}

Bridges(){}

bool operator < ( const Bridges b ) const{

return x < b.x || ( x == b.x && y < b.y );

}

}bridges[ MAX ];

vector< int > ady[ MAX ];

void dfs( int x , int parent ){

dfsNum[ x ] = low[ x ] = num++;

seen[ x ] = 1;

for( int i = 0 ; i < ady[ x ].size() ; ++i ){

int y = ady[ x ][ i ];

if( !seen[ y ] ){

dfs( y , x );

if( dfsNum[ x ] < low[ y ]){

if( x < y ) bridges[ numBridges++ ] = Bridges( x , y );

else bridges[ numBridges++ ] = Bridges( y , x );

}

low[ x ] = min( low[ x ] , low[ y ] );

}

else if( y != parent ){

low[ x ] = min( dfsNum[ y ] , low[ x ] );

}

}

}

int main(){

int t , u , k , v;

while( scanf("%d" , &V ) != EOF ){

for( int i = 0 ; i < V ; ++i ){

scanf("%d (%d)" , &u , &k );

ady[ u ].clear();

while( k-- ){

scanf("%d" , &v );

ady[ u ].push\_back( v );

}

}

memset( seen , 0 , sizeof( seen ) );

numBridges = 0;

for( int i = 0 ; i < V ; ++i ){

if( !seen[ i ] ){

num = 0;

dfs( i , -1 );

}

}

printf("%d critical links\n" , numBridges );

sort( bridges , bridges + numBridges );

for( int i = 0 ; i < numBridges ; ++i ) printf("%d - %d\n" , bridges[ i ].x , bridges[ i ].y );

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Internet Bandwidth

\*\*\*ID: 820

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Max Flow(Edmonds Karp)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstdio>

#include <algorithm>

#include <vector>

#include <queue>

using namespace std;

/\*\*

u: nodo origen

v: nodo destino

c: capacidad de nodo u a v

r: indice de arista inversa, ejemplo si arista act 1->2 retorno es 1<-2

\*\*/

struct Edge{

int u;

int v;

int c;

int r;

Edge( int uu, int vv, int cc, int rr): u(uu), v(vv), c(cc), r(rr){}

};

vector< Edge > E;

vector< vector<int> > ady; //posee indices de aristas

int maxflow( int source , int sink ){

for( int f = 0; ;){

///vector de enlaces previos para ver la ruta recorrida del argumenting path

///tiene indices de aristas

vector<int> prev( ady.size() , -1 );

queue<int> Q;

Q.push( source );

while( !Q.empty() ){

int u = Q.front(); Q.pop();

for( int i = 0 ; i < ady[ u ].size(); ++i ){

///indice de arista actual

int e = ady[ u ][ i ];

int v = E[ e ].v;

///si no puede pasar o si ya fue visitado

if( E[ e ].c == 0 || prev[ v ] >= 0 )continue;

Q.push( v );

prev[ v ] = e;

}

}

///si ya no hay argumenting paths

if( prev[ sink ] < 0 )return f;

///hallamos minima capacidad de argumentin path

int c = 0x7fffffff;

for( int x = sink ; x != source ; x = E[ prev[ x ] ].u ){

c = min( c , E[ prev[ x ] ].c );//c <?= E[ prev[ x ] ].c;

}

///para nodos de ida 1->2 restamos capacidad minima

///para nodos de vuevla 2<-1 sumamos capacidad minima

for( int x = sink ; x != source ; x = E[ prev[ x ] ].u ){

E[ prev[ x ] ].c -= c;

E[ E[ prev[ x ] ].r ].c += c;

}

f += c;

}

}

///indices de aristas comienzan x cero

///si tengo 1->2 su indice es 0, si 1<-2 su indice es 1

///x tanto r de primero es 1, de segundo es 0

void add( int u , int v , int c , bool dirigido = true ){

Edge e1( u , v , c , E.size() + 1 );

ady[ u ].push\_back( E.size() );

E.push\_back( e1 );

///capadidad 0 si es dirigido

Edge e2( v , u , ( dirigido )? 0 : c , E.size() - 1 );

ady[ v ].push\_back( E.size() );

E.push\_back( e2 );

}

int main(){

int V, s, t, A, u , v, c, casos = 0;

while( scanf("%d", &V) && V ){

//if( casos )printf("\n");

E.clear();

ady = vector< vector<int> >( V + 1 , vector<int> () );

scanf("%d %d %d" , &s, &t ,&A);

while( A-- ){

scanf("%d %d %d", &u, &v , &c);

add( u , v , c , false );

}

printf("Network %d\n", ++casos);

printf("The bandwidth is %d.\n\n", maxflow( s , t ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Page Hopping

\*\*\*ID: 821

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

#define INF 1<<30

int ady[ MAX ][ MAX ], seen[ MAX ];

void init(){

for( int i = 0 ; i < MAX ; ++i )

for( int j = 0 ; j < MAX ; ++j )

ady[ i ][ j ] = ( i == j ) ? 0 : INF;

}

void floyd(){

for( int k = 0 ; k < MAX ; ++k ){

if( !seen[ k ] )continue;

for( int i = 0 ; i < MAX ; ++i ){

if( !seen[ i ] || ady[ i ][ k ] == INF ) continue;

for( int j = 0 ; j < MAX ; ++j ){

if( !seen[ j ] || ady[ k ][ j ] == INF ) continue;

ady[ i ][ j ] = ( ady[ i ][ j ] > ady[ i ][ k ] + ady[ k ][ j ] )?ady[ i ][ k ] + ady[ k ][ j ] : ady[ i ][ j ];

}

}

}

}

int main(){

int u , v , q = 1;

double sum , cnt;

while( scanf("%d %d" , &u , &v ) , u | v ){

memset( seen , 0 , sizeof( seen ) );

seen[ u ] = seen[ v ] = 1;

init();

ady[ u ][ v ] = 1;

while( scanf("%d %d" , &u , &v ) , u | v ){

ady[ u ][ v ] = 1;

seen[ u ] = seen[ v ] = 1;

}

floyd();

sum = cnt = 0.0;

for( int i = 0 ; i < MAX ; ++i ){

if( !seen[ i ] ) continue;

for( int j = 0 ; j < MAX ; ++j ){

if( !seen[ j ] || i == j || ady[ i ][ j ] == INF ) continue;

cnt++;

sum += ady[ i ][ j ];

}

}

printf("Case %d: average length between pages = %.3lf clicks\n" , q++ , sum/cnt );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Walking on the Safe Side

\*\*\*ID: 825

\*\*\*Juez: UVA

\*\*\*Tipo: DAG Longest Path DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAX 105

int h , w;

int ady[ MAX ][ MAX ];

int dp[ MAX ][ MAX ];

int dx[] = { 0 , 1 };

int dy[] = { 1 , 0 };

void solve(){

int i , j , ans ,ny , nx , k;

memset( dp , 0 , sizeof( dp ) );

dp[ 0 ][ 0 ] = 1;

for( i = 0 ; i < h ; ++i ){

for( j = 0 ; j < w ; ++j ){

if( ady[ i ][ j ] ) continue;

for( k = 0 ; k < 2 ; ++k ){

nx = dx[ k ] + i; ny = dy[ k ] + j;

if( nx >= 0 && ny >= 0 && nx < h && ny < w && !ady[ nx ][ ny ] ){

dp[ nx ][ ny ] += dp[ i ][ j ];

}

}

}

}

printf("%d\n" , dp[ h - 1 ][ w - 1 ] );

}

char s[ MAX ];

int main(){

int t , y ,i , q , b;

scanf("%d" , &t );

for( q = 0 ; q < t ; ++q ){

if( q ) printf("\n");

scanf("%d %d" , &h , &w );

getchar();

memset( ady , 0 , sizeof( ady ) );

for( i = 0 ; i < h && gets( s ) ; ++i ){

b = 1;

char \*p = strtok( s , " " );

while( p != NULL ){

if( !b )ady[ i ][ atoi( p ) - 1 ] = 1;

p = strtok( NULL , " ");

b = 0;

}

}

solve();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Water Falls

\*\*\*ID: 833

\*\*\*Juez: UVA

\*\*\*Tipo: Slopes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <algorithm>

using namespace std;

#define EPS 1e-9

const long double PI = 2 \* acos(.0);

struct Point{

double x , y;

Point( double X , double Y ):x(X) , y(Y){}

bool operator <( Point other ) const{

if( fabs( x - other.x ) < EPS ){

return y + EPS < other.y;

}

return x + EPS < other.x;

}

bool operator ==( Point other )const{

return ( fabs( x - other.x ) < EPS && fabs( y - other.y ) < EPS );

}

Point operator -(Point other)const{

return Point( x - other.x , y - other.y );

}

Point(){}

};

typedef Point Vector;

double dot( Vector v1 , Vector v2 ){

return v1.x \* v2.x + v1.y \* v2.y;

}

double cross( Vector v1 , Vector v2 ){

return v1.x \* v2.y - v1.y \* v2.x;

}

Vector scale( Vector v , double s ){

return Vector( v.x \* s , v.y \* s );

}

Vector toVec( Point p1 , Point p2 ){

return p2 - p1;

}

Point translate( Point p , Vector v ){

return Point( p.x + v.x , p.y + v.y );

}

double norm( Vector v ){

return v.x \* v.x + v.y \* v.y;

}

double dist( Point p1 , Point p2 ){

return hypot( p1.x - p2.x , p1.y - p2.y );

}

double distToLine( Point a , Point b , Point p , Point &c ){

Vector AB = toVec( a , b ) , AP = toVec( a , p );

double s = dot( AB , AP )/norm( AB );

c = translate( a , scale( AB , s ) );

return dist( p , c );

}

double distToLineSeg( Point a , Point b , Point p , Point &c ){

Vector AB = toVec( a , b ) , AP = toVec( a , p );

double s = dot( AB , AP )/norm( AB );

if( s < 0.0 ){

c = Point( a.x , a.y );

return dist( a , p );

}

if( s > 1.0 ){

c = Point( b.x , b.y );

return dist( b , p );

}

return distToLine( a , b , p , c );

}

#define MAX 1005

struct Segment{

Point p1 , p2;

Segment( Point P1 , Point P2 ):p1( P1 ) , p2(P2){}

Segment(){}

}s[ MAX ];

int n;

bool ccw( Point p , Point q , Point o ){

Vector OP = toVec( o , p ), OQ = toVec( o , q );

return ( cross( OP , OQ ) + EPS < 0 );

}

int solve( int x , int y ){

while( true ){

double mini = 1LL<<50;

int idx = -1;

for( int i = 0 ; i < n ; ++i ){

if( s[ i ].p1.x <= x && x <= s[ i ].p2.x && ( ccw( Point( x , y ) , s[i].p2 , s[i].p1 ) ) ){

Point p;

double d = distToLine( s[ i ].p1 , s[ i ].p2 , Point( x , y ) , p );

if( d + EPS < mini ){

mini = d;

idx = i;

}

}

}

if( idx == -1 ) break;

if( s[ idx ].p1.y > s[ idx ].p2.y ){

x = s[ idx ].p2.x;

y = s[ idx ].p2.y;

}

else{

x = s[ idx ].p1.x;

y = s[ idx ].p1.y;

}

}

return x;

}

int main(){

int t , q , x1 , y1 , x2 ,y2 , m , x , y;

scanf("%d" , &t );

for( q = 0 ; q < t && scanf("%d" , &n ) ; ++q ){

if( q )puts("");

for( int i = 0 ; i < n && scanf("%d %d %d %d", &x1 , &y1 , &x2 , &y2 ) ; ++i ){

if( x1 < x2 )

s[ i ] = Segment( Point( x1 , y1 ) , Point( x2 , y2 ) );

else

s[ i ] = Segment( Point( x2 , y2 ) , Point( x1 , y1 ) );

}

scanf("%d" , &m );

for( int i = 0 ; i < m && scanf("%d %d" , &x , &y ) ; ++i ){

printf("%d\n" , solve( x , y ) );

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Continued Fractions

\*\*\*ID: 834

\*\*\*Juez: UVA

\*\*\*Tipo: Brute Force

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

map< int , map< int , int > > path;

map< int , map< int , bool > > entro;

void solve( int num , int denom , int len ){

if( denom == 0 ) return;

int i = 0;

if( len > 0 ) i = 1;

for( ; i \* denom <= num ; ++i ){

path[ num ][ denom ] = i;

entro[ num ][ denom ] = 1;

solve( denom , num - i \* denom , len + 1 );

}

}

void print( int num , int denom , int len ){

if( !entro[ num ][ denom ] ) return;

if( len == 1 ) printf(";");

else if( len > 1 ) printf(",");

printf("%d" , path[ num ][ denom ] );

print( denom , num - path[ num ][ denom ] \* denom , len + 1 );

}

int main(){

int num , denom;

while( scanf("%d %d" , &num , &denom ) != EOF ){

if( num == 1 ){

printf("[0;%d]\n" , denom );

continue;

}

printf("[");

entro.clear();

path.clear();

solve( num , denom , 0 );

print( num , denom , 0 );

printf("]\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Largest Submatrix

\*\*\*ID: 836

\*\*\*Juez: UVA

\*\*\*Tipo: DP, Maximum 2D sum

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 26

int a[ MAX ][ MAX ], sum , s , maxi, n;

int pr[ MAX ];

int Kadane2D(){

maxi = 0;

int resp = 0;

int k, l , x1 = 0 , x2 = 0 , y1 = 0, y2 = 0, j;

for( int z = 0 ; z < n; ++z ){

for( int i = 0 ; i < n ; ++i ) pr[ i ] = 0;

for( int x = z ; x < n ; ++x ){

s = 1<<31;

sum = 0;

for( int i = 0 ; i < n ; ++i ){

pr[ i ] += a[ x ][ i ];

sum += pr[ i ];

if( sum > s ){

s = sum;

}

if( sum < 0 ){

sum = 0;

}

}

if( s > maxi ){

maxi = s;

}

}

}

return maxi;

}

int main(){

int t , l;

char line[ MAX ];

scanf("%d\n" , &t );

gets( line );

for( int q = 0 ; q < t ; ++q ){

if( q )putchar('\n');

n = 0;

while( gets( line ) ){

l = strlen( line );

if( l == 0 )break;

for( int j = 0 ; j < l ; ++j ){

if( line[ j ] == '0' )a[ n ][ j ] = -1005;

else a[ n ][j ] = 1;

}

++n;

}

printf("%d\n" , Kadane2D() );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Light and Transparencies

\*\*\*ID: 837

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

#include<vector>

#include<algorithm>

using namespace std;

struct Segment{

double xmin;

double xmax;

double r;

Segment(double x1 , double y1 , double r1 ): xmin( x1 ),xmax( y1 ),r( r1 ){}

};

//a q segmento pertenece si son dos multiplico

double segment( vector<Segment> s , double x ){

double value = 1.0;

for(int i = 0 ; i < s.size() ; ++i )

if( s[ i ].xmin <= x && x < s[ i ].xmax ) value \*= s[ i ].r;

return value;

}

int main(){

int casos , n;

double x1 , y1 , x2 , y2 , r;

scanf( "%d" , &casos );

for( int q = 0 ; q < casos ; ++q ){

if( q ) putchar('\n');

scanf( "%d" , &n );

vector<Segment> p;

vector<double> v;

while( n-- ){

scanf( "%lf %lf %lf %lf %lf" , &x1 , &y1 , &x2 , &y2 , &r );

if( x2 < x1 )swap( x1 , x2 );

Segment p1( x1 , x2 , r );

p.push\_back( p1 );

v.push\_back( x1 );

v.push\_back( x2 );

}

//obtengo puntos ordenados para recorrerlos y ver si

//pertenecen a segmento determinado

sort( v.begin() , v.end() );

printf( "%d\n" , v.size() + 1 );

printf("-inf %.3lf 1.000\n" , v[ 0 ] );

for(int i = 0 ; i < v.size() - 1 ; ++i ){

printf( "%.3lf %.3lf %.3lf\n" , v[ i ] , v[ i + 1 ] , segment( p , v[ i ] ) );

}

printf( "%.3lf +inf 1.000\n" , v[ v.size() - 1 ] );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# A Multiplication Game

\*\*\*ID: 847

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int n;

while(cin>>n){

bool stan=true;

int p=1,last=1;

while(1){

if(stan){p\*=9;last=1;stan=false;}

else {p\*=2;stan=true;last=2;}

if(n<=p)break;

}

if(last==1)cout<<"Stan wins."<<endl;

else cout<<"Ollie wins."<<endl;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Deciding victory in Go

\*\*\*ID: 852

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Flood Fill

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 130

char ady[ MAX ][ MAX ];

int blanco, negro, cnt;

void dfs( int x , int y ){

if( x < 0 || y < 0 || x >= 9 || y >= 9 || ady[ x ][ y ] == '#') return;

if( ady[ x ][ y ] == 'O' ){ blanco++; return;}

else if( ady[ x ][ y ] == 'X'){ negro++; return; }

ady[ x ][ y ] = '#';

cnt++;

dfs( x + 1 , y );

dfs( x - 1 , y );

dfs( x , y + 1 );

dfs( x , y - 1 );

}

int main(){

int t, B , N;

scanf("%d" , &t );

while( t-- ){

for( int i = 0 ; i < 9 ; ++i )scanf( "%s" , ady[ i ] );

B = N = 0;

for( int i = 0 ; i < 9 ; ++i ){

for( int j = 0 ; j < 9 ; ++j ){

if( ady[ i ][ j ] == '.'){

blanco = negro = 0; cnt = 0;

dfs( i , j );

if( blanco == 0 && negro > 0 ) N += cnt;

else if( negro == 0 && blanco > 0 ) B += cnt;

}

else if( ady[ i ][ j ] == 'O') B++;

else if( ady[ i ][ j ] == 'X') N++;

}

}

printf("Black %d White %d\n" , N , B );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Lunch in Grid City

\*\*\*ID: 885

\*\*\*Juez: UVA

\*\*\*Tipo: Sorting

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdlib.h>

#define MAX 50005

int cmp (const void \* a, const void \* b){

return ( \*(int\*)a - \*(int\*)b );

}

int main(){

int t , h , w , k , x[ MAX ] , y[ MAX ];

scanf("%d" , &t );

while( t-- ){

scanf("%d %d %d" , &h , &w , &k );

for( int i = 0 ; i < k ; ++i )

scanf("%d %d" , &x[ i ] , &y[ i ] );

qsort( x , k , sizeof( int ) , cmp );

qsort( y , k , sizeof( int ) , cmp );

if( k % 2 == 0 ){

printf("(Street: %d, Avenue: %d)\n" ,x[ ( k/ 2 ) - 1] , y[ ( k / 2 ) - 1 ] );

}

else{

printf("(Street: %d, Avenue: %d)\n" ,x[ ( ( k + 1 )/ 2 ) - 1] , y[ ( ( k + 1 ) / 2 ) - 1 ] );

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Substitution Cipher

\*\*\*ID: 865

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, String manipulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

using namespace std;

int main(){

string s;

string abc,rabc;

int t , i , l;

cin>>t;

getline( cin , s );

cin.get();

bool b = false;

while(t-- ){

if( b )cout<<endl;

b=true;

getline(cin,abc);

getline(cin,rabc);

cout<<rabc<<endl<<abc<<endl;

while( getline( cin , s ) && s[0] !='\0'){

l = s.size();

string s2 = "";

for( i = 0 ; i < l ; i++ ){

int t = abc.find( s[ i ] );

if( t != string::npos ){

s2 += rabc[ (int) t];

}

else s2 += s[ i ];

}

cout<<s2<<endl;

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Numerical maze

\*\*\*ID: 868

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking | DFS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 55

int h , w , ady[ MAX ][ MAX ];

struct Node{

int x , y , val;

Node( int xx , int yy , int v ): x( xx ) , y( yy ) , val( v ){}

};

//bool seen[ MAX ][ MAX ];

int dx[] = { 1 , 0 , -1 , 0 };

int dy[] = { 0 , 1 , 0 , -1 };

bool bfs( int xini , int yini ){

queue<Node> Q;

//memset( seen , 0 ,sizeof( seen ) );

Q.push( Node( xini , yini , 1 ) );

int ymini = MAX;

while( !Q.empty() ){

Node act = Q.front(); Q.pop();

if( act.x == h - 1 ){

ymini = min( act.y , ymini );

continue;

}

//if( seen[ act.x ][ act.y ] ) continue;

for( int i = 0 ; i < 4 ; ++i ){

int nx = dx[ i ] + act.x, ny = dy[ i ] + act.y;

if( nx >= 0 && nx < h && ny >= 0 && ny < w ){

if( ady[ act.x ][ act.y ] == act.val ){

if( ady[ nx ][ ny ] == 1 ){

Q.push( Node( nx , ny , act.val + 1 ) );

}

}

else if( ady[ act.x ][ act.y ] < act.val ){

if( ady[ nx ][ ny ] == ady[ act.x ][ act.y ] + 1 ){

Q.push( Node( nx , ny , act.val ) );

}

}

}

}

}

if( ymini != MAX ){ printf("%d %d\n%d %d\n" , xini + 1 , yini + 1 , h , ymini + 1 ); return true;}

return false;

}

int main(){

int t;

scanf("%d" , &t );

for( int q = 0 ; q < t ; ++q ){

if( q ) putchar('\n');

scanf("%d %d" , &h ,&w );

for( int i = 0 ; i < h ; ++i ){

for( int j = 0 ; j < w ; ++j ){

scanf("%d" , &ady[ i ][ j ] );

}

}

for( int i = 0 ; i < w ; ++i ){

if( ady[ 0 ][ i ] == 1 && bfs( 0 , i )){

break;

}

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Airline Comparison

\*\*\*ID: 869

\*\*\*Juez: UVA

\*\*\*Tipo: Floyd Warshall Transitive Closure

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <iostream>

#define MAX 26

int ady[ MAX ][ MAX ] , ady2[ MAX ][ MAX ];

void Init(){

for( int i = 0 ; i < MAX ; ++i ){

for( int j = 0 ; j < MAX ; ++j ){

ady[ i ][ j ] = ady2[ i ][ j ] = 0;

}

}

}

void TransitiveClosure(){

for( int k = 0 ; k < MAX ; ++k ){

for( int i = 0 ; i < MAX ; ++i ){

for( int j = 0 ; j < MAX ; ++j ){

ady[ i ][ j ] |= ady[ i ][ k ] & ady[ k ][ j ];

ady2[ i ][ j ] |= ady2[ i ][ k ] & ady2[ k ][ j ];

}

}

}

}

int main(){

int t ,E;

char u , v;

scanf("%d" ,&t );

for( int q = 0 ; q < t ; ++q ){

if( q )putchar('\n');

scanf("%d" ,&E );

Init();

while( E-- ){

scanf("\n%c %c", &u , &v );

u = toupper( u ) ; v = toupper( v );

ady[ u - 'A' ][ v - 'A' ] = ady[ v - 'A' ][ u - 'A' ] = 1;

}

scanf("%d" ,&E );

while( E-- ){

scanf("\n%c %c", &u , &v );

u = toupper( u ); v = toupper( v );

ady2[ u - 'A' ][ v - 'A' ] = ady2[ v - 'A' ][ u - 'A' ] = 1;

}

TransitiveClosure();

int i , j;

for( i = 0 ; i < MAX ; ++i ){

for( j = 0 ; j < MAX ; ++j ){

if( ady[ i ][ j ] != ady2[ i ][ j ] ) break;

}

if( j < MAX )break;

}

if( i < MAX )printf("NO\n");

else printf("YES\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Counting Cells in a Blob

\*\*\*ID: 871

\*\*\*Juez: UVA

\*\*\*Tipo: Graph, Flood Fill

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 26

int ady[ MAX ][ MAX ];

int seen[ MAX ][ MAX ];

int h , w, cnt;

int dx[ 8 ] = { 1 , -1 , 0 , 0 , 1 , 1 , -1 , -1 };

int dy[ 8 ] = { 0 , 0 , 1 , -1 , 1 , -1 , 1 , -1};

void dfs( int x , int y ){

seen[ x ][ y ] = 1;

cnt++;

for(int i =0 ; i < 8 ; ++i ){

int nx = dx[ i ] + x, ny = dy[ i ] + y;

if( nx >= 0 && ny >= 0 && nx < h && ny < w && !seen[ nx ][ ny ] && ady[ nx ][ ny ] == 1 ){

dfs( nx , ny );

}

}

}

int main(){

int t, maxi;

scanf("%d" , &t );

char s[ MAX ];

gets( s );

gets( s );

for( int q = 0 ; q < t ; ++q ){

if( q )putchar('\n');

h = 0;

maxi = 0;

while( gets( s ) ){

int l = strlen( s );

if( l == 0 )break;

w = l;

for( int i = 0 ; i < w ; ++i ){

if( s[ i ] == '1' )ady[ h ][ i ] = 1;

else ady[ h ][ i ] = 0;

}

h++;

}

memset( seen , 0 , sizeof( seen ) );

for( int i = 0 ; i < h ; ++i ){

for( int j = 0 ; j < w ; ++j ){

if( ady[ i ][ j ] == 1 && ! seen[ i ][ j ] ){

cnt = 0;

dfs( i , j );

if( cnt > maxi )maxi = cnt;

}

}

}

printf("%d\n" , maxi );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Ordering

\*\*\*ID: 872

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Topological Sort, Backtracking

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

#include <sstream>

using namespace std;

#define MAX 35

int used[ MAX ];

///reglas representadas con entero a < b -> [ 0 ][ 1 ] = true

int contraint[ MAX ][ MAX ];

char let[ MAX ];

int idx;

char output[ MAX + 5 ];

bool check( int len , int in ){

//si x ejemplo tengo a b f g

//contrait a <b b < f

//primero entra "a" y se compara con "b" "f" y "g"

//contradiccion al contrait es q a > b x ello output va en indice j

for( int i = 0 ; i < len ; ++i ){

if( contraint[ in ][ output[ i ] - 'A' ] ) return false;

}

return true;

}

bool possible;

void dfs( int len ){

if( len == idx ){

possible = true;

putchar( output[ 0 ] );

for( int i = 1 ; i < len ;++i ){

printf(" %c", output[ i ] );

}

putchar( '\n' );

//output[ len ] ='\0';

//printf("%s\n", output );

return;

}

for( int i = 0 ; i < idx ; ++i ){

if( !used[ i ] && check( len , let[ i ] - 'A' ) ){

output[ len ] = let[ i ];

used[ i ] = 1;

dfs( len + 1 );

used[ i ] = 0;

}

}

}

int main(){

char line[ MAX + 5 ];

char a ,b;

int t, l;

bool bb = false;

scanf("%d", &t);

gets( line );

while( t-- ){

gets( line );

gets( line );

if( bb ) putchar( '\n' );

bb = true;

idx = 0;

for( int i = 0 ; line[ i ] ; ++i )

if( isalpha( line[ i ] ) ) { let[ idx ] = line[ i ]; used[ idx++ ] = 0;}

///se ordena para sacar lexicograficamente

sort( let, let + idx );

memset( contraint , 0 , sizeof( contraint ) );

gets( line );

l = strlen( line );

for( int i = 0 ; i < l ; ++i ){

if( line[ i ] == '<' )line[ i ] = ' ';

}

line[ l ] = ' ';

line[ l + 1 ] = '\0';

stringstream ss( line );

while( ss>>a>>b ) contraint[ a - 'A' ][ b - 'A' ] = 1;

possible = false;

dfs( 0 );

if( !possible ) printf("NO\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Factorial Factors

\*\*\*ID: 884

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory, sieve(modified)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

using namespace std;

#define MAX 1000001

int tabla[MAX];

int aux[MAX];

void solve(){

tabla[1]=aux[1]=0;

tabla[2]=aux[2]=1;

int i,j,n,tmp;

for(i=3;i<MAX;i++){

n=i;

tmp=1;

for(j=2;j\*j<=n;j++){

if(n%j==0){

tmp=j;

break;

}

}

if(tmp==1)aux[i]=1;

else{

aux[i]=aux[tmp]+aux[n/tmp];

}

tabla[i]=tabla[i-1]+aux[i];

}

}

int main(){

solve();

int n;

while(scanf("%d",&n)!=EOF){

printf("%d\n",tabla[n]);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Word Problem

\*\*\*ID: 895

\*\*\*Juez: UVA

\*\*\*Tipo: Sort, String

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1005

bool isLetter(char c){return c>='A' && c<='Z' || c>='a' && c<='z';}

char s[ 55 ];

int a[ MAX ][ 27 ] ,aux[ 27 ];

int main(){

int i , len = 0 , j , cnt;

while( scanf("%s", &s ) , s[ 0 ] != '#'){

for( i = 0 ; s[ i ] ; ++i ) a[ len ][ s[ i ] - 'a' ]++;

len++;

}

gets( s );

while( gets( s ) , s[ 0 ] != '#' ){

memset( aux , 0 , sizeof( aux ) );

for( i = 0 ; s[ i ] ; ++i ){

if( isLetter( s[ i ] ))aux[ s[ i ] - 'a' ]++;

}

cnt = 0;

for( i = 0 ; i < len ; ++i ){

for( j = 0 ; j < 26 ; ++j ){

if( a[ i ][ j ] > aux[ j ] ) break;

}

if( j == 26 ) cnt++;

}

printf("%d\n" , cnt );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Brick Wall Patterns

\*\*\*ID: 900

\*\*\*Juez: UVA

\*\*\*Tipo: Fibonacci Numbers, math

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int n;

long long dp[51];

dp[0]=1;

dp[1]=1;

for(int i=2;i<=50;i++){

dp[i]=dp[i-1]+dp[i-2];

}

while(scanf("%d",&n) && n){

printf("%lld\n",dp[n]);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Password search

\*\*\*ID: 902

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, string, suffix array , KMP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstdio>

#include <cassert>

#include <algorithm>

#include <iostream>

#include <map>

using namespace std;

int main()

{

int N;

string ss;

while (cin >> N >> ss){

map<string,int> times;

int best = 0;

string ans;

for (int i=0; i<=ss.size()-N; i++){

string s = ss.substr(i, N);

int tt = ++times[ s ];

if ( tt > best ){

best = tt;

ans = s;

}

}

cout << ans << endl;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Winterim Backpacking Trip

\*\*\*ID: 907

\*\*\*Juez: UVA

\*\*\*Tipo: Binary Search | DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <string.h>

#define MAX 610

int d[ MAX ] , n , k;

int possible( int val ){

int i, sum = 0 , days = 0;

for( i = 0 ; i <= n ; ++i ){

if( d[ i ] > val ) return 0;

if( sum + d[ i ] > val ){

sum = 0;

days++;

}

sum += d[ i ];

}

return days <= k;

}

int main(){

int left , right , mid , ans, aux , i;

while( scanf("%d %d" , &n , &k ) == 2 ){

for( i = 0 ; i <= n && scanf("%d" , &d[ i ] ) == 1 ; ++i );

left = 0;

right = 1<<30;

while( left < right ){

mid = ( left + right )>>1;

if( possible( mid ) ){

right = mid;

}

else left = mid + 1;

}

printf("%d\n" , right );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Re-connecting Computer Sites

\*\*\*ID: 908

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, MST, Kruskal

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1000005

int uf[ MAX ];

void makeSet(int V ){

for( int i = 1 ; i <= V ; ++i )

uf[i] = i;

}

int Find( int x ){

return ( uf[ x ] == x ) ? x : uf[ x ] = Find( uf[ x ] );

}

void Union( int x , int y ){

uf[ Find( x ) ] = Find( y );

}

struct Arista{

int u , v , w;

Arista( int uu , int vv , int ww ): u( uu ) , v( vv ) , w( ww ) {}

Arista() {}

bool operator <( const Arista &a ) const{

return w < a.w;

}

} graph[ MAX ];

int main(){

int u,v , c , T, ans1, ans2, V, K,M;

bool first = true;

while( scanf("%d" , &V) != EOF){

if( !first ) printf("\n");

first = false;

ans1 = ans2 = 0;

for( int i = 0 ; i < V - 1 && scanf("%d %d %d" , &u , &v , &c ) ; ++i ){

ans1 += c;

}

scanf("%d" , &K );

for( int i = 0 ; i < K && scanf("%d %d %d" , &graph[i].u, &graph[i].v, &graph[i].w) ; ++i);

scanf("%d" , &M );

for( int i = 0 ; i < M && scanf("%d %d %d" , &graph[i+K].u, &graph[i+K].v, &graph[i+K].w) ; ++i);

int E = K + M;

makeSet(V);

sort( graph, graph + E );

for( int i = 0 ; i < E ; ++i ){

int u = graph[i].u;

int v = graph[i].v;

int c = graph[i].w;

if( Find(u) != Find(v) ){

Union(u,v);

ans2 += c;

}

}

printf("%d\n%d\n", ans1, ans2 );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# TV game

\*\*\*ID: 910

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <string.h>

#define MAX 27

int V , m, final[ MAX ];

int ady[ MAX ][ 2 ];

long long dp[ MAX ][ 35 ];

long long solve( int cur , int len ){

if( len == m ){

if( final[ cur ] ) return 1;

return 0;

}

if( dp[ cur ][ len ] != -1 ) return dp[ cur ][ len ];

return dp[ cur ][ len ] = solve( ady[ cur ][ 0 ] , len + 1 ) + solve( ady[ cur ][ 1 ] , len + 1 );

}

int main(){

char u , v0 , v1 , x , i;

while( scanf("%d" , &V ) != EOF ){

memset( ady , 0 , sizeof( ady ) );

for( i = 0 ; i < V && scanf(" %c %c %c %c" , &u , &v0 , &v1 , &x ) == 4 ; ++i ){

ady[ u - 'A' ][ 0 ] = v0 - 'A';

ady[ u - 'A' ][ 1 ] = v1 - 'A';

( x == 'x' )?final[ u - 'A' ] = 1:final[ u - 'A'] = 0;

}

scanf("%d" , &m );

memset( dp , -1 , sizeof( dp ) );

printf( "%lld\n" , solve( 0 , 0 ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Joana and the Odd Numbers

\*\*\*ID: 913

\*\*\*Juez: UVA

\*\*\*Tipo: Recurrences

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main(){

long long n , pos;

while( scanf("%lld" , &n ) != EOF ){

pos = ( ( n + 1 )/2 ) \* ( n + 1 )/2;

printf("%lld\n" , pos \* pos - ( pos - 3 ) \* ( pos - 3 ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Jumping Champion

\*\*\*ID: 914

\*\*\*Juez: UVA

\*\*\*Tipo: Number theory, sieve, linear search

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1000007

int prime[MAX];

int primos[MAX];

int tam;

///sieve

void sieve(){

prime[0]=prime[1]=0;

for(int i = 2; i < MAX; ++i){

prime[i]=1;

}

for(int i=2;i\*i<MAX;i++){

if(prime[i]==1){

for(int k=2;k\*i<MAX;k++){

prime[k\*i]=0;

}

}

}

int k=0;

for(int i=2;i<MAX;i++){

if(prime[i])primos[k++]=i;

}

tam=k;

}

int main(){

sieve();

int casos,L,U,num,cont,dif,idx;

scanf("%d",&casos);

while(casos--){

scanf("%d %d",&L,&U);

int ini;

cont=0;

///linear search

for(ini=0;ini<tam;ini++){

if(primos[ini]>=L){

break;

}

}

int diff[1004];

for(int i=ini;i<tam;i++){

if(primos[i]>U)break;

cont++;

if(i<tam-1 && primos[i+1]<=U)diff[primos[i+1]-primos[i]]++;

}

int max=0;

for(int i=0;i<1004;i++){

if(diff[i]>max){max=diff[i];num=i;}

}

bool b=false;

for(int i=0;i<1004;i++){

if(i!=num && diff[i]==max){b=true;break;}

}

if(cont<2 || b)printf("No jumping champion\n");

else printf("The jumping champion is %d\n",num);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Sunny Mountains

\*\*\*ID: 920

\*\*\*Juez: UVA

\*\*\*Tipo: Distance Point to Line

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define EPS 1e-9

#define MAX 105

const long double PI = 2 \* acos(.0);

struct Point{

double x , y;

Point( double X , double Y ):x(X) , y(Y){}

bool operator <( Point other ) const{

if( fabs( x - other.x ) < EPS ){

return y + EPS < other.y;

}

return x + EPS < other.x;

}

bool operator ==( Point other )const{

return ( fabs( x - other.x ) < EPS && fabs( y - other.y ) < EPS );

}

Point operator -(Point other)const{

return Point( x - other.x , y - other.y );

}

Point(){}

}p[ MAX ];

typedef Point Vector;

double dot( Vector v1 , Vector v2 ){

return v1.x \* v2.x + v1.y \* v2.y;

}

double cross( Vector v1 , Vector v2 ){

return v1.x \* v2.y - v1.y \* v2.x;

}

Vector scale( Vector v , double s ){

return Vector( v.x \* s , v.y \* s );

}

Vector toVec( Point p1 , Point p2 ){

return p2 - p1;

}

Point translate( Point p , Vector v ){

return Point( p.x + v.x , p.y + v.y );

}

double norm( Vector v ){

return v.x \* v.x + v.y \* v.y;

}

double dist( Point p1 , Point p2 ){

return hypot( p1.x - p2.x , p1.y - p2.y );

}

double distToLine( Point a , Point b , Point p , Point &c ){

Vector AB = toVec( a , b ) , AP = toVec( a , p );

double s = dot( AB , AP )/norm( AB );

c = translate( a , scale( AB , s ) );

return dist( p , c );

}

double distToLineSeg( Point a , Point b , Point p , Point &c ){

Vector AB = toVec( a , b ) , AP = toVec( a , p );

double s = dot( AB , AP )/norm( AB );

if( s < 0.0 ){

c = Point( a.x , a.y );

return dist( a , p );

}

if( s > 1.0 ){

c = Point( b.x , b.y );

return dist( b , p );

}

return distToLine( a , b , p , c );

}

bool collinear( Point p , Point q , Point o ){

Vector OP = toVec( o , p ), OQ = toVec( o , q );

return fabs( cross( OP , OQ ) ) < EPS;

}

struct Line{

double a , b , c;

Line( double A , double B , double C ): a(A),b(B),c(C){}

Line(){}

};

bool lineIntersection( Line l1 , Line l2 , Point &p ){

double det = l1.a \* l2.b - l2.a \* l1.b;

if( fabs( det ) < EPS ) return false;

p.x = (l1.c \* l2.b - l2.c \* l1.b)/det;

p.y = (l1.a \* l2.c - l2.a \* l1.c)/det;

if( fabs(p.x) < EPS ) p.x = 0.0;

if( fabs(p.y) < EPS ) p.y = 0.0;

return true;

}

void pointToLine( Point p1 , Point p2 , Line &l ){

l.a = (p2.y - p1.y);

l.b = (p1.x - p2.x);

l.c = l.a \* p1.x + l.b \* p1.y;

}

int main(){

int t , n;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d" , &n ) ){

for( int i = 0 ; i < n && scanf("%lf %lf" , &p[ i ].x , &p[ i ] .y ) ; ++i );

sort( p , p + n );

Point ini = p[ n - 1 ];

double ans = 0.0;

for( int i = n - 1 ; i >= 1 ; --i ){

if( ini.y + EPS < p[ i - 1 ].y ){

ans += dist( ini , p[ i - 1 ] );

ini = p[ i - 1 ];

}

else if( i - 2 >= 0 && ini.y + EPS < p[ i - 2 ].y ){

Point c;

Line l1; pointToLine( p[ i - 1 ] , p[ i - 2 ] , l1);

Line l2; pointToLine( ini , Point( 0 , ini.y ) , l2);

lineIntersection( l1 , l2 , c );

ans += dist( c , p[ i - 2 ] );

ini = p[ i - 2 ];

}

}

printf("%.2lf\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Spreading the News

\*\*\*ID: 924

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, BFS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 2505

vector<int> ady[ MAX ];

int V;

struct Estado{

int x;

int d;

Estado( int xx ,int dd):x(xx) , d(dd){}

};

bool seen[ MAX ];

int maxi , day , boom[ MAX ];

void bfs(int x ){

memset( seen , 0 , sizeof( seen ) );

memset( boom , 0 , sizeof( boom ) );

queue< Estado > Q;

Q.push( Estado( x , 0 ) );

maxi = -1; day = -1;

int cnt;

seen[ x ] = 1;

while( !Q.empty() ){

Estado act = Q.front(); Q.pop();

boom[ act.d ]++;

if( act.d > 0 && boom[ act.d ] > maxi ){

maxi = boom[ act.d ];

day = act.d;

}

for( int i = 0 ; i < ady[ act.x ].size(); ++i ){

int v = ady[ act.x ][ i ];

if( !seen[ v ] ){

seen[ v ] = true;

Q.push( Estado( v , act.d + 1) );

}

}

}

}

int main(){

int source , k, v;

scanf("%d" , &V);

for( int i = 0 ; i < V ; ++i )ady[ i ].clear();

for( int u = 0 ; u < V ; ++u ){

scanf("%d", &k);

while( k-- ){

scanf("%d" ,&v );

ady[ u ].push\_back( v );

}

}

scanf("%d",&k);

while( k-- ){

scanf("%d",&source);

if( ady[ source ].size() == 0 ){

printf("0\n");

continue;

}

bfs( source );

printf("%d %d\n", maxi , day );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# No more prerequisites, please!

\*\*\*ID: 925

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Transitive Closure

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

#include <cstring>

#define MAX 130

int ady[ MAX ][ MAX ] , V;

char cities[ MAX ][ 30 ];

int get\_Id( char \*s ){

for( int i = 0 ; i < V ; ++i ){

if( strcmp( cities[ i ] , s ) == 0 )return i;

}

}

void Init(){

for( int i = 0 ; i < MAX ; ++i ){

for( int j = 0 ; j < MAX ; ++j ){

ady[ i ][ j ] = 0;

}

}

}

void TransitiveClosure(){

for( int k = 0 ; k < MAX ; ++k ){

for( int i = 0 ; i < MAX ; ++i ){

for( int j = 0 ; j < MAX ; ++j ){

ady[ i ][ j ] |= ady[ i ][ k ] & ady[ k ][ j ];

}

}

}

}

int f( const void \* v1 , const void \*v2){

return strcmp( ( char \*)v1 , (char \*)v2 );

}

int main(){

int t , E , k, uu ,vv;

char u[ 30 ] , v[ 30 ];

scanf("%d" ,&t );

while( t-- ){

scanf("%d" , &V );

for( int i = 0 ; i < V ; ++i ){

scanf("%s" , &cities[ i ] );

}

qsort( cities , V , sizeof( cities[ 0 ] ) , f );

Init();

scanf("%d" , &E );

while( E-- ){

scanf("%s %d" , &u , &k );

uu = get\_Id( u );

while( k-- ){

scanf("%s" ,&v );

vv = get\_Id( v );

ady[ vv ][ uu ] = 1;

}

}

TransitiveClosure();

int possible[ MAX ], idx, cont;

bool print[ MAX ];

for( int i = 0 ; i < V ; ++i ){

idx = 0;

for( int j = 0 ; j < V ; ++j ){

///posibles prerequisitos

if( ady[ j ][ i ] == 1 ){

print[ idx ] = 1;

possible[ idx++ ] = j;

}

}

for( int j = 0 ; j < idx ; ++j ){

for( int k = 0 ; k < idx ; ++k ){

if( ady[ possible[ j ] ][ possible[ k ] ] )print[ j ] = 0;

}

}

if( idx != 0 ){

cont = 0;

for( int j = 0 ; j < idx ; ++j )if( print[ j ] )cont++;

printf("%s %d" , cities[ i ] , cont );

for( int j = 0 ; j < idx ; ++j ){

if( !print[ j ] )continue;

printf(" %s" , cities[ possible[ j ] ] );

}

printf("\n");

}

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Walking Around Wisely

\*\*\*ID: 926

\*\*\*Juez: UVA

\*\*\*Tipo: Counting Path DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <string.h>

#define MAX 35

enum{ Q , N , S , W , E};

struct Connected{

int N , S , W ,E;

}ady[ MAX ][ MAX ];

int n , xini , yini , xfin , yfin;

unsigned long long dp[ MAX ][ MAX ];

int dx[] = { 0 , 1 };

int dy[] = { 1 , 0 };

void solve(){

int i , j , ans ,ny , nx , k;

memset( dp , 0 , sizeof( dp ) );

dp[ xini - 1 ][ yini - 1 ] = 1;

for( i = xini - 1 ; i < xfin ; ++i ){

for( j = yini - 1 ; j < yfin ; ++j ){

for( k = 0 ; k < 2 ; ++k ){

nx = dx[ k ] + i; ny = dy[ k ] + j;

if( nx >= 0 && ny >= 0 && nx < xfin && ny < yfin ){

//izq - der

if( k == 0 && ady[ i ][ j ].E == 1 || ady[ nx ][ ny ].W == 1 ) continue;

//arrib - abajo

else if( ady[ i ][ j ].S == 1 || ady[ nx ][ ny ].N == 1 ) continue;

dp[ nx ][ ny ] += dp[ i ][ j ];

}

}

}

}

printf("%llu\n" , dp[ xfin - 1 ][ yfin - 1 ] );

}

int main(){

char c;

int t ,x , y , q ;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d%d %d%d %d" , &n , &xini , &yini , &xfin , &yfin , &q );

memset( ady , 0 , sizeof( ady ) );

while( q-- ){

scanf("%d %d %c" , &x , &y , &c );

switch( c ){

case 'S':

ady[ x - 1 ][ y - 1 ].W = 1;

break;

case 'N':

ady[ x - 1 ][ y - 1 ].E = 1;

break;

case 'W':

ady[ x - 1 ][ y - 1 ].N = 1;

break;

case 'E':

ady[ x - 1 ][ y - 1 ].S = 1;

break;

}

}

solve();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Integer Sequences from Addition of Terms

\*\*\*ID: 927

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

long long pow( int a , int b ){

long long ans = 1;

for( int i = 1 ; i <= b ; ++i ) ans \*= a;

return ans;

}

#define MAX 100005

long long a[ MAX ], ini[ MAX ] , fin[ MAX ], sum , ant, c[ 55 ], d ;

int main(){

int t, l, n , k;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

for( int i = 0 ; i <= n ; ++i ) scanf("%lld" , &c[ i ] );

scanf("%d %d" , &d , &k );

l = 1;

sum = 0;

ant = 1;

memset( a , 0LL , sizeof( a ) );

for( ; ; ++l ){

for( int j = 0 ; j <= n ; ++j ){

a[ l ] += c[ j ] \* pow( l , j );

}

sum += d \* l;

ini[ l ] = ant;

fin[ l ] = sum;

ant = fin[ l ] + 1;

if( sum > 1000000LL )break;

}

for( int i = 1 ; i <= l ; ++i ){

if( k >= ini[ i ] && k <= fin[ i ] ){

printf("%lld\n" , a[ i ] );

break;

}

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Eternal Truths

\*\*\*ID: 908

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, BFS, shortest path

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

#include<queue>

using namespace std;

#define MAX 305

int h,w;

const int INF= ( 1 << 21);

struct Estado{

int x;

int y;

int d;

int dir;

Estado(int x1,int y1,int d1):x(x1),y(y1),d(d1){}

Estado(int x1,int y1,int d1,int av1):x(x1),y(y1),d(d1),dir(av1){}

};

char ady[MAX][MAX];

///ARREGLO PARA DISTANCIAS, TBN PONEMOS LA CANTIDAD DE PASOS A DAR 1 2 3

int dist[MAX][MAX][3];

int dx[4]={1,0,-1,0};

int dy[4]={0,1,0,-1};

int xfin,yfin;

int bfs(int x,int y){

Estado inicial(x,y,0,0);

queue<Estado> Q;

Q.push(inicial);

for(int i=0; i<h; i++)

for(int j = 0; j<w; j++)

for(int k = 0; k<3;k++)

dist[i][j][k] = INF;

dist[x][y][0]=0;

int len;

while(!Q.empty()){

Estado actual=Q.front();Q.pop();

len=actual.dir+1;

for(int i=0;i<4;i++){

int nx=actual.x;

int ny=actual.y;

bool b=true;

for(int j=0;j<len;j++){

nx=dx[i]+nx;

ny=dy[i]+ny;

if(nx<0 ||nx>=h || ny<0 || ny>=w || ady[nx][ny]=='#'){b=false;break;}

}

if(b){

if(dist[nx][ny][len%3]>dist[actual.x][actual.y][actual.dir]+1){

dist[nx][ny][len%3]=dist[actual.x][actual.y][actual.dir]+1;

Estado vecino(nx,ny,dist[actual.x][actual.y][actual.dir]+1,len%3);

Q.push(vecino);

}

}

}

}

int min=INF;

for(int i=0;i<3;i++){

if(dist[xfin][yfin][i]<min){

min=dist[xfin][yfin][i];

}

}

if(min==INF)return -1;

return min;

}

int main(){

int casos,xini,yini;

string s;

scanf("%d",&casos);

while(casos--){

scanf("%d %d",&h,&w);

for(int i=0;i<h;i++){

cin>>s;

for(int j=0;j<w;j++){

ady[i][j]=s[j];

if(s[j]=='S'){xini=i;yini=j;}

if(s[j]=='E'){xfin=i;yfin=j;}

}

}

int dist=bfs(xini,yini);

if(dist==-1)printf("NO\n");

else printf("%d\n",dist);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Number Maze

\*\*\*ID: 929

\*\*\*Juez: UVA

\*\*\*Tipo: Dijkstra

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#include <queue>

using namespace std;

#define MAX 1005

#define INF 1<<30

int h , w , ady[ MAX ][ MAX ] , dx[ 4 ] = { 0 , 0 , 1 , -1 } , dy[ 4 ] = { 1 , -1 , 0 , 0 };

struct Node{

int x , y, d;

Node( int xx , int yy , int dd ): x( xx ) , y( yy ) , d( dd ){}

};

struct cmp{

bool operator()( Node n1 , Node n2 ){

return n1.d > n2.d;

}

};

int getId( int x , int y ){ return x \* w + y; }

int seen[ MAX ][ MAX ], dist[ MAX \* MAX ];

int dijkstra( int xini , int yini ){

priority\_queue< Node , vector< Node > , cmp > Q;

Q.push( Node( xini , yini , ady[ xini ][ yini ] ) );

int i , u , v , nx , ny;

for( i = 0 ; i < w \* h ; ++i ) dist[ i ] = INF;

memset( seen , 0 , sizeof( seen ) );

seen[ xini ][ yini ] = 1;

dist[ 0 ] = ady[ xini ][ yini ];

while( !Q.empty() ){

Node act = Q.top(); Q.pop();

u = getId( act.x , act.y );

if( u == h \* w - 1 ) break;

for( i = 0 ; i < 4 ; ++i ){

nx = dx[ i ] + act.x;

ny = dy[ i ] + act.y;

v = getId( nx , ny );

if( nx >= 0 && ny >= 0 && nx < h && ny < w && !seen[ nx ][ ny ] && dist[ v ] > dist[ u ] + ady[ nx ][ ny ] ){

seen[ nx ][ ny ] = 1;

dist[ v ] = dist[ u ] + ady[ nx ][ ny ];

Q.push( Node( nx , ny , dist[ getId( nx , ny ) ] ) );

}

}

}

printf("%d\n" , dist[ getId( h - 1 , w - 1 ) ] );

}

int main(){

int t , i , j;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d %d" , &h , &w ) == 2 ){

for( i = 0 ; i < h ; ++i )

for( j = 0 ; j < w && scanf("%d" , &ady[ i ][ j ] ) ; ++j );

dijkstra( 0 , 0 );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Checking the N-Queens Problem

\*\*\*ID: 932

\*\*\*Juez: UVA

\*\*\*Tipo: Brute Force

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

#include <cstring>

#include <cstdio>

#include <string>

#include <vector>

#include <iostream>

using namespace std;

#define MAX 35

#define pii pair<int ,int>

#define mp make\_pair

#define x first

#define y second

int n;

char ady[ MAX ][ MAX ];

int dx[ 8 ] = { 1 , -1 , 0 , 0 , 1 , 1 , -1 , -1 };

int dy[ 8 ] = { 0 , 0 , 1 , -1 , 1 , -1 , 1 , -1};

int abs( int x ){ return ( x < 0 )? -x: x; }

vector< pii > Q;

bool valid( vector< pii > v ){

int len = v.size() , x1 , y1 , x2 , y2;

for( int i = 0 ; i < len ; ++i ){

x1 = v[ i ].x; y1 = v[ i ].y;

for( int j = i + 1 ; j < len ; ++j ){

x2 = v[ j ].x; y2 = v[ j ].y;

if( abs( x1 - x2 ) == abs( y1 - y2 ) || x1 == x2 || y1 == y2 ) return false;

}

}

return true;

}

bool solve( ){

int len = Q.size() , x1 , y1 , x2 , y2 ,nx , ny;

vector<pii> v, aux;

for( int i = 0 ; i < len ; ++i ){

x1 = Q[ i ].x; y1 = Q[ i ].y;

v.clear();

for( int j = 0 ; j < len ; ++j ) if( i != j ) v.push\_back( Q[ j ] );

int k = -1;

for( int k = 0 ; k < 8 ; ++k ){

int c = 0;

while( 1 ){

c++;

nx = dx[ k ] \* c + x1; ny = dy[ k ] \* c + y1;

if( nx >= 0 && nx < n && ny >= 0 && ny < n ){

aux = v;

aux.push\_back( pii( nx , ny ) );

if( valid( aux ) ){

ady[ x1 ][ y1 ] = '0';

ady[ nx ][ ny ] = 'X';

return true;

}

}else break;

}

}

}

return false;

}

int main(){

bool first = true;

while( scanf("%d" , &n ) == 1 ){

if( !first ) putchar('\n');

first = false;

Q.clear();

for( int i = 0 ; i < n ; ++i ){

for( int j = 0 ; j < n ; ++j ){

scanf(" %c" , &ady[ i ][ j ] );

if( ady[ i ][ j ] == 'X' ){

Q.push\_back( pii( i , j ) );

}

}

}

if( valid( Q ) ) puts("YES");

else{

puts("NO");

if( solve() ){

puts("YES");

for( int i = 0 ; i < n ; ++i )

{

for( int j = 0 ; j < n ; ++j )

printf("%c" , ady[ i ][ j ] );

printf("\n");

}

}

else puts("NO");

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Genes

\*\*\*ID: 939

\*\*\*Juez: UVA

\*\*\*Tipo: Data Structure

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

map< string , string > type;

map< string , vector< string > > p;

void solve( string u , string p1 , string p2 ){

if( type.find( p1 ) == type.end() ){

solve( p1 , p[ p1 ][ 0 ] , p[ p1 ][ 1 ] );

}

if( type.find( p2 ) == type.end() ){

solve( p2 , p[ p2 ][ 0 ] , p[ p2 ][ 1 ] );

}

string type\_p1 = type[ p1 ] , type\_p2 = type[ p2 ];

if( type\_p1 == "non-existent" && type\_p2 == "non-existent" ){

type[ u ] = type\_p1;

}

else if( type\_p1 == "dominant"){

if( type\_p2 == "recessive" || type\_p2 == "dominant" ){

type[ u ] = "dominant";

}

else type[ u ] = "recessive";

}

else if( type\_p1 == "recessive" ){

if( type\_p2 == "recessive" ){

type[ u ] = type\_p1;

}

else if( type\_p2 == "dominant" ){

type[ u ] = "dominant";

}

else type[ u ] = "non-existent";

}

else if( type\_p2 == "dominant"){

type[ u ] = "recessive";

}

else type[ u ] = "non-existent";

}

int main(){

int t;

scanf("%d" , &t );

string u , v;

while( t-- > 0 && cin>>u>>v ){

if( v == "dominant" || v == "recessive" || v == "non-existent" ){

type[ u ] = v;

}

else{

p[ v ].push\_back( u );

}

}

for( map< string , vector< string > >:: iterator it = p.begin() ; it != p.end() ; ++it ){

solve( it -> first , ( it -> second )[ 0 ] , ( it -> second )[ 1 ] );

}

for( map< string , string > :: iterator it = type.begin() ; it != type.end() ; ++it ){

printf("%s %s\n" , (it -> first ).c\_str() , ( it ->second ).c\_str() );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Permutations

\*\*\*ID: 941

\*\*\*Juez: UVA

\*\*\*Tipo: Math

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 25

char s[ MAX ];

long long n, fact[ MAX ] ,sum;

int del[ 300 ];

int main(){

int t , i , j , len;

fact[ 0 ] = 1;

for( i = 1 ; i < 21 ; ++i ) fact[ i ] = i \* fact[ i - 1 ];

scanf("%d" , &t );

while( t-- > 0 && scanf("%s %lld" , &s , &n ) ){

len = strlen( s );

sort( s , s + len );

memset( del , 0 , sizeof( del ) );

string str;

for( i = 0 ; s[ i ] ; ++i ){

sum = 0;

str = "";

for( j = 0 ; s[ j ] ; ++j ) if( !del[ s[ j ] ] ) str += s[ j ];

for( j = i ; j < s[ j ] ; ++j )

{

if( sum + fact[ len - 1 ] > n ){

n -= sum;

del[ str[ j - i ] ] = 1;

printf("%c" , str[ j - i ] );

len--;

break;

}

sum += fact[ len - 1 ];

}

}

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Happy Numbers

\*\*\*ID: 944

\*\*\*Juez: UVA

\*\*\*Tipo: Cycle Finding

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int f( int n ){

int sum = 0;

while( n > 0){

sum += ( n % 10 )\* ( n %10 );

n /= 10;

}

return sum;

}

#define MAX 100000

int dp[ MAX ];

bool happy[ MAX ];

void TortoiseHare(int x){

int s = 0, p = 1;

int i = f(x), j = f(f(x));

while (i != j) i = f(i), j = f(f(j));

j = i; i = x;

while (i != j) i = f(i), j = f(j), s++;

j = f(i);

while (i != j) j = f(j), p++;

if( p == 1 ){ happy[ x ] = true; dp[ x ] = s + 1; }

}

void Solve(){

for( int i = 1 ; i< MAX ;++i ){

happy[ i ] = false; dp[ i ] = -1;

TortoiseHare( i );

}

}

int main(){

Solve();

int n, L ,H;

bool b = false;

while( scanf("%d %d", &L, &H ) != EOF ){

if( b ) printf("\n");

b = true;

for( int i = L; i <= H; ++i ){

if( happy[ i ] ){

printf("%d %d\n", i , dp[ i ] );

}

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Popes

\*\*\*ID: 957

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <queue>

using namespace std;

int main(){

int Y , P , x;

while( scanf("%d %d" , &Y , &P ) == 2 ){

queue<int> Q;

int best[ 3 ] = { -1 , -1 , -1 };

for( int i = 0 ; i < P ; ++i ){

scanf("%d" , &x );

Q.push( x );

while( Q.size() > 0 && Q.back() - Q.front() >= Y ) Q.pop();

if( (int)Q.size() > best[ 0 ] ){

best[ 0 ] = Q.size();

best[ 1 ] = Q.front();

best[ 2 ] = Q.back();

}

}

printf("%d %d %d\n" , best[ 0 ] , best[ 1 ] , best[ 2 ] );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Taxicab Numbers

\*\*\*ID: 962

\*\*\*Juez: UVA

\*\*\*Tipo: Math

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1000100001

#define pb push\_back

vector< long long > cubes;

set< long long > s;

map< long long , long long > mp;

void gen(){

long long i , j;

for( i = 1 ; i \* i \* i < MAX; ++i ){

cubes.pb( i \* i \* i );

}

for( i = 0 ; i < cubes.size() ; ++i ){

for( j = 0 ; j < cubes.size() ; ++j ){

if( cubes[ i ] + cubes[ j ] > MAX ) break;

mp[ cubes[ i ] + cubes[ j ] ]++;

}

}

}

int main(){

gen();

vector< long long > v;

long long n , rg ,ans;

map< long long , long long >::iterator it = mp.begin();

for( ; it != mp.end() ; ++it ){

if( it -> second > 2 ){

v.pb( it -> first );

}

}

int i;

while( scanf("%lld %lld" , &n , &rg ) != EOF ){

for( i = 0 ; i < v.size() ; ++i ){

if( v[ i ] >= n ) break;

}

ans = 0;

for( ; i < v.size() ; ++i ){

if( v[ i ] > n + rg ) break;

ans++;

printf("%lld\n" , v[ i ] );

}

if( ans == 0 ) puts("None");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Circular

\*\*\*ID: 967

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory, Sieve

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstdio>

#include <cstring>

#include <algorithm>

using namespace std;

char prime[1000010];

int cnt[1000010];

int check(int n) {

if (n < 2 || n >= 1000000) return 0;

int t = 1;

while (t <= n) t \*= 10;

t /= 10;

for (int m = n;;) {

if (!prime[m]) return 0;

m = (m % t) \* 10 + (m / t);

if (m == n) break;

}

return 1;

}

int main() {

memset(prime, 1, sizeof(prime));

prime[0] = prime[1] = 0;

for (int i = 2; i <= 1000; i++) {

if (!prime[i]) continue;

for (int j = i\*i; j <= 1000000; j += i)

prime[j] = 0;

}

cnt[0] = 0;

for (int n = 1; n <= 1000000; n++)

cnt[n] = cnt[n-1] + check(n);

int a, b;

while (scanf("%d %d", &a, &b) == 2 && a >= 0) {

if (a > b) swap(a, b);

int k = cnt[b] - cnt[a-1];

if (k == 0)

printf("No Circular Primes.\n");

else if (k == 1)

printf("1 Circular Prime.\n");

else

printf("%d Circular Primes.\n", k);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Kaprekar Numbers

\*\*\*ID: 974

\*\*\*Juez: UVA

\*\*\*Tipo: Brute Force

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int a[ 20 ] = { 9, 45 , 55, 99,297,703,999,2223,2728,4879,4950,5050,5292,7272,7777,9999,17344,22222,38962};

int main(){

int t , q , L , U , i , b;

scanf("%d" , &t );

for( q = 0 ; q < t && scanf("%d %d" , &L , &U ); ++q ){

if( q ) printf("\n");

printf("case #%d\n" , q + 1 );

b = 0;

for( i = 0 ; i < 20 ; ++i ){

if( a[ i ] >= L && a[ i ] <= U ){

printf("%d\n" , a[ i ] );

b = 1;

}

}

if( !b ) puts("no kaprekar numbers");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Lemmings Battle!

\*\*\*ID: 978

\*\*\*Juez: UVA

\*\*\*Tipo: Binary search

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t , batles, green , blue , x , len;

vector<int> g , b;

vector<int>::iterator low;

scanf("%d" , &t );

for( int q = 0 ; q < t ; ++q ){

if( q )putchar('\n');

scanf("%d %d %d" , &batles ,&green , &blue );

for( int i = 0 ; i < green ; ++i ){

scanf("%d" , &x );

g.push\_back( x );

}

for( int i = 0 ; i < blue ; ++i ){

scanf("%d" , &x );

b.push\_back( x );

}

sort( g.begin() , g.end() );

sort( b.begin() , b.end() );

while( 1 ){

len = min( batles , min( (int)g.size() , (int)b.size() ) );

vector<int> gi , bi;

for( int i = 0 ,lg, lb ; i < len ; ++i ){

lg = g.size() - 1 ; lb = b.size() - 1;

if( g[ lg ] > b[ lb ] ){

gi.push\_back( g[ lg ] - b[ lb ] );

}

else if( b[ lb ] > g[ lg ]){

bi.push\_back( b[ lb ] - g[ lg ] );

}

g.pop\_back();

b.pop\_back();

}

//insertamos sobrevivientes en arreglo ordenado

for( int i = 0 ; i < gi.size() ; ++i ){

low = lower\_bound( g.begin() , g.end() , gi[ i ] );

g.insert( low , gi[ i ] );

}

for( int i = 0 ; i < bi.size() ; ++i ){

low = lower\_bound( b.begin() , b.end() , bi[ i ] );

b.insert( low , bi[ i ] );

}

if( g.size() == 0 ){

if( b.size() == 0 ){

puts("green and blue died");

break;

}

else{

puts("blue wins");

for( int i = b.size() - 1 ; i >=0 ; --i )

printf("%d\n" , b[ i ] );

break;

}

}

else if( g.size() > 0 && b.size() == 0){

puts("green wins");

for( int i = g.size() - 1 ; i >=0 ; --i )

printf("%d\n" , g[ i ] );

break;

}

}

g.clear();

b.clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Round and Round Maze

\*\*\*ID: 985

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, BFS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stack>

using namespace std;

#define MAX 505

int xfin , yfin, F, C;

bool ady[ MAX ][ MAX ][ 4 ], visited[ MAX ][ MAX ][ 4 ];

///N, E, S, O

int dx[ 4 ] = {-1 , 0 , 1 , 0 };

int dy[ 4 ] = { 0 , 1 , 0 , -1 };

struct Estado{

int x;

int y;

Estado( int xx ,int yy ): x( xx ), y( yy ){}

Estado(){}

}Q[ MAX\*MAX\*4];

int bfs(){

int start, size, x, y;

start = size = 0;

Q[ size++ ] = Estado( 0 , 0 );

memset( visited, 0 , sizeof( visited ) );

visited[ 0 ][ 0 ][ 0 ] = 1;

for( int k = 0 ; size!= 0 ; ++k ){

for( int j = size - 1; j >= 0 ; --j ){

x = Q[ start ].x;

y = Q[ start ].y;

start++;

size--;

if( x == xfin && y == yfin ){

return k;

}

for( int i = 0 ; i < 4 ; ++i ){

if( ady[ x ][ y ][ i ] == false) continue;

int nx = x + dx[ (i + k ) % 4 ];

int ny = y + dy[ (i + k ) % 4 ];

if( nx >= 0 && nx < F && ny >= 0 && ny < C && !visited[ nx ][ ny ][ k%4 ] ){

Q[start + size ] = Estado( nx, ny );

size++;

visited[ nx ][ ny ][ k%4 ] = 1;

}

}

}

}

return -1;

}

int main(){

char s[ 5 ];

int l;

while( scanf( "%d %d", &F,&C ) != EOF ){

xfin = F - 1;

yfin = C - 1;

for( int i = 0 ; i < F ; ++i ){

for( int j = 0 ; j < C ; ++j ){

///exit

if( i == xfin && j == yfin )continue;

scanf("%s", s );

l = strlen( s );

for( int q = 0 ; q < 4 ; ++q ) ady[ i ][ j ][ q ] = 0;

for( int q = 0 ; q < l ; ++q ){

if( s[ q ] == 'N' ) ady[ i ][ j ][ 0 ] = 1;

if( s[ q ] == 'E' ) ady[ i ][ j ][ 1 ] = 1;

if( s[ q ] == 'S' ) ady[ i ][ j ][ 2 ] = 1;

if( s[ q ] == 'W' ) ady[ i ][ j ][ 3 ] = 1;

}

}

}

int resp = bfs();

if( resp != -1 )printf("%d\n", resp );

else printf("no path to exit\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Many paths, one destination

\*\*\*ID: 988

\*\*\*Juez: UVA

\*\*\*Tipo: Topological Sort, Longesth Path DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstring>

#include <cstdio>

#include <vector>

#include <queue>

using namespace std;

#define MAX 105

int V, dp[ MAX ], outdegree[ MAX ] , indegree[ MAX ];

vector< int > ady[ MAX ];

void countingPath(){

int i , act , y , ans = 0;

queue< int > Q;

memset( dp , 0 , sizeof( dp ) );

for( i = 0 ; i < V ; ++i ){

if( !indegree[ i ] ){

Q.push( i );

dp[ i ] = 1;

}

}

while( !Q.empty() ){

act = Q.front(); Q.pop();

for( i = 0 ; i < ady[ act ].size() ; ++i ){

y = ady[ act ][ i ];

indegree[ y ]--;

if( !indegree[ y ] ) Q.push( y );

dp[ y ] += dp[ act ];

}

}

for( i = 0 ; i < V ; ++i ){

if( !outdegree[ i ] ) ans += dp[ i ];

}

printf("%d\n" , ans );

}

int main(){

int i , j , k , v, q = 0;

while( scanf("%d" , &V ) != EOF ){

if( q++ ) printf("\n");

memset( indegree , 0 , sizeof( indegree ) );

memset( outdegree , 0 , sizeof( outdegree ) );

for( i = 0 ; i < V && scanf("%d" , &k ) == 1 ; ++i ){

for( j = 0 ; j < k && scanf("%d" , &v ) ; ++j ){

ady[ i ].push\_back( v );

outdegree[ i ]++;

indegree[ v ]++;

}

}

countingPath();

for( i = 0 ; i < V ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Su Doku

\*\*\*ID: 989

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 10

int ady[ MAX ][ MAX ];

int dx[ 4 ] = { 1 , -1 , 0 , 0 };

int dy[ 4 ] = { 0 , 0 , 1 , -1 };

bool used[ MAX ][ MAX ];

int sq3[ 5 ] = { 2 , 5 , 8 }, sq2[ 5 ] = { 1 , 3 };

bool seen[ MAX ][ MAX ];

bool finish;

int N;

void solve( int x , int y ){

if( x == N && y == 0 ){

for( int i = 0 ; i < N ; ++i ){

printf("%d" , ady[ i ][ 0 ] );

for( int j = 1 ; j < N ; ++j ){

printf(" %d" , ady[ i ][ j ]);

}

printf("\n");

}

finish = true;

return;

}

if( finish ) return;

if( used[ x ][ y ] ){

if( y + 1 < N ){ solve( x , y + 1 ); if( finish ) return;}

else solve( x + 1 , 0 );

return;

}

if( seen[ x ][ y ] )return;

int possible[ 15 ] = { 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 };

//en direcciones

for( int i = 0 ; i < 4 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = dx[ i ] \* k + x, ny = dy[ i ] \* k + y;

if( nx >= 0 && ny >= 0 && nx < N && ny < N ){

possible[ ady[ nx ][ ny ] ] = 0;

}

else break;

}

}

//en cuadrado

int cuadrado = -1, xx = 0 , yy = 0, iniX = 0, iniY = 0;

if( N == 9 ){

for( int i = 0 ; i < 3 ; ++i ){

int j = 0;

for( ; j < 3 ; ++j ){

cuadrado++;

if( x <= sq3[ i ] && y <= sq3[ j ] ){

xx = sq3[ i ];

yy = sq3[ j ];

break;

}

}

if( j != 3 )break;

}

iniX = 0; iniY = ( cuadrado \* 3 ) % 9;

if( cuadrado <= 2 )iniX = 0;

else if( cuadrado <= 5 ) iniX = 3;

else if( cuadrado <= 8 ) iniX = 6;

}

else{

for( int i = 0 ; i < 2 ; ++i ){

int j = 0;

for( ; j < 2 ; ++j ){

cuadrado++;

if( x <= sq2[ i ] && y <= sq2[ j ] ){

xx = sq2[ i ];

yy = sq2[ j ];

break;

}

}

if( j != 2 )break;

}

iniX = 0; iniY = ( cuadrado \* 2 ) % 4;

if( cuadrado <= 1 )iniX = 0;

else if( cuadrado <= 3 ) iniX = 2;

}

for( int i = iniX ; i <= xx ; ++i ){

for( int j = iniY ; j <= yy ; ++j ){

if( ady[ i ][ j ] != 0 ) possible[ ady[ i ][ j ] ] = 0;

}

}

for( int i = 1 ; i <= N ; ++i ){

if( possible[ i ] ){

ady[ x ][ y ] = i;

seen[ x ][ y ] = 1;

if( y + 1 < N ){ solve( x , y + 1 ); if( finish ) return; }

else solve( x + 1 , 0 );

if( finish ) return;

seen[ x ][ y ] = 0;

ady[ x ][ y ] = 0;

}

}

if( finish ) return;

}

int main(){

bool first = false;

while( scanf("%d" , &N ) == 1 ){

N \*= N;

if( first ) printf("\n");

first = true;

for( int i = 0 ; i < N ; ++i )

{

for( int j = 0 ; j < N ; ++j ){

scanf("%d" , &ady[ i ][ j ] );

used[ i ][ j ] = 0;

seen[ i ][ j ] = 0;

if( ady[ i ][ j ] != 0 ) { used[ i ][ j ] = 1; seen[ i ][ j ] = 1;}

}

}

finish = false;

solve( 0 , 0 );

if( !finish )printf("NO SOLUTION\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Diving for gold

\*\*\*ID: 990

\*\*\*Juez: UVA

\*\*\*Tipo: DP, Knapsack

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 2005

int dp[ MAX ], dep[ MAX ] , v[ MAX ], seen[ 35 ][ 1005 ];

int main(){

int t , w , n, c;

bool b = false;

while( ~scanf("%d %d" , &t , &w ) ){

if( b )putchar('\n');

b = true;

memset( dp , 0 , sizeof( dp ) );

memset( seen , 0 , sizeof( seen ) );

scanf("%d" , &n );

for( int i = 0 ; i < n ; ++i )

scanf("%d %d" , &dep[ i ] , &v[ i ] );

for( int i = n - 1 ; i >= 0 ; --i ){

c = 3 \* w \* dep[ i ];

for( int j = t ; j >= c ; --j ){

if( dp[ j ] < dp[ j - c ] + v[ i ] ){

dp[ j ] = dp[ j - c ] + v[ i ];

seen[ i ][ j ] = 1;

}

}

}

printf("%d\n" , dp[ t ] );

int cnt = 0;

for( int i = 0 , j = t ; i < n ; ++i ){

if( seen[ i ][ j ] ){

cnt++;

j -= ( dep[ i ] \* 3 \* w );

}

}

printf("%d\n" , cnt );

for( int i = 0 , j = t ; i < n ; ++i ){

if( seen[ i ][ j ] ){

printf("%d %d\n" , dep[ i ] , v[ i ] );

j -= ( dep[ i ] \* 3 \* w );

}

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Safe Salutations

\*\*\*ID: 908

\*\*\*Juez: UVA

\*\*\*Tipo: DP, Catalan numbers

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#define MAX 10

int a[MAX+1];

int main(){

a[0] = 1;

int j;

for(int i = 1; i <= MAX; i++){

a[i] = 0;

for(j = 0; j < i; j++){

a[i] += a[j] \* a[i-j-1];

}

}

int n;

bool button = false;

while(scanf("%d",&n)!=EOF){

if(button)

printf("\n");

else

button = true;

printf("%d\n",a[n]);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Product of digits

\*\*\*ID: 993

\*\*\*Juez: UVA

\*\*\*Tipo: Math

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<vector>

#include<algorithm>

#include<stdio.h>

#include<math.h>

#include<string>

#include<sstream>

#include<time.h>

using namespace std;

int minProd(int n){

vector<int> factores;

int i;

for(i=9;i>=2;i--){

while(n%i==0){

factores.push\_back(i);

n=n/i;

}

if(n==1){break;}

}

//si n>9 ejemplo 22, factores tendria solo 2 y n seria 11 saldira de ciclo

if(n>9){

printf("-1\n");

return 0;

}

//recorremos arreglo e imprimos pero alreves xq pide el minimo

for(int i=factores.size()-1;i>=0;i--){

printf("%d",factores.at(i));

}

printf("\n");

return 0;

}

int main(){

int casos,n,i;

while(scanf("%d",&casos)!=EOF){

for(i=0;i<casos;i++){

scanf("%d",&n);

if(n<0 || n>1000000000){break;}

if(n==0){printf("0\n"); continue;}

if(n==1){printf("1\n"); continue;}

minProd(n);

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Celebrity jeopardy

\*\*\*ID: 1124

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 105

char s[ MAX ];

int main(){

while( gets( s ) ){

printf("%s\n" , s );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 4 values whose sum is 0

\*\*\*ID: 1152

\*\*\*Juez: UVA

\*\*\*Tipo: Meet in the middle

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int A[16000100], len;

///a + b = -( c+ d )

int main(){

int n , t;

scanf("%d" , &t );

for( int q = 0 ; q < t ; ++q ){

if( q )printf("\n");

scanf("%d" , &n );

int a[ n ], b[ n ], c[ n ],d[ n ];

for( int i = 0; i < n ; ++i )scanf("%d %d %d %d" , &a[ i ], &b[ i ], &c[ i ] , &d[ i ] );

len = 0;

for( int i = 0 ; i < n ; ++i ){

for( int j = 0 ; j < n ; ++j ){

A[ len++ ] = a[ i ] + b[ j ];

}

}

sort( A , A + len );

long long resp = 0;

pair<int\*, int\*> find;

for( int i = 0 ; i < n ; ++i ){

for( int j = 0 ; j < n ; ++j ){

find = equal\_range(A, A + len , -( c[ i ] + d[ j ] ) );

resp += find.second - find.first;

}

}

printf("%lld\n" , resp );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Air Raid

\*\*\*ID: 1184

\*\*\*Juez: UVA

\*\*\*Tipo: Bipartite Matching

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

/\*

Min Path Cover on DAG

Minimo numero de taxis requeridos para visitar todos los vertices tomando en cuenta

que cada taxi sirve a 2 vertices a la vez al coger una arista, y que el taxi puede visitar

otros vertices que continuan a la actual. Tambien cada vertice no puede tener mas de 1 taxi.

\*/

#define MAXL 125

#define MAXR 255

int seen[ MAXL ], matchL[ MAXL ] , matchR[ MAXR ];

vector< int > ady[ MAXL ];

int bpm( int left ){

if( seen[ left ] ) return 0;

seen[ left ] = 1;

int right , i;

for( i = 0 ; i < ady[ left ].size() ; ++i ){

right = ady[ left ][ i ];

if( matchR[ right ] == -1 || bpm( matchR[ right ] ) ){

matchR[ right ] = left;

matchL[ left ] = right;

return 1;

}

}

return 0;

}

int main(){

int t , E , V , u , v , cnt , i;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d %d" , &V , &E ) ){

memset( matchL , -1 , sizeof( matchL ) );

memset( matchR , -1 , sizeof( matchR ) );

while( E-- > 0 && scanf("%d %d" , &u , &v ) == 2 ){

ady[ u - 1 ].push\_back( v - 1 + V );

}

cnt = 0;

for( i = 0 ; i < V ; ++i ){

memset( seen , 0 , sizeof( seen ) );

if( bpm( i ) ) cnt++;

}

printf("%d\n" , V - cnt );

for( i = 0 ; i <= V ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Machine Schedule

\*\*\*ID: 1194

\*\*\*Juez: UVA

\*\*\*Tipo: BPM - Min Vertex Cover

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAXL 105

#define MAXR 105

int seen[ MAXL ], matchL[ MAXL ] , matchR[ MAXR ];

vector< int > ady[ MAXL ];

int bpm( int left ){

if( seen[ left ] ) return 0;

seen[ left ] = 1;

int right , i;

for( i = 0 ; i < ady[ left ].size() ; ++i ){

right = ady[ left ][ i ];

if( matchR[ right ] == -1 || bpm( matchR[ right ] ) ){

matchR[ right ] = left;

matchL[ left ] = right;

return 1;

}

}

return 0;

}

int L;

int matchings(){

memset( matchR , -1 , sizeof( matchR ) );

memset( matchL , -1 , sizeof( matchL ) );

int ans = 0 , i;

for( i = 0 ; i < L ; ++i ){

memset( seen , 0 , sizeof( seen ) );

if( bpm( i ) ) ans++;

}

return ans;

}

int main(){

int n , m ,k , A , B , i;

while( scanf("%d" , &n ) , n ){

scanf("%d %d" , &m , &L );

for( i = 0 ; i < L && scanf("%d %d %d" , &k , &A , &B); ++i ){

if( !A || !B ) continue;

ady[ A ].push\_back( B );

}

L = n;

printf( "%d\n" , matchings() );

for( i = 0 ; i <= n ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Calling Extraterrestrial Intelligence Again

\*\*\*ID: 1195

\*\*\*Juez: UVA

\*\*\*Tipo: Primes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 10000

int prime[ MAX ] , primes[ MAX ], len = 0;

void sieve(){

memset( prime , 1 , sizeof( prime ) );

prime[ 0 ] = 0;

prime[ 1 ] = 0;

for(int i = 2 ; i \* i <= MAX ; ++i ){

if( prime[ i ] ){

for(int k = 2 ; k \* i <= MAX ; ++k){

prime[ k \* i ] = 0;

}

}

}

for( int i = 2 ; i <= MAX ; ++i ){

if( prime[ i ] ) primes[ len++ ] = i;

}

}

int main(){

int m , a, b, p1 , p2, pp1 , pp2;

long long maxi;

sieve();

while( scanf("%d %d %d" , &m , &a , &b ) , m | a | b ){

maxi = 0;

for( int i = len - 1 ; i >= 0 ; --i ){

for( int j = i ; j < len ; ++j ){

p1 = primes[ i ];

p2 = primes[ j ];

if( p1 \* p2 > m || p1/( double)p2 < a/(double) b ) break;

if( p1 <= m/(double)p2 && p1/(double)p2 <= 1 && p1/(double)p2 >= a/(double)b ){

if( maxi < p1 \* p2 ){

maxi = p1 \* p2;

pp1 = p1;

pp2 = p2;

}

}

}

}

printf("%d %d\n" , pp1 , pp2 );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Tiling Up Blocks

\*\*\*ID: 1196

\*\*\*Juez: UVA

\*\*\*Tipo: DP, LIS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

#define pii pair<int,int>

#define mp make\_pair

int n;

pii a[ MAX ];

int dp[ MAX ];

bool cmp( pii p1 , pii p2 ){

if( p1.first <= p2.first && p1.second <= p2.second )

return 1;

return p1 < p2;

}

int main(){

int x , y;

while( scanf("%d" , &n ) , n ){

for( int i = 0 ; i < n && scanf("%d %d" , &x , &y) ; ++i ){

a[ i ] = mp( x , y );

dp[ i ] = 1;

}

sort( a , a + n );

for( int i = 0 ; i < n ; ++i ){

for( int j = i + 1 ; j < n ; ++j ){

if( a[i].first <= a[j].first && a[i].second <= a[j].second ){

dp[ j ] = max( dp[ j ] , dp[ i ] + 1 );

}

}

}

int maxi = 0;

for( int i = 0 ; i < n ; ++i )

maxi = max( maxi , dp[ i ] );

printf("%d\n" , maxi );

}

printf("\*\n");

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Suspects

\*\*\*ID: 1197

\*\*\*Juez: UVA

\*\*\*Tipo: Flood Fill | Union Find

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 30005

int parent[ MAX ];

int Find( int x ){

return ( x == parent[ x ] )? x: parent[ x ] = Find( parent[ x ] );

}

void Union( int x , int y ){

parent[ Find( x ) ] = Find( y );

}

int V;

void makeSet(){

for( int i = 0 ; i < V ; ++i ) parent[ i ] = i;

}

int numNodes( int x ){

int ans = 0;

for( int i = 0 ; i < V ; ++i ){

ans += ( x == Find( i ) )?1:0;

}

return ans;

}

int main(){

int g , k , ant, x ;

while( scanf("%d %d" , &V , &g ) , V | g ){

makeSet();

while( g-- ){

scanf("%d" , &k );

for( int q = 0 ; q < k ; ++q ){

scanf("%d" , &x );

if( q ) Union( ant , x );

ant = x;

}

}

printf("%d\n" , numNodes( Find( 0 ) ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Geodetic Set Problem

\*\*\*ID: 1198

\*\*\*Juez: UVA

\*\*\*Tipo: Floyd Warshall

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

#define INF 1<<30

int V, ady[ MAX ][ MAX ] , path[ MAX ][ MAX ];

char s[ 2 \* MAX ];

void init(){

for( int i = 0 ; i < V ; ++i ){

for( int j = 0 ; j < V ; ++j ){

ady[ i ][ j ] = ( i == j )?0:INF;

}

}

}

void floyd(){

for( int k = 0 ; k < V ; ++k ){

for( int i = 0 ; i < V ; ++i ){

if( ady[ i ][ k ] == INF ) continue;

for( int j = 0 ; j < V ; ++j ){

if( ady[ k ][ j ] == INF || ady[ i ][ j ] < ady[ i ][ k ] + ady[ k ][ j ] ) continue;

ady[ i ][ j ] = ady[ i ][ k ] + ady[ k ][ j ];

}

}

}

}

set< int > possible[ MAX ][ MAX ];

int a[ MAX ], seen[ MAX];

int main(){

int x , q , len , cnt;

set< int > :: iterator it;

while( scanf("%d" , &V ) != EOF ){

getchar();

init();

for( int i = 0 ; i < V && gets( s ) ; ++i ){

stringstream ss( s );

while( ss>>x ){

x--;

ady[ i ][ x ] = ady[ x ][ i ] = 1;

}

}

floyd();

for( int i = 0 ; i < V ; ++i ){

for( int j = 0 ; j < V ; ++j ){

for( int k = 0 ; k < V ; ++k ){

if( ady[ i ][ k ] + ady[ k ][ j ] <= ady[ i ][ j ] ){

possible[ i ][ j ].insert( k );

}

}

}

}

scanf("%d" , &q );

getchar();

while( q-- ){

gets( s );

stringstream ss( s );

len = 0;

while( ss>>x ){

a[ len++ ] = --x;

}

memset( seen , 0 , sizeof( seen ) );

cnt = 0;

for( int i = 0 ; i < len ; ++i ){

for( int j = i + 1 ; j < len ; ++j ){

for( it = possible[ a[ i ] ][ a[ j ] ].begin() ; it != possible[ a[ i ] ][ a[ j ] ].end() ; ++it ){

if( !seen[ \*it ] ) cnt++;

seen[ \*it ] = 1;

}

}

}

if( cnt == V ) puts("yes");

else puts("no");

}

for( int i = 0 ; i < V ; ++i )

for( int j = 0 ; j < V ; ++j )

possible[ i ][ j ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# A DP Problem

\*\*\*ID: 1200

\*\*\*Juez: UVA

\*\*\*Tipo: String Manipulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 260

char s[ MAX ];

int ten[] = { 1 , 10 , 100 , 1000 , 10000 };

int main(){

double ans;

int resp;

char ant;

int t , xl , xr, sl , sr , i , j , num , k , len;

scanf("%d" , &t );

while( t-- > 0 && scanf("%s" , &s ) ){

xl = xr = sl = sr = 0;

len = strlen( s );

s[ len++ ] = '='; s[ len ] = '\0';

for( i = 0 ; s[ i ] ; ++i ){

if( s[ i ] == 'x' ){

j = i - 1;

num = 0;

k = 0;

while( j >= 0 && s[ j ] >= '0' && s[ j ] <= '9' ) num += ten[ k++ ] \* ( s[ j-- ] - '0' );

if( num == 0 ){

if( j < 0 || s[ j ] =='+' ) xl++;

else xl--;

}else if( s[ j ] == '-') xl -= num;

else xl += num;

}

if( s[ i ] == '+' || s[ i ] == '-' || s[ i ] == '=' ){

j = i - 1;

if( s[ j ] == 'x' ){ if( s[ i ] == '=' ){ i++; break; } else continue;}

num = 0;

k = 0;

while( j >= 0 && s[ j ] >= '0' && s[ j ] <= '9' ) num += ten[ k++ ] \* ( s[ j-- ] - '0' );

if( num == 0 ) sl += num;

else{

if( s[ j ] == '-' ) sl -= num;

else sl += num;

}

if( s[ i ] == '=' ){ i++; break;}

}

}

for( ; s[ i ] ; ++i ){

if( s[ i ] == 'x' ){

j = i - 1;

num = 0;

k = 0;

while( j >= 0 && s[ j ] >= '0' && s[ j ] <= '9' ) num += ten[ k++ ] \* ( s[ j-- ] - '0' );

if( num == 0 ){

if( s[ j ] == '=' || s[ j ] =='+' ) xr++;

else xr--;

}else if( s[ j ] == '-' ) xr -= num;

else xr += num;

}

if( s[ i ] == '+' || s[ i ] == '-' || s[ i ] == '=' ){

j = i - 1;

if( s[ j ] == 'x' ){ if( s[ i ] == '=' ){ i++; break; } else continue;}

num = 0;

k = 0;

while( j >= 0 && s[ j ] >= '0' && s[ j ] <= '9' ) num += ten[ k++ ] \* ( s[ j-- ] - '0' );

if( num == 0 ) sr += num;

else{

if( s[ j ] == '-' ) sr-= num;

else sr += num;

}

}

}

if( xl != xr ){

xl -= xr;

sr -= sl;

ans = sr/(double)xl;

if( ans < 0 ) resp = (int)ans - 1;

else resp = (int) ans;

printf("%d\n" , resp );

}

else if( xl == xr && sl == sr ) puts("IDENTITY");

else puts("IMPOSSIBLE");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Taxi Cab Scheme

\*\*\*ID: 1201

\*\*\*Juez: UVA

\*\*\*Tipo: Min Path Cover

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAXL 505

#define MAXR 1010

int seen[ MAXL ], matchL[ MAXL ] , matchR[ MAXR ] , L;

vector< int > ady[ MAXL ];

int bpm( int left ){

if( seen[ left ] ) return 0;

seen[ left ] = 1;

int right , i;

for( i = 0 ; i < ady[ left ].size() ; ++i ){

right = ady[ left ][ i ];

if( matchR[ right ] == -1 || bpm( matchR[ right ] ) ){

matchR[ right ] = left;

matchL[ left ] = right;

return 1;

}

}

return 0;

}

int matchings(){

memset( matchR , -1 , sizeof( matchR ) );

memset( matchL , -1 , sizeof( matchL ) );

int ans = 0 , i;

for( i = 0 ; i < L ; ++i ){

memset( seen , 0 , sizeof( seen ) );

if( bpm( i ) ) ans++;

}

return ans;

}

struct Data{

int time , x1 , x2 , y1 , y2 , dist;

Data(){}

}d[ MAXL ];

int main(){

int t , q , n , h , m , i , j;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d" , &L ) ){

for( i = 0 ; i < L && scanf("%d:%d %d %d %d %d" , &h ,&m , &d[ i ].x1 , &d[ i ].y1 , &d[ i ].x2 , &d[ i ].y2 ); ++i ){

d[ i ].time = h \* 60 + m;

d[ i ].dist = abs( d[ i ].x2 - d[ i ].x1 ) + abs( d[ i ].y2 - d[ i ].y1 );

}

for( i = 0 ; i < L ; ++i ){

for( j = 0 ; j < L ; ++j ){

if( i != j && ( abs( d[ j ].x1 - d[ i ].x2 ) + abs( d[ j ].y1 - d[ i ].y2 ) +

d[ i ].time + d[ i ].dist ) < d[ j ].time ){

ady[ i ].push\_back( j + L );

}

}

}

printf("%d\n" , L - matchings() );

for( i = 0 ; i < L ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Argus

\*\*\*ID: 1203

\*\*\*Juez: UVA

\*\*\*Tipo: Priority\_queue

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define pii pair<int , int>

struct comp {

bool operator() (const pii &a, const pii &b) {

return a.second > b.second || ( a.second == b.second && a.first > b.first );

}

};

int main(){

char s[ 25 ];

int q , k , p, reg[ 3005 ] = { 0 } , val[ 3005 ] = { 0 };

priority\_queue< pii, vector<pii> , comp > Q;

while( scanf("%s" , &s ) , s[ 0 ] != '#' ){

scanf("%d %d" , &q , &p );

reg[ q ] = p;

val[ q ] = p;

Q.push( pii( q , reg[ q ] ) );

}

scanf("%d" , &k );

while( !Q.empty() && k-- ){

pii act = Q.top(); Q.pop();

printf("%d\n", act.first );

reg[ act.first ] += val[ act.first ];

Q.push( pii( act.first , reg[ act.first ] ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# AGTC

\*\*\*ID: 1207

\*\*\*Juez: UVA

\*\*\*Tipo: Edit Distance

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1005

char A[ MAX ] ,B[ MAX ];

int lenA , lenB , dp[ MAX ][ MAX ];

int MIN( int a , int b , int c ){ return min( a , min( b , c ) ); }

int editDistance(){

int i , j;

for( i = 0 ; i <= lenA ; ++i ) dp[ i ][ 0 ] = i;

for( j = 0 ; j <= lenB ; ++j ) dp[ 0 ][ j ] = j;

for( i = 1 ; i <= lenA ; ++i ){

for( j = 1 ; j <= lenB ; ++j ){

dp[ i ][ j ] = MIN( dp[ i - 1 ][ j ] + 1 ,

dp[ i ][ j - 1 ] + 1 ,

dp[ i - 1 ][ j - 1 ] + ( A[ i - 1 ] == B[ j - 1 ] ? 0 : 1 ) );

}

}

return dp[ lenA ][ lenB ];

}

int main(){

while( scanf("%d" , &lenA ) != EOF ){

if( lenA > 0 ) scanf("%s" , &A );

scanf("%d" , &lenB );

if( lenB > 0 ) scanf("%s" , &B );

printf("%d\n" , editDistance() );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# WordFish

\*\*\*ID: 1209

\*\*\*Juez: UVA

\*\*\*Tipo: Permutation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 25

char s[ MAX ], aux[ MAX ], ans[ MAX ];

int main(){

int n, res , mini , maxi;

while( scanf("%s" , &s ) != EOF ){

n = strlen( s );

strcpy( aux , s );

maxi = 0;

res = 0;

for( int i = 0 ; i < 10 ; ++i ){

prev\_permutation( aux , aux + n );

mini = 1<<30;

for( int j = 0 ; j < n - 1 ; ++j ){

mini = min( mini , abs( aux[ j ] - aux[ j + 1 ] ) );

}

if( mini >= maxi ){

maxi = mini;

res = i;

strcpy( ans , aux );

}

}

for( int i = 0 ; i < 11; ++i ){

mini = 1<<30;

for( int j = 0 ; j < n - 1 ; ++j ){

mini = min( mini , abs( s[ j ] - s[ j + 1 ] ) );

}

if( mini > maxi ){

maxi = mini;

strcpy( ans , s );

}

next\_permutation( s , s + n );

}

printf("%s%d\n" , ans , maxi );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Sum of Consecutive Prime Numbers

\*\*\*ID: 1210

\*\*\*Juez: UVA

\*\*\*Tipo: Sieve

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10015

int prime[ MAX ] , len , p[ MAX ];

void sieve(){

memset( prime , 1 , sizeof( prime ) );

prime[ 0 ] = 0;

prime[ 1 ] = 0;

for(int i = 2 ; i \* i <= MAX ; ++i ){

if( prime[ i ] ){

for(int k = 2 ; k \* i <= MAX ; ++k){

prime[ k \* i ] = 0;

}

}

}

len = 0;

for( int i = 0 ; i < MAX ; ++i ) if( prime[ i ] ){ p[ len++ ] = i;}

}

int dp[ MAX ];

void gen(){

int i , j;

for( i = 0 ; i < len ; ++i ){

int sum = p[ i ];

dp[ sum ]++;

for( j = i + 1 ; j < len ; ++j ){

sum += p[ j ];

if( sum > MAX ) break;

dp[ sum ]++;

}

}

}

int main(){

sieve();

gen();

int n;

while( scanf("%d" , &n ) , n ){

printf("%d\n" , dp[ n ] );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Sum of different primes

\*\*\*ID: 1213

\*\*\*Juez: UVA

\*\*\*Tipo: DP, knapsack

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1125

int prime[ MAX ] ,primes[ MAX ] , sz;

void sieve(){

memset( prime , 1 , sizeof( prime ) );

prime[ 0 ] = 0;

prime[ 1 ] = 0;

for(int i = 2 ; i \* i <= MAX ; ++i ){

if( prime[ i ] ){

for(int k = 2 ; k \* i <= MAX ; ++k){

prime[ k \* i ] = 0;

}

}

}

sz = 0;

for( int i = 2 ; i <= MAX ; ++i ){

if( prime[ i ] ) primes[ sz++ ] = i;

}

}

int n , k;

int dp[ MAX ][ 15 ];

int main(){

sieve();

memset( dp , 0 , sizeof( dp ) );

dp[ 0 ][ 0 ] = 1;

for( int i = 0 ; i < sz ; ++i ){

for( int j = MAX ; j >= primes[ i ] ; --j ){

for( int q = 1 ; q < 15 ; ++q ){

dp[ j ][ q ] += dp[ j - primes[ i ] ][ q - 1 ];

}

}

}

while( scanf("%d %d" , &n , &k ) , n | k ){

printf("%d\n" , dp[ n ][ k ]);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Bug Sensor Problem

\*\*\*ID: 1216

\*\*\*Juez: UVA

\*\*\*Tipo: MST

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 100005

#define MAXE 1000005

int uf[ MAX ] , V , E , numComp;

int Find( int x ){ return ( x == uf[ x ] )? x : uf[ x ] = Find( uf[ x ] ); }

void Union( int x , int y ){ uf[ Find( x ) ] = Find( y ); }

void MakeSet(){ for( int i = 0 ; i < V ; ++i ) uf[ i ] = i; numComp = V; }

struct Edge{

int u , v;

double w;

Edge( int uu , int vv , double ww ) : u(uu) , v(vv) , w(ww){}

bool operator < ( const Edge e1 ) const{

return w < e1.w;

}

Edge(){}

}graph[ MAXE ];

int a[ MAX ][2];

double dist( double x1 , double y1 , double x2 , double y2 ){

return sqrt( ( x1 - x2 ) \* ( x1 - x2 ) + ( y1 - y2 ) \* ( y1 - y2 ) );

}

int main(){

int t , n , x , y;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d" , &n ) ){

V = 0;

while( scanf("%d" , &x ) , x != -1 ){

scanf("%d" , &y );

a[V][0] = x; a[V++][1] = y;

}

E = 0;

for( int i = 0 ; i < V ; ++i ){

for( int j = i + 1 ; j < V ; ++j ){

graph[ E++ ] = Edge( i , j , dist( a[ i ][0 ] , a[i][1] , a[j][0] , a[j][1] ) );

}

}

MakeSet();

sort( graph , graph + E );

if( V <= n ){

printf("0\n");

continue;

}

for( int i = 0 ; i < E ; ++i ){

int u = graph[i].u;

int v = graph[i].v;

double w = graph[i].w;

if( Find( u ) != Find( v ) ){

Union( u , v );

numComp--;

}

if( numComp <= n ){

printf("%d\n" , (int)ceil(w) );

break;

}

}

}

return 0;

}

/\*

3

2

1 1

2 1

8 7

-1

3

1 1

2 1

8 7

-1

1

1 1

99999 99999

-1

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Editor

\*\*\*ID: 1223

\*\*\*Juez: UVA

\*\*\*Tipo: Suffix Array

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<cmath>

#include<cstring>

#include<stdio.h>

#define MAXN 5010

#define Clear(a) memset(a,0,sizeof(a))

#define Max(a,b) a>b?a:b

#define Min(a,b) a<b?a:b

using namespace std;

#define F(x) ((x)/3+((x)%3==1?0:tb))

#define G(x) ((x)<tb?(x)\*3+1:((x)-tb)\*3+2)

int wa[ MAXN ] , wb[ MAXN ],wv[ MAXN ], wss[ MAXN ];

int LCP[ MAXN ], pos[ MAXN \* 3 ], rank[ MAXN ], r[ MAXN \* 3 ];

char s[ MAXN ];

int c0(int r[],int a,int b) {return r[a]==r[b]&&r[a+1]==r[b+1]&&r[a+2]==r[b+2];}

int c12(int k,int r[],int a,int b) {

if( k == 2 )

return r[a]<r[b]||r[a]==r[b]&&c12(1,r,a+1,b+1);

else return r[a]<r[b]||r[a]==r[b]&&wv[a+1]<wv[b+1];

}

void sort(int r[],int a[],int b[],int n,int m)

{

int i;

for( i = 0 ; i < n ; ++i ) wv[ i ] = r[ a[ i ] ];

for( i = 0 ; i < m ; ++i ) wss[ i ] = 0;

for( i = 0 ; i < n ; ++i ) wss[wv[i]]++;

for( i = 1 ; i < m ; ++i ) wss[ i ] += wss[i-1];

for( i = n - 1 ; i >= 0 ; --i ) b[--wss[wv[i]]]=a[i];

return;

}

void dc3(int r[],int pos[],int n,int m)

{

int i , j , \*rn = r + n ,\*san=pos+n,ta=0,tb=(n+1)/3,tbc=0,p;

r[ n ] = r[ n + 1 ] = 0;

for( i = 0 ; i < n ; ++i ) if( i % 3 != 0 ) wa[ tbc++ ]=i;

sort( r + 2 , wa , wb , tbc , m );

sort( r + 1 , wb , wa , tbc , m );

sort( r , wa , wb , tbc , m );

for( p = 1 , rn[ F( wb[ 0 ] ) ] = 0 ,i=1; i<tbc; ++i)

rn[ F( wb[ i ] ) ] = c0( r , wb[i-1] , wb[i] )?p-1:p++;

if( p < tbc ) dc3( rn , san , tbc , p );

else for( i = 0 ; i < tbc ; ++i ) san[rn[i]] = i;

for( i = 0 ; i < tbc; ++i )if(san[i]<tb) wb[ta++]=san[i]\*3;

if( n % 3 == 1 ) wb[ta++] = n - 1;

sort( r , wb , wa , ta , m );

for( i = 0 ; i < tbc ; ++i ) wv[wb[i]=G(san[i])]=i;

for( i = 0 ,j=0 , p=0 ; i<ta && j < tbc ; ++p )

pos[ p ] =c12(wb[j]%3,r,wa[i],wb[j])?wa[i++]:wb[j++];

for( ; i < ta ; ++p ) pos[ p ] = wa[ i++ ];

for( ; j < tbc ; ++p ) pos[ p ] = wb[ j++ ];

return;

}

void getLCP( int r[], int pos[], int n, int rank[], int LCP[])

{

int i , j , k = 0 ;

for( i = 1 ; i <= n ;++i) rank[ pos[ i ] ] = i;

for( i = 0 ; i < n ; LCP[ rank[ i++ ] ] = k )

for( k ? k--:0, j = pos[rank[i]-1]; r[i+k]==r[j+k] ; ++k );

return;

}

// sa[i] i [1..n] (0..n-1)

// rank[i] i [0..n-1]

// height[i] i [1..n]

void get\_suffix(char s[], int rank[], int pos[], int LCP[])

{

int n = strlen( s );

for( int i = 0 ; i <= n ; ++i ) r[ i ] = s[ i ];

dc3( r , pos, n + 1 , 256);

getLCP( r, pos , n , rank , LCP );

}

char str[ MAXN];

int main(){

int t;

scanf("%d" , &t );

while( t-- > 0 && scanf("%s",str) ){

int len = strlen( str ) , i , lrs = 0;

Clear( rank );Clear( pos ); Clear( LCP );

get\_suffix( str , rank, pos , LCP );

for( i = 1 ; i <= len ; ++i ) lrs = max( lrs , LCP[ i ] );

printf("%d\n" , lrs );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Digit Counting

\*\*\*ID: 1225

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main(){

int n , t;

int a[ 10005 ][ 10 ];

for( int i = 1 ; i <= 10000 ; ++i ){

int aux = i;

for( int j = 0 ; j < 10 ; ++j ){

a[ i ][ j ] = a[ i - 1 ][ j ];

}

while( aux > 0 ){

a[ i ][ aux % 10 ]++;

aux /= 10;

}

}

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

printf("%d" , a[ n ][ 0 ] );

for( int i = 1 ; i < 10 ; ++i ){

printf(" %d" , a[ n ][ i ] );

}

putchar('\n');

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The longest constant gene

\*\*\*ID: 1227

\*\*\*Juez: UVA

\*\*\*Tipo: Suffix Array

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<cmath>

#include<cstring>

#include<stdio.h>

#define MAXN 7000009

#define Clear(a) memset(a,0,sizeof(a))

#define Max(a,b) a>b?a:b

#define Min(a,b) a<b?a:b

using namespace std;

#define F(x) ((x)/3+((x)%3==1?0:tb))

#define G(x) ((x)<tb?(x)\*3+1:((x)-tb)\*3+2)

int wa[ MAXN ] , wb[ MAXN ],wv[ MAXN ], wss[ MAXN ];

int LCP[ MAXN ], pos[ MAXN \* 3 ], rank[ MAXN ], r[ MAXN \* 3 ];

char s[ MAXN ];

int c0(int r[],int a,int b) {return r[a]==r[b]&&r[a+1]==r[b+1]&&r[a+2]==r[b+2];}

int c12(int k,int r[],int a,int b) {

if( k == 2 )

return r[a]<r[b]||r[a]==r[b]&&c12(1,r,a+1,b+1);

else return r[a]<r[b]||r[a]==r[b]&&wv[a+1]<wv[b+1];

}

void sort(int r[],int a[],int b[],int n,int m)

{

int i;

for( i = 0 ; i < n ; ++i ) wv[ i ] = r[ a[ i ] ];

for( i = 0 ; i < m ; ++i ) wss[ i ] = 0;

for( i = 0 ; i < n ; ++i ) wss[wv[i]]++;

for( i = 1 ; i < m ; ++i ) wss[ i ] += wss[i-1];

for( i = n - 1 ; i >= 0 ; --i ) b[--wss[wv[i]]]=a[i];

return;

}

void dc3(int r[],int pos[],int n,int m)

{

int i , j , \*rn = r + n ,\*san=pos+n,ta=0,tb=(n+1)/3,tbc=0,p;

r[ n ] = r[ n + 1 ] = 0;

for( i = 0 ; i < n ; ++i ) if( i % 3 != 0 ) wa[ tbc++ ]=i;

sort( r + 2 , wa , wb , tbc , m );

sort( r + 1 , wb , wa , tbc , m );

sort( r , wa , wb , tbc , m );

for( p = 1 , rn[ F( wb[ 0 ] ) ] = 0 ,i=1; i<tbc; ++i)

rn[ F( wb[ i ] ) ] = c0( r , wb[i-1] , wb[i] )?p-1:p++;

if( p < tbc ) dc3( rn , san , tbc , p );

else for( i = 0 ; i < tbc ; ++i ) san[rn[i]] = i;

for( i = 0 ; i < tbc; ++i )if(san[i]<tb) wb[ta++]=san[i]\*3;

if( n % 3 == 1 ) wb[ta++] = n - 1;

sort( r , wb , wa , ta , m );

for( i = 0 ; i < tbc ; ++i ) wv[wb[i]=G(san[i])]=i;

for( i = 0 ,j=0 , p=0 ; i<ta && j < tbc ; ++p )

pos[ p ] =c12(wb[j]%3,r,wa[i],wb[j])?wa[i++]:wb[j++];

for( ; i < ta ; ++p ) pos[ p ] = wa[ i++ ];

for( ; j < tbc ; ++p ) pos[ p ] = wb[ j++ ];

return;

}

void getLCP( int r[], int pos[], int n, int rank[], int LCP[])

{

int i , j , k = 0 ;

for( i = 1 ; i <= n ;++i) rank[ pos[ i ] ] = i;

for( i = 0 ; i < n ; LCP[ rank[ i++ ] ] = k )

for( k ? k--:0, j = pos[rank[i]-1]; r[i+k]==r[j+k] ; ++k );

return;

}

// sa[i] i [1..n] (0..n-1)

// rank[i] i [0..n-1]

// height[i] i [1..n]

void get\_suffix(char s[], int rank[], int pos[], int LCP[])

{

int n = strlen( s );

for( int i = 0 ; i <= n ; ++i ) r[ i ] = s[ i ];

dc3( r , pos, n + 1 , 256);

getLCP( r, pos , n , rank , LCP );

}

int l[ 10 ] , n;

char str[ MAXN];

char separator[ 10 ] = { ' ' , '$' , '%' , '&' , '(' , ')' , '\*' , '+' , '.'};

int getN( int pos ){

for( int i = 1 ; i <= n ; ++i ){

if( pos >= l[ i - 1 ] && pos < l[ i ] ) return i - 1;

}

}

int main(){

int t , i , len, j , k , len\_lcs , lcs , seen;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d" , &n ) ){

len = len\_lcs = 0;

for( i = 1 ; i <= n && scanf("%s" , &s ) ; ++i ){

l[ i ] = l[ i - 1 ] + strlen( s );

if( i > 1 ) str[ len++ ] = separator[ i - 1 ];

for( j = 0 ; s[ j ] ; ++j ){

str[ len++ ] = s[ j ];

}

}

str[ len ] = '\0';

Clear( rank ); Clear( pos ); Clear( LCP );

get\_suffix( str , rank, pos , LCP );

for( i = n - 1 ; i <= len ; ++i ){

seen = 0;

lcs = 1<<30;

for( j = i , k = 0 ; k < n ; ++k , --j ){

seen |= 1<<( getN( pos[ j ] ) );

if( k < n - 1 )lcs = min( lcs , LCP[ j ] );

}

if( seen == ( ( 1<< n ) - 1 ) && len\_lcs < lcs ){

len\_lcs = lcs;

}

}

printf("%d\n" , len\_lcs );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# MODEX

\*\*\*ID: 1230

\*\*\*Juez: UVA

\*\*\*Tipo: BigMod

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

#include <cstring>

#include <stdlib.h>

#include <cctype>

#include <cmath>

#include <cstdio>

#include <cstdlib>

#include <ctime>

#include <functional>

#include <numeric>

#include <utility>

#include <deque>

#include <stack>

#include <bitset>

#include <map>

#include <set>

#include <string>

#include <vector>

#include <queue>

#include <limits>

#include <fstream>

#include <list>

#include <sstream>

#include <iostream>

#include <iomanip>

using namespace std;

long long f( long long x , long long MOD ){

return x % MOD \* x % MOD;

}

long long bigmod( long long a , long long b , long long MOD ){

if( b == 1 ) return a;

if( b % 2 == 0 ){

return f( bigmod( a , b/2 , MOD ) , MOD ) % MOD;

}

return a % MOD \* bigmod( a , b - 1 , MOD ) % MOD;

}

int main(){

int t;

long long x , y, MOD;

scanf("%d" , &t );

while( t-- ){

scanf("%lld %lld %lld" , &x , &y , &MOD );

printf("%lld\n" , bigmod( x , y , MOD ) );

}

scanf("%d" , &t );

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Skyline

\*\*\*ID: 1232

\*\*\*Juez: UVA

\*\*\*Tipo: Segment Tree

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 100005

int n = MAX;

int tree[ 4 \* MAX ] , lazy[ 4 \* MAX ];

void propagation( int left , int right, int by , int node ){

if( lazy[ node ] == -1 )

return;

int mid = ( left + right ) >>1 , L = 2 \* node + 1 , R = 2 \* node + 2;

tree[ L ] = max( tree[ L ] , tree[ node ] );

tree[ R ] = max( tree[ R ] , tree[ node ] );

lazy[ L ] = lazy[ R ] = 1;

lazy[ node ] = -1;

tree[ node ] = 0;

}

int query( int from , int to , int by , int left = 0 , int right = n - 1 , int node = 0 ){

if( from <= left && right <= to && tree[ node ] <= by ){

lazy[ node ] = 1;

tree[ node ] = by;

return right - left + 1;

}

if( from <= left && right <= to && tree[ node ] > by && lazy[ node ] == 1 ){

return 0;

}

if( left == right && tree[ node ] > by ){

return 0;

}

int mid = ( left + right ) >>1 , L = 2 \* node + 1 , R = 2 \* node + 2;

int ans = 0;

propagation( left , right , by , node );

if( from <= mid )

ans += query( from , to , by , left , mid , L );

if( to > mid )

ans += query( from , to , by , mid + 1 , right , R );

tree[ node ] = max( tree[ L ] , tree[ R ] );

if( tree[ L ] == tree[ R ] && lazy[ L ] != -1 && lazy[ L ] == lazy[ R ] ){

lazy[ node ] = 1;

}else

lazy[ node ] = -1;

return ans;

}

void build( int left = 0 , int right = n - 1 , int node = 0 ){

if( left == right ){

lazy[ node ] = -1;

tree[ node ] = 0;

return;

}

int mid = ( left + right ) >>1 , L = 2 \* node + 1 , R = 2 \* node + 2;

build( left , mid , L );

build( mid + 1 , right , R );

lazy[ node ] = -1;

tree[ node ] = tree[ L ];

}

int l\_[ MAX ] , r\_[MAX] , h\_[ MAX ];

int main(){

int t , m , l , r , h , ans, maxi;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d" , &m ) ){

ans = 0;

maxi = 0;

for( int i = 0 ; i < m && scanf("%d %d %d" , &l\_[i] , &r\_[i] , &h\_[i] ) ; ++i ){

maxi = max( maxi , r\_[i] + 1 );

}

n = maxi;

build( 0 , n );

for( int i = 0 ; i < m ; ++i ){

ans += query( l\_[i] , r\_[i] - 1 , h\_[i] );

}

printf("%d\n" , ans );

}

scanf("%d" , &t);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# RACING

\*\*\*ID: 1234

\*\*\*Juez: UVA

\*\*\*Tipo: MST

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

using namespace std;

#define MAX 100005

int uf[ MAX ] , V;

int Find( int x ){ return ( x == uf[ x ] )? x : uf[ x ] = Find( uf[ x ] ); }

void Union( int x , int y ){ uf[ Find( x ) ] = Find( y ); }

void MakeSet(){ for( int i = 1 ; i <= V ; ++i ) uf[ i ] = i; }

struct Edge{

int u , v , w;

bool operator < ( const Edge e1 ) const{

return w > e1.w;

}

Edge(){}

}graph[ MAX ];

int main(){

int t , E , i , u , v, w , ans;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d %d" , &V , &E ) ){

MakeSet();

for( i = 0 ; i < E && scanf("%d %d %d" , &graph[ i ].u , &graph[ i ].v , &graph[ i ].w ); ++i );

sort( graph , graph + E );

ans = 0;

for( i = 0 ; i < E ; ++i ){

u = graph[ i ].u;

v = graph[ i ].v;

w = graph[ i ].w;

if( Find( u ) != Find( v ) ) Union( u , v );

else ans += w;

}

printf("%d\n" , ans );

}

scanf("%d" , &t );

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Anti Brute Force Lock

\*\*\*ID: 1235

\*\*\*Juez: UVA

\*\*\*Tipo: Kruskal

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 505

int uf[ MAX ];

int V;

void MakeSet(){ int i; for( i = 0 ; i <= V ; ++i ) uf[ i ] = i; }

int Find( int x ){ return ( x == uf[ x ] )? x : uf[ x ] = Find( uf[ x ] );}

void Union( int x , int y ){ uf[ Find( x ) ] = Find( y ) ;}

char s[ MAX ][ 5 ];

struct Edge{

int u , v , w;

Edge( int uu , int vv , int ww ): u( uu ) , v( vv ) , w( ww ){}

bool operator <( const Edge e1 ) const{

return w < e1.w;

}

Edge(){}

}ady[ 300000 ];

int min( int a , int b ){ return ( a < b )? a : b; }

int getWeight( int a , int b ){

int i , cnt = 0 , w1 ,w2;

for( i = 0 ; i < 4 ; ++i ){

w1 = w2 = 0;

if( s[ a ][ i ] - '0' < s[ b ][ i ] - '0' ){

w1 = s[ b ][ i ] - s[ a ][ i ];

w2 = s[ a ][ i ] + 10 - s[ b ][ i ];

}

else if( s[ a ][ i ] - '0' > s[ b ][ i ] - '0' ){

w1 = s[ a ][ i ] - s[ b ][ i ];

w2 = s[ b ][ i ] + 10 - s[ a ][ i ];

}

cnt += min( w1 , w2 );

}

return cnt;

}

int main(){

int t , i , j , len , u , v , w , ans;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &V );

strcpy( s[ 0 ] , "0000");

for( i = 1 ; i <= V && scanf("%s" , &s[ i ] ) == 1 ; ++i );

len = 0;

MakeSet();

ans = 1<<30;

for( i = 1 ; i <= V ; ++i ){

w = getWeight( 0 , i );

if( ans > w ){

ans = w;

u = i;

}

for( j = i + 1 ; j <= V ; ++j ){

ady[ len++ ]= Edge( i , j , getWeight( i , j ) );

}

}

Union( 0 , u );

sort( ady , ady + len );

for( i = 0 ; i < len ; ++i ){

u = ady[ i ].u;

v = ady[ i ].v;

w = ady[ i ].w;

if( Find( u ) != Find( v ) ){

ans += w;

Union( u , v );

}

}

printf("%d\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Expert Enough?

\*\*\*ID: 1237

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

struct Node{

string name;

int L , R;

Node(){}

}v[ MAX ];

int main(){

int a[ MAX ];

int t , n , Q , x, idx, cnt;

scanf("%d" , &t );

char name[ 25 ];

string ans;

for( int q = 0 ; q < t ; ++q ){

if( q )putchar('\n');

scanf("%d" , &n );

for( int i = 0 ; i < n ; ++i ){

scanf("%s %d %d" , &name , &v[ i ].L , &v[ i ].R );

v[ i ].name = string( name );

}

scanf("%d" , &Q );

while( Q-- ){

scanf("%d" , &x );

cnt = 0;

for( int i = 0 ; i < n ; ++i ){

if( v[ i ].L <= x && x <= v[ i ].R ){ cnt++; ans = v[ i ].name;}

if( cnt > 1 )break;

}

if( cnt == 1 ) printf("%s\n" , ans.c\_str() );

else puts("UNDETERMINED");

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Jollybee Tournament

\*\*\*ID: 1241

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, recursion, simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

int a[ 1<<11 ];

int main(){

int n , x, m ,t;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &n , &m );

memset( a , 0 , sizeof( a ) );

while( m-- ){

scanf("%d" , &x );

a[ x ] = 1;

}

int ans = 0 , len = 1;

n = 1<<n;

while( n > 1 ){

len = 0;

for( int i = 1 ; i <= n ; i += 2 ){

if( a[ i ] && a[ i + 1 ] ){

a[ ++len ] = 1;

}

else if( !a[ i ] && !a[ i + 1 ]){

a[ ++len ] = 0;

}

else{

a[ ++len ] = 0;

ans++;

}

}

n = len;

}

printf("%d\n" , ans);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Find Terrorists

\*\*\*ID: 1246

\*\*\*Juez: UVA

\*\*\*Tipo: Prime Factorization

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

int prime[ MAX ];

void sieve(){

memset( prime , 1 , sizeof( prime ) );

prime[ 0 ] = 0;

prime[ 1 ] = 0;

for(int i = 2 ; i \* i <= MAX ; ++i ){

if( prime[ i ] ){

for(int k = 2 ; k \* i <= MAX ; ++k){

prime[ k \* i ] = 0;

}

}

}

}

bool getFactors( int n ){

map< int , int > f;

for( int i = 2 ; i \* i <= n ; ++i ){

while( n % i == 0 ){

n /= i;

f[ i ]++;

}

}

if( n > 1 ) f[ n ]++;

map< int , int > :: iterator it;

int ans = 1;

for( it = f.begin() ; it != f.end() ; ++it ) ans \*= ( it -> second + 1 );

return prime[ ans ];

}

bool correct[ MAX ];

int main(){

int t , i , a , b;

scanf("%d" , &t );

sieve();

for( i = 2 ; i < MAX ; ++i ){

correct[ i ] = getFactors( i );

}

bool possible;

while( t-- > 0 && scanf("%d %d" , &a , &b ) ){

possible = false;

for( i = a ; i <= b ; ++i ){

if( correct[ i ] ){

if( possible ) printf(" ");

printf("%d" , i );

possible = true;

}

}

if( !possible ) printf("-1");

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Interstar Transport

\*\*\*ID: 1247

\*\*\*Juez: UVA

\*\*\*Tipo: Floyd Warshall

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 27

#define INF 10000

int V , ady[ MAX ][ MAX ] , path[ MAX ][ MAX ], intermediate[ MAX ][ MAX ];

void init(){

for( int i = 0 ; i < V ; ++i ){

for( int j = 0 ; j < V ; ++j ){

ady[ i ][ j ] = ( i == j )?0:INF;

}

}

}

void init2(){

for( int i = 0 ; i < V ; ++i ){

for( int j = 0 ; j < V ; ++j ){

if( ady[ i ][ j ] == INF || ady[ i ][ j ] == 0 ){ path[ i ][ j ] = 0; intermediate[ i ][ j ] = INF; }

else path[ i ][ j ] = i, intermediate[ i ][ j ] = 0;

intermediate[ i ][ j ] = 0;

}

}

}

void floyd(){

for( int k = 0 ; k < V ; ++k ){

for( int i = 0 ; i < V ; ++i ){

if( ady[ i ][ k ] == INF ) continue;

for( int j = 0 ; j < V ; ++j ){

if( ady[ k ][ j ] != INF){

if( ady[ i ][ k ] + ady[ k ][ j ] < ady[ i ][ j ] ){

ady[ i ][ j ] = ady[ i ][ k ] + ady[ k ][ j ];

path[ i ][ j ] = path[ k ][ j ];

intermediate[ i ][ j ] = intermediate[ i ][ k ] + intermediate[ k ][ j ] + 1;

}

else if( ady[ i ][ k ] + ady[ k ][ j ] == ady[ i ][ j ] ){

if( intermediate[ i ][ j ] > intermediate[ i ][ k ] + intermediate[ k ][ j ] + 1 ){

intermediate[ i ][ j ] = intermediate[ i ][ k ] + intermediate[ k ][ j ] + 1;

path[ i ][ j ] = path[ k ][ j ];

}

}

}

}

}

}

}

void print( int x , int y ){

if( x == INF || y == INF ) return;

if( x == y ) printf("%c" , x + 'A' );

else{

print( x , path[ x ][ y ] );

printf(" %c" , y + 'A');

}

return;

}

int main(){

char u , v;

int E , w , Q;

scanf("%d %d" , &V , &E );

init();

for( int i = 0 ; i < E && scanf(" %c %c %d" , &u , &v , &w ) ; ++i ){

ady[ u - 'A' ][ v - 'A' ] = ady[ v - 'A' ][ u - 'A' ] = w;

}

init2();

floyd();

scanf("%d" , &Q );

while( Q-- ){

scanf(" %c %c" , &u , &v );

print( u - 'A' , v - 'A' );

printf("\n");

}

return 0;

}

/\*

5 6

A B 1

B C 1

C D 1

A D 6

E A 1

E D 2

1

A D

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Sales

\*\*\*ID: 1260

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 10005

int main(){

int t , n , a[ MAX ], ans;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

ans = 0;

for( int i = 0 ; i < n ; ++i ) {

scanf("%d" , &a[ i ] );

for( int j = 0 ; j < i ; ++j ){

if( a[ j ] <= a[ i ] ) ans++;

}

}

printf("%d\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Longest Path

\*\*\*ID: 10000

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Floyd Warshall, DAG

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <cstring>

#include <cstdio>

using namespace std;

#define MAX 105

#define INF 1<<30

int ady[ MAX ][ MAX ];

int V;

void Init(){

for( int i = 0 ; i <= V ; ++i ){

for( int j = 0 ; j <= V ; ++j ){

ady[ i ][ j ] = -1;

}

}

}

void Floyd(){

for( int k = 1 ; k <= V; ++k ){

for( int i = 1 ; i <= V; ++i ){

if( ady[ i ][ k ] < 0)continue;

for( int j = 1; j <= V; ++j ){

if( ady[ k ][ j ] < 0 || ady[ i ][ k ] + ady[ k ][ j ] <= ady[ i ][ j ]) continue;

ady[ i ][ j ] = ady[ i ][ k ] + ady[ k ][ j ] ;

}

}

}

}

int main(){

int inicial, p , q, k, max ,t = 1;

while ( scanf("%d" ,&V) && V ){

scanf("%d", &inicial );

Init();

while( scanf("%d %d" ,&p ,&q) && (p|q) ){

ady[ p ][ q ] = 1;

}

Floyd();

max = ady[ inicial ][ 1 ];

k = 1;

for( int i = 2; i<= V ; ++i ){

if( inicial != i ){

if( ady[ inicial ][ i ] > max ){

max = ady[ inicial ][ i ];

k = i;

}

}

}

printf("Case %d: The longest path from %d has length %d, finishing at %d.\n\n", t++ , inicial , ady[ inicial ][ k ] , k);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Bicoloring

\*\*\*ID: 10004

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, DFS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

using namespace std;

#define MAX 205

int ady[MAX][MAX];

bool visitado[MAX];

char color[MAX];

bool bicolor;

void Init(int n){

for(int i=0;i<n;i++){

visitado[i]=false;

color[i]=' ';

for(int j=0;j<n;j++){

ady[i][j]=0;

}

}

}

void solve(int inicial,int tamano){

visitado[inicial]=true;

for(int i=0;i<tamano;i++){

if(ady[inicial][i]==1 && !visitado[i]){

if(color[i]==' '){

if(color[inicial]=='R')color[i]='B';

else color[i]='R';

}

solve(i,tamano);

}

}

visitado[inicial]=false;

}

void dfs(int inicial,int tamano){

visitado[inicial]=true;

color[inicial]='R';

for(int i=0;i<tamano;i++){

if(ady[inicial][i]==1 && !visitado[i]){

if(color[i]==' '){

color[i]='B';

}

solve(i,tamano);

}

}

}

int main(){

int vertex,no,x,y,xini,yini;

while(scanf("%d",&vertex) && vertex!=0){

scanf("%d",&no);

bicolor=true;

Init(vertex);

for(int i=0;i<no;i++){

scanf("%d %d",&x,&y);

if(!i){xini=x;yini=y;}

ady[x][y]=1;

ady[y][x]=1;

}

dfs(xini,vertex);

for(int i=0;i<vertex;i++){

for(int j=0;j<vertex;j++){

if(ady[i][j]==1){

if(color[i]==color[j]){bicolor=false;break;}

}

}

if(!bicolor)break;

}

if(bicolor)printf("BICOLORABLE.\n");

else printf("NOT BICOLORABLE.\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Packing polygons

\*\*\*ID: 10005

\*\*\*Juez: UVA

\*\*\*Tipo: Circle por medio de 2 puntos y radio

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define EPS 1e-8

#define MAX 105

struct Point{

double x , y;

Point( double xx , double yy ): x( xx ) , y( yy ){}

Point(){}

double operator \*( const Point &p1 )const{

return x \* p1.x + y \* p1.y;

}

double operator ^( const Point &p1 )const{

return x \* p1.y - y \* p1.x;

}

Point operator -( const Point &p1 )const{

return Point( x - p1.x , y - p1.y );

}

bool operator <( Point p2 ) const{

return x < p2.x || ( x == p2.x && y < p2.y );

}

};

double dist( Point p1 , Point p2 ){

return sqrt( ( p1.x - p2.x ) \*( p1.x - p2.x ) + ( p1.y - p2.y ) \* ( p1.y - p2.y ) );

}

bool circle2ptsRad( Point p1 , Point p2 , double r, Point \*ctr )

{

double d2 = ( p1.x - p2.x ) \* ( p1.x - p2.x ) + ( p1.y - p2.y ) \* ( p1.y - p2.y );

double det = r \* r / d2 - 0.25;

if( det < 0.0 ) return false;

double h = sqrt( det );

ctr -> x = ( p1.x + p2.x ) \* 0.5 + ( p1.y - p2.y ) \* h;

ctr -> y = ( p1.y + p2.y ) \* 0.5 + ( p2.x - p1.x ) \* h;

return true;

}

/\*

Punto dentro de circunferencia: Retorna true si p esta dentro de circunferencia con center

y radio r

\*/

bool insideCircle( Point center , double r , Point p ){

return ( dist( center , p ) <= r + EPS );

}

Point v[ MAX ];

int main(){

int n , i , j , k;

double radio;

Point center;

bool b;

while( scanf("%d" , &n ) , n ){

for( i = 0 ; i < n && scanf("%lf %lf" , &v[ i ].x , &v[ i ].y ) ; ++i );

scanf("%lf" , &radio );

b = false;

for( i = 0 ; i < n ; ++i ){

for( j = 0 ; j < n ; ++j ){

if( i == j ) continue;

if( circle2ptsRad( v[ i ] , v[ j ] , radio , &center ) ){

for( k = 0 ; k < n ; ++k ){

if( !insideCircle( center , radio , v[ k ] ) ) break;

}

if( k == n ){

b = true;

goto end;

}

}

}

}

end:

if( !b ) puts("There is no way of packing that polygon.");

else puts("The polygon can be packed in the circle.");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Carmichael Numbers

\*\*\*ID: 10006

\*\*\*Juez: UVA

\*\*\*Tipo: Number theory, pseudoprimes, math

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#define MAX 255

int prime[MAX];

void sieve(){

for(int i = 2; i <= MAX; ++i){

prime[i]=1;

}

for(int i=2;i\*i<=MAX;i++){

if(prime[i]==1){

for(int k=2;k\*i<=MAX;k++){

prime[k\*i]=0;

}

}

}

}

int main(){

int n,i,factores[MAX],idx,aux;

sieve();

while(scanf("%d",&n)&& n!=0){

idx=0;

aux=n;

for(i=2;i\*i<=n;i++){

while(n%i==0 && prime[i]==1){

factores[idx++]=i;

n/=i;

}

}

if(n>1) {factores[idx++]=n;}

///KORSELT: numero/factor[i] - 1 % factor[i]-1 ==0

int cont=0;

for(i=0;i<idx;i++){

if(i>0 && factores[i]==factores[i-1])break;

if(((aux/factores[i])-1)%(factores[i]-1)==0)cont++;

}

if(cont==idx && cont>=3){

printf("The number %d is a Carmichael number.\n",aux);

}

else printf("%d is normal.\n",aux);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Count the Trees

\*\*\*ID: 10007

\*\*\*Juez: UVA

\*\*\*Tipo: Bignum, Catalan Numbers, binary trees

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Catalan Number\* n! -> number of trees + its permutations

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

const int MAXD = 800, DIG = 9, BASE = 1000000000;

const unsigned long long BOUND = numeric\_limits <unsigned long long> :: max () - (unsigned long long) BASE \* BASE;

struct bignum

{

int D, digits [MAXD / DIG + 2];

inline void trim ()

{

while (D > 1 && digits [D - 1] == 0)

D--;

}

inline void init (long long x)

{

memset (digits, 0, sizeof (digits));

D = 0;

do

{

digits [D++] = x % BASE;

x /= BASE;

}

while (x > 0);

}

inline bignum (long long x)

{

init (x);

}

inline bignum (int x = 0)

{

init (x);

}

inline bignum (char \*s)

{

memset (digits, 0, sizeof (digits));

int len = strlen (s), first = (len + DIG - 1) % DIG + 1;

D = (len + DIG - 1) / DIG;

for (int i = 0; i < first; i++)

digits [D - 1] = digits [D - 1] \* 10 + s [i] - '0';

for (int i = first, d = D - 2; i < len; i += DIG, d--)

for (int j = i; j < i + DIG; j++)

digits [d] = digits [d] \* 10 + s [j] - '0';

trim ();

}

inline char \*str ()

{

trim ();

char \*buf = new char [DIG \* D + 1];

int pos = 0, d = digits [D - 1];

do

{

buf [pos++] = d % 10 + '0';

d /= 10;

}

while (d > 0);

reverse (buf, buf + pos);

for (int i = D - 2; i >= 0; i--, pos += DIG)

for (int j = DIG - 1, t = digits [i]; j >= 0; j--)

{

buf [pos + j] = t % 10 + '0';

t /= 10;

}

buf [pos] = '\0';

return buf;

}

inline bool operator < (const bignum &o) const

{

if (D != o.D)

return D < o.D;

for (int i = D - 1; i >= 0; i--)

if (digits [i] != o.digits [i])

return digits [i] < o.digits [i];

return false;

}

inline bool operator == (const bignum &o) const

{

if (D != o.D)

return false;

for (int i = 0; i < D; i++)

if (digits [i] != o.digits [i])

return false;

return true;

}

inline bignum operator << (int p) const

{

bignum temp;

temp.D = D + p;

for (int i = 0; i < D; i++)

temp.digits [i + p] = digits [i];

for (int i = 0; i < p; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum operator >> (int p) const

{

bignum temp;

temp.D = D - p;

for (int i = 0; i < D - p; i++)

temp.digits [i] = digits [i + p];

for (int i = D - p; i < D; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum range (int a, int b) const

{

bignum temp = 0;

temp.D = b - a;

for (int i = 0; i < temp.D; i++)

temp.digits [i] = digits [i + a];

return temp;

}

inline bignum operator + (const bignum &o) const

{

bignum sum = o;

int carry = 0;

for (sum.D = 0; sum.D < D || carry > 0; sum.D++)

{

sum.digits [sum.D] += (sum.D < D ? digits [sum.D] : 0) + carry;

if (sum.digits [sum.D] >= BASE)

{

sum.digits [sum.D] -= BASE;

carry = 1;

}

else

carry = 0;

}

sum.D = max (sum.D, o.D);

sum.trim ();

return sum;

}

inline bignum operator - (const bignum &o) const

{

bignum diff = \*this;

for (int i = 0, carry = 0; i < o.D || carry > 0; i++)

{

diff.digits [i] -= (i < o.D ? o.digits [i] : 0) + carry;

if (diff.digits [i] < 0)

{

diff.digits [i] += BASE;

carry = 1;

}

else

carry = 0;

}

diff.trim ();

return diff;

}

inline bignum operator \* (const bignum &o) const

{

bignum prod = 0;

unsigned long long sum = 0, carry = 0;

for (prod.D = 0; prod.D < D + o.D - 1 || carry > 0; prod.D++)

{

sum = carry % BASE;

carry /= BASE;

for (int j = max (prod.D - o.D + 1, 0); j <= min (D - 1, prod.D); j++)

{

sum += (unsigned long long) digits [j] \* o.digits [prod.D - j];

if (sum >= BOUND)

{

carry += sum / BASE;

sum %= BASE;

}

}

carry += sum / BASE;

prod.digits [prod.D] = sum % BASE;

}

prod.trim ();

return prod;

}

inline double double\_div (const bignum &o) const

{

double val = 0, oval = 0;

int num = 0, onum = 0;

for (int i = D - 1; i >= max (D - 3, 0); i--, num++)

val = val \* BASE + digits [i];

for (int i = o.D - 1; i >= max (o.D - 3, 0); i--, onum++)

oval = oval \* BASE + o.digits [i];

return val / oval \* (D - num > o.D - onum ? BASE : 1);

}

inline pair <bignum, bignum> divmod (const bignum &o) const

{

bignum quot = 0, rem = \*this, temp;

for (int i = D - o.D; i >= 0; i--)

{

temp = rem.range (i, rem.D);

int div = (int) temp.double\_div (o);

bignum mult = o \* div;

while (div > 0 && temp < mult)

{

mult = mult - o;

div--;

}

while (div + 1 < BASE && !(temp < mult + o))

{

mult = mult + o;

div++;

}

rem = rem - (o \* div << i);

if (div > 0)

{

quot.digits [i] = div;

quot.D = max (quot.D, i + 1);

}

}

quot.trim ();

rem.trim ();

return make\_pair (quot, rem);

}

inline bignum operator / (const bignum &o) const

{

return divmod (o).first;

}

inline bignum operator % (const bignum &o) const

{

return divmod (o).second;

}

inline bignum power (int exp) const

{

bignum p = 1, temp = \*this;

while (exp > 0)

{

if (exp & 1) p = p \* temp;

if (exp > 1) temp = temp \* temp;

exp >>= 1;

}

return p;

}

};

bignum dp[300];

void Catalan(){

dp[0] = 1;

int j;

for(int i = 1; i <= 300; i++){

dp[i]=(bignum(i)\*4-2)\*dp[i-1]/(bignum(i)+1);

}

}

bignum dpF[300];

void fact(){

int i;

dpF[0]=1;

for(i=1;i<=300;i++){

dpF[i]=dpF[i-1]\*i;

}

}

int main(){

int n;

Catalan();

fact();

while(scanf("%d",&n) && n!=0){

printf("%s\n",(dp[n]\*dpF[n]).str());

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# What's Cryptanalysis?

\*\*\*ID: 10008

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, sorting

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<iostream>

#include<stdio.h>

#include<vector>

using namespace std;

bool f(pair<int,char> p1, pair<int,char> p2){

if(p1.first<p2.first)return true;

if(p1.first==p2.first)return p1.second>p2.second;

return false;

}

int main(){

int casos;

scanf("%d",&casos);

string s;

cin.get();

int a[26]={0};

vector<pair<int,char> > v;

while(casos--){

getline(cin,s);

for(int i=0;i<s.length();i++){

if(isalpha(s[i])){a[tolower(s[i])-'a']++;}

}

}

for(int i=0;i<26;i++){

if(a[i]!=0)v.push\_back(make\_pair(a[i],(char)(i+'a')));

}

sort(v.begin(),v.end(),f);

for(int i=v.size()-1;i>=0;i--){

cout<<(char)toupper(v[i].second)<<" "<<v[i].first<<endl;

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# All Roads Lead Where

\*\*\*ID: 10009

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, BFS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 27

vector<int> ady[ MAX ];

int prev[ MAX ] , dist[ MAX ];

void bfs( int x, int fin ){

queue< int > Q;

memset( dist , -1 , sizeof( dist ) );

memset( prev , -1 , sizeof( prev ) );

Q.push( x );

dist[ x ] = 0;

while( !Q.empty() ){

int u = Q.front(); Q.pop();

if( u == fin )return;

for( int i = 0 ; i < ady[ u ].size(); ++i ){

int v = ady[ u ][ i ];

if( dist[ v ] == -1 ){

dist[ v ] = dist[ u ] + 1;

prev[ v ] = u;

Q.push( v );

}

}

}

}

int main(){

int t, E, Q;

scanf("%d" ,&t );

char s1[ 105 ], s2[ 105 ];

for( int q = 0 ; q < t ; ++q ){

if( q )putchar('\n');

scanf("%d %d", &E, &Q );

while( E-- ){

scanf("%s %s", &s1 ,&s2);

ady[ s1[ 0 ] -'A' ].push\_back( s2[ 0 ]-'A' );

ady[ s2[ 0 ] - 'A' ].push\_back( s1[ 0 ] - 'A' );

}

while( Q-- ){

scanf("%s %s", &s1 , &s2 );

bfs( s1[ 0 ] - 'A' , s2[ 0 ] -'A');

string resp = "";

int act = s2[ 0 ]-'A';

while( 1 ){

resp += (char)( act + 'A');

if( act == s1[ 0 ]-'A' )break;

act = prev[ act ];

}

reverse( resp.begin() , resp.end() );

printf("%s\n", resp.c\_str() );

}

for( int i = 0 ; i < MAX ; ++i )ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Where's Waldorf?

\*\*\*ID: 10010

\*\*\*Juez: UVA

\*\*\*Tipo: Brute Force

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 55

char ady[ MAX ][ MAX ] , s[ MAX ];

int h , w , len;

int dx[ 8 ] = { 1 , -1, 0 , 0 , 1 , 1 , -1 , -1 };

int dy[ 8 ] = { 0 , 0 , 1 , -1 , 1 , -1 , 1 , -1};

bool solve( int x , int y){

int i , k , nx , ny;

for( i = 0 ; i < 8 ; ++i ){

k = 0;

while( true ){

nx = dx[ i ] \* k + x; ny = dy[ i ] \* k + y;

if( nx >= 0 && ny >= 0 && nx < h && ny < w && ady[ nx ][ ny ] == s[ k ] ){

k++;

}

else break;

if( k == len ){

return true;

}

}

}

return false;

}

int main(){

int t , i , j , k;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d %d" , &h , &w ) ){

for( i = 0 ; i < h && scanf("%s" , &ady[ i ]); ++i )

for( j = 0 ; j < w ; ++j )

ady[ i ][ j] = tolower( ady[ i ][ j ] );

scanf("%d" , &k );

while( k-- > 0 && scanf("%s" , &s ) ){

for( i = 0 ; s[ i ] ; ++i ) s[ i ] = tolower( s[ i ] );

len = strlen( s );

for( i = 0 ; i < h ; ++i ){

for( j = 0 ; j < w ; ++j ){

if( solve( i , j ) ){ goto end;}

}

}

end:

printf("%d %d\n" , i + 1 , j + 1 );

}

if( t ) printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Joseph's Cousin

\*\*\*ID: UVA

\*\*\*Juez: 10015

\*\*\*Tipo: joseph problem, simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <math.h>

#define MAX 32768

int primes[MAX];

void Sieve(){

int idx=0;

primes[idx++] = 2;

for (int i = 3; i < MAX; i += 2) {

bool isprime = true;

int cap = (int)sqrt(i)+1;

for (int j = 0; j < idx; j++) {

if (primes[j] >= cap) break;

if (i % primes[j] == 0) {

isprime = false;

break;

}

}

if (isprime) primes[idx++] = i;

}

}

int survivor(int n){

int i, s;

for (s = 0, i = 1; i <= n; i++) {s = (s + primes[ n- i]) % i; }

return (s + 1);

}

int main(){

Sieve();

int n;

while ( scanf("%d",&n) && n ){

printf("%d\n",survivor(n));

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Flip-Flop the Squarelotron

\*\*\*ID: 10016

\*\*\*Juez: UVA

\*\*\*Tipo: Array Manipulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 110

int a[MAX ][ MAX ] , ring , n, tmp;

void upsideDown( int ring , int len ){

int l = len - ring;

for( int i = ring ; i < len ; ++i ){

tmp = a[ ring ][ i ];

a[ ring ][ i ] = a[ len - 1 ][ i ];

a[ len - 1 ][ i ] = tmp;

}

l -= 2;

//extremos de fila

for( int i = ring + 1, j = 1; l > 0 ; ++i , j++ , l-= 2 ){

tmp = a[ i ][ ring ];

a[ i ][ ring ] = a[ len - j - 1 ][ ring ];

a[ len - j - 1 ][ ring ] = tmp;

tmp = a[ i ][ len - 1 ];

a[ i ][ len - 1 ] = a[ len - j - 1 ][ len - 1 ];

a[ len - j - 1 ][ len - 1 ] = tmp;

}

}

void leftRight( int ring , int len ){

int l = len - ring;

for( int i = ring ; i < len ; ++i ){

tmp = a[ i ][ ring ];

a[ i ][ ring ] = a[ i ][ len - 1 ];

a[ i ][ len - 1 ] = tmp;

}

l -= 2;

//extremos de fila

for( int i = ring + 1, j = 1; l > 0 ; ++i , j++ , l-=2 ){

tmp = a[ ring ][ i ];

a[ ring ][ i ] = a[ ring ][ len - j - 1 ];

a[ ring ][ len - j - 1 ] = tmp;

tmp = a[ len - 1 ][ i];

a[ len - 1 ][ i ] = a[ len - 1 ][ len - j - 1 ];

a[ len - 1 ][ len - j - 1 ] = tmp;

}

}

void mainDiagonal( int ring , int len ){

for( int i = ring ; i < len - 1 ; ++i ){

tmp = a[ ring ][ i ];

a[ ring ][ i ] = a[ i ][ ring ];

a[ i ][ ring ] = tmp;

tmp = a[ len - 1 ][ i ];

a[ len - 1 ][ i ] = a[ i ][ len - 1 ];

a[ i ][ len - 1 ] = tmp;

}

}

void mainInverse( int ring , int len ){

for( int i = len - 1 , j = ring ; i > ring ; --i , j++ ){

tmp = a[ ring ][ i ];

a[ ring ][ i ] = a[ j ][ len - 1 ];

a[ j ][ len - 1 ] = tmp;

tmp = a[ len - 1 ][ i ];

a[ len - 1 ][ i ] = a[ j ][ ring ];

a[ j ][ ring ] = tmp;

}

}

void print(){

bool f;

for( int i = 0 ; i < n ; ++i ){

f = false;

for( int j = 0 ; j < n ; ++j ){

if( f ) putchar(' ');

printf("%d", a[ i ][ j ]);

f = true;

}

printf("\n");

}

}

int main(){

int t , k ,op;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

for( int i = 0 ;i < n ; ++i ){

for( int j = 0 ; j < n ; ++j ){

scanf("%d" , &a[ i ][ j ] );

}

}

ring= n/2;

if( n & 1 ) ring++ ;

for( int i = 0 ; i < ring ; ++i ){

scanf("%d" , &k );

while( k-- ){

scanf( "%d" , &op );

if( op == 1 )upsideDown( i , n - i );

else if( op == 2 ) leftRight( i , n - i );

else if( op == 3 ) mainDiagonal( i , n - i );

else mainInverse( i , n - i );

}

}

print();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Never Ending Towers of Hanoi

\*\*\*ID: 10017

\*\*\*Juez: UVA

\*\*\*Tipo: Recursion

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int len;

#define MAX 256

int A[ MAX ] , lenA , B[ MAX ], lenB , C[ MAX ],lenC;

void move(int n, int s , int i , int d ){

if( n > 0 ){

move( n - 1 , s , d , i );

if( len < 0 ) return;

len--;

if( s == 1 ){

lenA--;

if( d == 2 ){ B[ lenB++ ] = A[ lenA ]; }

else if( d == 3 ){ C[ lenC++ ] = A[ lenA ]; }

}

else if( s == 2 ){

lenB--;

if( d == 1 ){ A[ lenA++ ] = B[ lenB ]; }

else if( d == 3 ){ C[ lenC++ ] = B[ lenB ]; }

}

else if( s == 3 ){

lenC--;

if( d == 1 ){ A[ lenA++ ] = C[ lenC ]; }

else if( d == 2 ){ B[ lenB++ ] = C[ lenC ]; }

}

printf("A=>");

if( lenA > 0 )printf(" ");

for( int j = 0 ; j < lenA ; ++j ) printf(" %d" , A[ j ] );

printf("\nB=>");

if( lenB > 0 )printf(" ");

for( int j = 0 ; j < lenB ; ++j ) printf(" %d" , B[ j ] );

printf("\nC=>");

if( lenC > 0 )printf(" ");

for( int j = 0 ; j < lenC ; ++j ) printf(" %d" , C[ j ] );

printf("\n\n");

if( len < 0 ) return;

move( n - 1 , i , s , d );

if( len < 0 ) return;

}

}

int main(){

int n , t = 1;

while( scanf("%d %d" , &n , &len ) , n | len ){

for( int i = n ; i > 0 ; --i ) A[ n - i ] = i;

len--;

lenA = n; lenB = lenC = 0;

printf("Problem #%d\n\n" , t++ );

printf("A=>");

if( lenA > 0 )printf(" ");

for( int j = 0 ; j < lenA ; ++j ) printf(" %d" , A[ j ] );

printf("\nB=>");

if( lenB > 0 )printf(" ");

for( int j = 0 ; j < lenB ; ++j ) printf(" %d" , B[ j ] );

printf("\nC=>");

if( lenC > 0 )printf(" ");

for( int j = 0 ; j < lenC ; ++j ) printf(" %d" , C[ j ] );

printf("\n\n");

move( n , 1 , 2 , 3 );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Reverse and Add

\*\*\*ID: 10018

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//old code

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

bool isPal(string c){

int l=c.length();

for(int i=0,j=l-1;i<=l/2 && j>=l/2;i++,j--){

if(c[i]!=c[j])return false;

}

return true;

}

long long toLong(string s){

stringstream ss(s);

long long n;

ss>>n;

return n;

}

string toStr(long long n){

stringstream ss;

ss<<n;

string s;

ss>>s;

return s;

}

int main(){

int casos,it,i;

cin>>casos;

string n,nr;

while(casos--){

cin>>n;

i=0;

while(i<1005){

nr=n;

reverse(nr.begin(),nr.end());

n=toStr(toLong(n)+toLong(nr));

i++;

if(isPal(n)){

cout<<i<<" "<<n<<endl;

break;

}

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Funny Encryption Method

\*\*\*ID: 10019

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc , simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int bitcount( int n ){

int cnt = 0;

while( n ){

n = n & ( n - 1 );

cnt++;

}

return cnt;

}

int pot( int a , int b ){

int r = 1;

for( int i = 1 ; i <= b ; ++i ) r \*= a;

return r;

}

int toBase10( string s , int base ){

int resp = 0, dig;

for( int i = s.length() - 1 , j = 0 ; i >= 0 ; --i, ++j ){

if( '0' <= s[ i ] && s[ i ] <= '9' ) dig = s[ i ] - '0';

else dig = ( s[ i ] - 'A' ) + 10;

resp += dig \* pot( base , j );

}

return resp;

}

string toStr( int n ){

stringstream ss;

ss<<n;

string s;

ss>>s;

return s;

}

int main(){

int n , t , b1 , b2;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

b1 = bitcount( n );

n = toBase10( toStr( n ) , 16 );

b2 = bitcount( n );

printf("%d %d\n" , b1 , b2 );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Shoemaker's Problem

\*\*\*ID: 10026

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1005

struct Node{

int day , cost, id;

bool operator <( const Node n1 ) const{

int f1 = day \* n1.cost , f2 = n1.day \* cost ;

return f1 < f2;

}

Node(){}

}a[ MAX ];

int main(){

int t , n;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

for( int i = 0 ; i < n ; ++i ){

scanf("%d %d" , &a[ i ].day , &a[ i ].cost );

a[ i ].id = i + 1;

}

stable\_sort( a ,a + n);

printf("%d" , a[ 0 ].id );

for( int i = 1 ; i < n ; ++i )

printf(" %d" , a[ i ].id );

printf("\n");

if( t > 0 ) printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Edit Step Ladders

\*\*\*ID: 10029

\*\*\*Juez: UVA

\*\*\*Tipo: LIS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 25005

#define MAXN 26

char s[ MAX ][ 17 ] , aux[ 17 ];

int len , dp[ MAX ] , l;

//borro s[ act ] en indice idx s[ act ][ idx ]

void borrar( int act , int idx ){

int i , l = 0;

for( i = 0 ; i < idx ; ++i ) aux[ l++ ] = s[ act ][ i ];

for( i = idx + 1 ; s[ act ][ i ] ; ++i ) aux[ l++ ] = s[ act ][ i ];

aux[ l++ ] = '\0';

}

//insert en s[ act] en indice idx caracter car s[ act ][ idx ] = car

void insertar( int act , int idx , int car ){

int i , l = 0;

for( i = 0 ; i < idx ; ++i ) aux[ l++ ] = s[ act ][ i ];

aux[ l++ ] = car + 'a';

for( i = idx ; s[ act ][ i ] ; ++i ) aux[ l++ ] = s[ act ][ i ];

aux[ l++ ] = '\0';

}

void cambiar( int act , int idx , int car ){

memcpy( aux , s[ act ] , sizeof( s[ act ] ) );

aux[ idx ] = car + 'a';

}

void transformar( int act , int op , int idx, int car ){

if( op == 0 ) borrar( act , idx );

else if( op == 2 ) insertar( act , idx , car );

else cambiar( act , idx , car );

}

int bs(){

int left = 0 , right = len , mid , idx = -1 , val;

while( left < right ){

mid = ( left + right ) >> 1;

val = strcmp( s[ mid ] , aux );

if( val == 0 ){

idx = mid;

break;

}

if( val < 0 ){

left = mid + 1;

}

else{

right = mid;

}

}

return idx;

}

int main(){

len = 0;

int ans = 0 , i , j , op , ll , idx;

for( i = 0 ; i < MAX ;++i ) dp[ i ] = 1;

while( scanf("%s" , &s[ len ] ) != EOF ){

for( op = 0 ; op < 3 ; ++op ){

for( i = 0 ; i < s[ len ][ i ] ; ++i ){

for( j = 0 ; j < 26 ; ++j ){

transformar( len , op , i , j );

//no tiene caso buscar palabras por que me lo dan ordenado

if( strcmp( s[ len ] , aux ) < 0 ) continue;

idx = bs();

if( idx >= 0 && dp[ len ] < dp[ idx ] + 1 ){

dp[ len ] = dp[ idx ] + 1;

}

}

}

}

len++;

}

for( i = 0 ; i < len ; ++i ) ans = max( ans , dp[ i ] );

printf("%d\n" , ans );

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Divisibility

\*\*\*ID: 100036

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

int dp[ MAX ][ 105 ][ 2 ] , a[ MAX ]; //posicion , suma % k , signo

int k, n;

int solve( int pos , int sum ){

if( pos == n ){

if( sum ) return 0;

else return 1;

}

if( sum < 0 ){ if( dp[ pos ][ sum ][ 0 ] != -1 ) return dp[ pos ][ sum ][ 0 ];}

else{

if( dp[ pos ][ sum ][ 1 ] != -1 ) { return dp[ pos ][ sum ][ 1 ]; }

}

int ans = ( solve( pos + 1 , ( sum % k + a[ pos ] % k ) % k ) | solve( pos + 1 , ( sum % k - a[ pos ] % k ) ) % k );

if( sum < 0 ) return dp[ pos ][ sum ][ 0 ] = ans;

return dp[ pos ][ sum ][ 1 ] = ans;

}

int main(){

int t;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &n , &k );

memset( dp , -1 , sizeof( dp ) );

for( int i = 0 ; i < n ; ++i ){

scanf("%d" , &a[ i ] );

a[ i ] %= k;

}

if( solve( 1 , a[ 0 ] % k ) > 0 ) puts("Divisible");

else puts("Not divisible");

}

return 0;

}

/\*

7

4 7

17 5 -21 15

4 5

17 5 -21 15

10 30

9 -97 3 5 -1 0 11 -46 19 17

10 97

9 -97 3 5 -1 0 11 -46 19 17

12 16

-11 7 -73 40 5 -2 66 21 -64 16 16 16

5 15

-1 -2 -3 -4 -5

5 9

-5 -4 2 -2 0

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Jolly Jumpers

\*\*\*ID: 10038

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 100005

bool seen[ MAX ];

int main(){

int n , ant,x;

while( scanf("%d" , &n ) != EOF ){

memset( seen , 0 , sizeof( seen ) );

scanf("%d" , &ant );

for( int i = 1 ; i < n && scanf("%d" , &x ) ; ++i ){

seen[ abs( x - ant ) ] = 1;

ant = x;

}

bool correct = true;

for( int i = 1 ; i <= n - 1 ; ++i ){

if( !seen[ i ] ){

correct = false;

break;

}

}

if( correct )

puts("Jolly");

else

puts("Not jolly");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Vito's family

\*\*\*ID: 10041

\*\*\*Juez: UVA

\*\*\*Tipo: Sorting, Median

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define pb push\_back

#define mp make\_pair

#define x first

#define y second

#define pii pair<int , int >

#define psi pair<string , int>

#define all( x ) x.begin(),x.end()

int MIN( int a , int b ){ return ( a < b ) ? a : b; }

int MAX( int a , int b ){ return ( a < b ) ? b : a; }

string toStr(int n){string s;ostringstream buffer;buffer<<n;s=buffer.str();return s;}

int toInt(string str){int n;istringstream buffer(str);buffer>>n;return n;}

long long pot(long long a,long long b){if(!b)return 1;if(b&1)return a\*pot(a\*a,b>>1); else return pot(a\*a,b>>1);}

#define MAX 505

int a[ MAX ] , n;

int getSum( int val ){

int sum = 0;

for( int i = 0 ; i < n ; ++i ){

sum += abs( val - a[i] );

}

return sum;

}

int main(){

int t , mini , maxi;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d" , &n ) ){

mini = 1<<30;

maxi = 0;

set<int> \_s;

for( int i = 0 ; i < n && scanf("%d" , &a[i]) ; ++i ){

mini = min( mini , a[i] );

maxi = max( maxi , a[i] );

}

sort( a , a + n );

int ans = 1<<30;

int median = n/2;

ans = getSum( a[ median ] );

if( median - 1 >= 0 )

ans = min( ans , getSum( a[ median - 1 ] ) );

if( median + 1 < n )

ans = min( ans , getSum( a[ median + 1 ] ) );

printf("%d\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Monocycle

\*\*\*ID: 10047

\*\*\*Juez: UVA

\*\*\*Tipo: BFS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 26

enum{ GREEN , BLACK , RED , BLUE , WHITE , N , S , W , E };

int h , w;

char ady[ MAX ][ MAX ];

struct Estado{

int x , y , d , color , dir;

Estado( int xx , int yy , int dd , int cc , int di ): x( xx ) , y( yy ) , d( dd ) , color( cc ) , dir( di ){}

};

int seen[ MAX ][ MAX ][ 5 ][ 10 ];

bool visto( Estado s ){

return seen[ s.x ][ s.y ][ s.color ][ s.dir ];

}

void visitado( Estado s ){

seen[ s.x ][ s.y ][ s.color ][ s.dir ] = 1;

}

int dx[ 4 ] = { -1 , 1 , 0 , 0 } , dy[ 4 ] = { 0 , 0 , -1 , 1 };

void bfs( int xini , int yini ){

int nx , ny , ans = -1;

memset( seen , 0 , sizeof( seen ) );

queue< Estado > Q;

Q.push( Estado( xini , yini , 0 , GREEN , N ) );

seen[ xini ][ yini ][ GREEN ][ N ] = 1;

while( !Q.empty() ){

Estado act = Q.front(); Q.pop();

if( ady[ act.x ][ act.y ] == 'T' && act.color == GREEN ){

ans = act.d;

break;

}

if( act.dir == N || act.dir == S ){

Estado vec = act;

vec.d++;

vec.dir = W;

if( !visto( vec ) ){

Q.push( vec );

visitado( vec );

}

vec.dir = E;

if( !visto( vec ) ){

Q.push( vec );

visitado( vec );

}

}

else if( act.dir == W || act.dir == E ){

Estado vec = act;

vec.dir = N;

vec.d++;

if( !visto( vec ) ){

Q.push( vec );

visitado( vec );

}

vec.dir = S;

if( !visto( vec ) ){

Q.push( vec );

visitado( vec );

}

}

nx = act.x + dx[ act.dir - N ];

ny = act.y + dy[ act.dir - N ];

if( nx >= 0 && nx < h && ny >= 0 && ny < w && ady[ nx ][ ny ] != '#' ){

Estado vec( nx , ny , act.d + 1 , ( act.color + 1 ) % ( WHITE + 1 ) , act.dir );

if( !visto( vec ) ){

Q.push( vec );

visitado( vec );

}

}

}

if( ans == -1 ) puts("destination not reachable");

else printf("minimum time = %d sec\n" , ans );

}

int main(){

int xini , yini , i , j, q = 0 ;

while( scanf("%d %d" , &h , &w ) , h | w ){

for( i = 0 ; i < h && scanf("%s" , &ady[ i ] ) ; ++i ){

for( j = 0 ; j < w ; ++j ){

if( ady[ i ][ j ] == 'S' ){ xini = i; yini = j; }

}

}

if( q )printf("\n");

printf("Case #%d\n" , ++q );

bfs( xini , yini );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Hartals

\*\*\*ID: 10050

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 3700

bool seen[ MAX ];

int main(){

int t , d, n, x;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d %d" , &d , &n ) ){

memset( seen , 0 , sizeof( seen ) );

for( int i = 0 ; i < n && scanf("%d" , &x ) ; ++i ){

int inc = x;

while( inc <= d ){

seen[ inc ] = 1;

inc += x;

}

}

int f = 6, s = 7, ans = 0;

for( int i = 1 ; i <= d ; ++i ){

if( i == f ){

f += 7;

continue;

}

if( i == s ){

s += 7;

continue;

}

if( seen[ i ] ){

ans++;

}

}

printf("%d\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Necklace

\*\*\*ID: 10054

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Euler Circuit

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 55

struct Edge{

int v, id;

Edge( int vv, int i ):v(vv), id(i){}

};

vector<Edge> ady[ MAX ];

int degree[ MAX ];

int aux[ MAX][ MAX ], V;

bool connected(){

for( int k = 1 ; k < V ; ++k ){

for( int i = 1 ; i < V ; ++i ){

for( int j = 1 ; j < V ; ++j ){

aux[ i ][ j ] |= aux[ i ][ k ] & aux[ k ][ j ];

}

}

}

for( int i = 1 ; i < V ; ++i ){

for( int j = 1 ; j < V ; ++j ){

if( i != j && degree[ i ] && degree[ j ] && !aux[ i ][ j ] )

return false;

}

}

return true;

}

deque<int> resp;

bool seen[ 1005 ];

void tour( int x , int id ){

while( ady[ x ].size() > 0 ){

Edge y = ady[ x ].back();

ady[ x ].pop\_back();

if( !seen[ y.id ] ){

seen[ y.id ] =true;

tour( y.v , y.id );

resp.push\_back( y.v );

}

}

}

int main(){

int t, E , u , v, min;

scanf("%d", &t );

for( int q = 1 ; q <= t ; ++q ){

if( q - 1 )printf("\n");

scanf("%d", &E );

for( int i = 0 ; i < MAX ; ++i )ady[ i ].clear();

memset( degree , 0 , sizeof( degree ) );

memset( aux , 0 ,sizeof( aux ) );

V = 1;

min = 9999999;

for( int i = 1 ; i <= E ; ++i ){

seen[ i ] = false;

scanf("%d %d", &u ,&v );

ady[ u ].push\_back( Edge( v , i ) );

ady[ v ].push\_back( Edge( u , i ) );

degree[ u ]++;

degree[ v ]++;

aux[ u ][ v ] = aux[ v ][ u ] = 1;

min = std::min( u , std::min( v , min) );

V = std::max( u , std::max( v, V ) );

}

V++;

//check connection

printf("Case #%d\n",q);

if( !connected() ){

printf("some beads may be lost\n");

continue;

}

//check circuit property

bool possible = true;

for( int i = 1 ; i < V ; ++i ){

if( degree[ i ] & 1 ){

possible = false;

break;

}

}

if( !possible ){

printf("some beads may be lost\n");

}

else{

for( int i = 1 ; i < V ; ++i ){

reverse(ady[ i ].begin() , ady[ i ].end() );

}

resp.clear();

tour( min , -1 );

for( int i = 0 ; i < resp.size() - 1; ++i ){

printf( "%d %d\n", resp[ i ] , resp[ i + 1 ]);

}

printf("%d %d\n", resp[ resp.size() - 1 ] , resp[ 0 ] );

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Jimmy's Riddles

\*\*\*ID: 10058

\*\*\*Juez: UVA

\*\*\*Tipo: Strings Grammar Check

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

/\*

STATEMENT = ACTION | STATEMENT , ACTION

ACTION = ACTIVE\_LIST VERB ACTIVE\_LIST

ACTIVE\_LIST = ACTOR | ACTIVE\_LIST and ACTOR

ACTOR = NOUN | ARTICLE NOUN

ARTICLE = a | the

NOUN = tom | jerry | goofy | mickey | jimmy | dog | cat | mouse

VERB = hate | love | know | like | VERBs

\*/

#define MAX 67000

char s[ MAX ] , token[ MAX ];

int idx , type , error;

char article[ 2 ][ 4 ] = { "a" , "the"};

char noun[ 8 ][ 8 ] = { "tom" , "jerry" , "goofy" , "mickey" , "jimmy" , "dog" , "cat" , "mouse" };

char verb[ 4 ][ 6 ] = { "hate" , "love" , "know" , "like" };

bool isLetter( char c ){ return c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z'; }

void next(){

if( error ){ type = 'f'; return;}

if( idx == strlen( s ) ){ type = 'f'; token[ 0 ] = '\0'; return; } //f - finish

//Eliminamos espacios en blanco

for( ; s[ idx ] ; ++idx ) if( s[ idx ] != ' ' ){ break;}

//Si puro espacios en blanco entonces terminamos

if( idx == strlen( s ) ){ type = 'f'; token[ 0 ] = '\0'; return; }

//Buscamos palabra

int len = 0 , i , j;

token[ 0 ] = '\0';

//Obtengo token

for( ; s[ idx ] ; ++idx ){

if( !isLetter( s[ idx ] ) )break;

token[ len++ ] = s[ idx ];

}

token[ len ] = '\0';

//si no tenglo palabra tengo coma

if( len == 0 ){

token[ 0 ] = ','; token[ 1 ] = '\0';

type = ','; idx++;

return;

}

//Article

for( i = 0 ; i < 2 ; ++i ) if( !strcmp( article[ i ] , token ) ){ type = 'a'; return; }

//Noun

for( i = 0 ; i < 8 ; ++i ) if( !strcmp( noun[ i ] , token ) ){ type = 'n'; return; }

//And

if( !strcmp("and" , token ) ){ type = '&'; return; }

//Verbo = Verbos | hates | ...

for( i = 0 ; i < 4 ; ++i ){

for( j = 0 ; j < len ; ++j )

if( token[ j ] != verb[ i ][ j ] ) break;

if( j == len ){ type = 'v'; return; }

for( ; j < len ; ++j )

if( token[ j ] != 's') break;

if( j == len ){ type = 'v'; return; }

}

error = 1;

type = 'f';

return;

}

int action(){

//( Articulo Noun | Noun ) and ( Articulo Noun | Noun )...

while( true ){

if( type == 'a' ) next();

if( type != 'n' ) return 0;

next();

if( type != '&' ) break;

next();

}

// \_\_\_\_ Verb

if( type != 'v' ) return 0;

next();

// ( Articulo Noun | Noun ) and ( Articulo Noun | Noun )...

while( true ){

if( type == 'a' ) next();

if( type != 'n' ) return 0;

next();

if( type != '&' ) break;

next();

}

return 1;

}

int statement(){

next();

while( type != 'f' ){

if( !action() ) return 0;

if( type == 'f' ) return 1;

if( type != ',') return 0;

next();

}

return 0;

}

int main(){

while( gets( s ) ){

token[ 0 ] = '\0'; idx = error = 0; type = -1;

if( statement() && !error ) puts("YES I WILL");

else puts("NO I WON'T");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# How many zeros and how many digits

\*\*\*ID: 10061

\*\*\*Juez: UVA

\*\*\*Tipo: Logarithms

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

/\*

Dado un numero N y una base B hallar:

- Numero de zeros al final del N! en base B

- Numero de digitos de N! en base B

\*/

#define MAX 1050005

double f[ MAX ];

int trailzero( int num , int base )

{

unsigned max = UINT\_MAX , factordim , i ;

for( i = 2 ; i <= base ; ++i )

{

factordim = 0;

while( base % i == 0 )

{

factordim++;

base /= i;

}

if( factordim > 0 )

{

int numfactor = 0 , j = i;

for ( ; j <= num; j \*= i )

numfactor += num / j;

if( max > numfactor / factordim )

max = numfactor / factordim;

}

}

return max;

}

int main(){

int num , base , i;

for( i = 1 ; i < MAX ; ++i ) f[ i ] = log( i ) + f[ i - 1 ];

while( scanf("%d %d" , &num , &base ) != EOF ){

printf("%u %0.lf\n" , trailzero( num , base ) ,

floor( f[ num ] / ( f[ base ] - f[ base - 1 ] ) + 1e-9 ) + 1.0 );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Tell me the frequencies

\*\*\*ID: 10062

\*\*\*Juez: UVA

\*\*\*Tipo: ad hoc, sorting

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1005

bool cmp( pair<int ,int> p1 , pair<int, int> p2 ){

return p1.first < p2.first || ( p1.first == p2.first && p1.second > p2.second );

}

int main(){

char s[MAX];

int l, idx;

bool b = false;

while( gets(s) ){

if( b ) putchar( '\n' );

b = true;

l = strlen( s );

int a[ 130 ] = { 0 };

for( int i = 0 ; i < l ; ++i ){

a[ (int)s[ i ] ]++;

}

vector< pair<int ,int > > v;

for( int i = 0 ; i < 130 ; ++i ){

if( a[ i ] ) v.push\_back( make\_pair( a[ i ] , i) );

}

sort( v.begin() , v.end() , cmp );

for( int i = 0 ; i < v.size(); ++i ){

printf("%d %d\n", v[ i ].second , v[ i ] .first);

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Knuth's Permutation

\*\*\*ID: 10063

\*\*\*Juez: UVA

\*\*\*Tipo: Next\_permutation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

#include <cstring>

#include <stdlib.h>

#include <cctype>

#include <cmath>

#include <cstdio>

#include <cstdlib>

#include <ctime>

#include <functional>

#include <numeric>

#include <utility>

#include <deque>

#include <stack>

#include <bitset>

#include <map>

#include <set>

#include <string>

#include <vector>

#include <queue>

#include <climits>

#include <fstream>

#include <list>

#include <sstream>

#include <iostream>

#include <iomanip>

using namespace std;

#define MAX 12

char s[ MAX ];

int len;

void solve( int l , string str ){

if( l == len ){

printf("%s\n" , str.c\_str() );

return;

}

int i;

string left , right , aux;

for( i = 0 ; i <= l ; ++i ){

left = str.substr( 0 , i );

right = str.substr( i );

aux = ""; aux += s[ l ];

solve( l + 1 , left + aux + right );

}

}

int main(){

int i , q = 0;

while( scanf("%s" , s ) != EOF ){

if( q ) printf("\n");

q = 1;

len = strlen( s );

string aux = ""; aux += s[ 0 ];

solve( 1 , aux );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Distinct Subsequences

\*\*\*ID: 10069

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

const int MAXD = 105, DIG = 9, BASE = 1000000000;

const unsigned long long BOUND = numeric\_limits <unsigned long long> :: max () - (unsigned long long) BASE \* BASE;

struct bignum

{

int D, digits [MAXD / DIG + 2];

inline void trim ()

{

while (D > 1 && digits [D - 1] == 0)

D--;

}

inline void init (long long x)

{

memset (digits, 0, sizeof (digits));

D = 0;

do

{

digits [D++] = x % BASE;

x /= BASE;

}

while (x > 0);

}

inline bignum (long long x)

{

init (x);

}

inline bignum (int x = 0)

{

init (x);

}

inline bignum (char \*s)

{

memset (digits, 0, sizeof (digits));

int len = strlen (s), first = (len + DIG - 1) % DIG + 1;

D = (len + DIG - 1) / DIG;

for (int i = 0; i < first; i++)

digits [D - 1] = digits [D - 1] \* 10 + s [i] - '0';

for (int i = first, d = D - 2; i < len; i += DIG, d--)

for (int j = i; j < i + DIG; j++)

digits [d] = digits [d] \* 10 + s [j] - '0';

trim ();

}

inline char \*str ()

{

trim ();

char \*buf = new char [DIG \* D + 1];

int pos = 0, d = digits [D - 1];

do

{

buf [pos++] = d % 10 + '0';

d /= 10;

}

while (d > 0);

reverse (buf, buf + pos);

for (int i = D - 2; i >= 0; i--, pos += DIG)

for (int j = DIG - 1, t = digits [i]; j >= 0; j--)

{

buf [pos + j] = t % 10 + '0';

t /= 10;

}

buf [pos] = '\0';

return buf;

}

inline bool operator < (const bignum &o) const

{

if (D != o.D)

return D < o.D;

for (int i = D - 1; i >= 0; i--)

if (digits [i] != o.digits [i])

return digits [i] < o.digits [i];

return false;

}

inline bool operator == (const bignum &o) const

{

if (D != o.D)

return false;

for (int i = 0; i < D; i++)

if (digits [i] != o.digits [i])

return false;

return true;

}

inline bignum operator << (int p) const

{

bignum temp;

temp.D = D + p;

for (int i = 0; i < D; i++)

temp.digits [i + p] = digits [i];

for (int i = 0; i < p; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum operator >> (int p) const

{

bignum temp;

temp.D = D - p;

for (int i = 0; i < D - p; i++)

temp.digits [i] = digits [i + p];

for (int i = D - p; i < D; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum range (int a, int b) const

{

bignum temp = 0;

temp.D = b - a;

for (int i = 0; i < temp.D; i++)

temp.digits [i] = digits [i + a];

return temp;

}

inline bignum operator + (const bignum &o) const

{

bignum sum = o;

int carry = 0;

for (sum.D = 0; sum.D < D || carry > 0; sum.D++)

{

sum.digits [sum.D] += (sum.D < D ? digits [sum.D] : 0) + carry;

if (sum.digits [sum.D] >= BASE)

{

sum.digits [sum.D] -= BASE;

carry = 1;

}

else

carry = 0;

}

sum.D = max (sum.D, o.D);

sum.trim ();

return sum;

}

};

#define MAXX 10005

#define MAXZ 105

char x[ MAXX ] ,z[ MAXZ ];

bignum dp[ MAXX ][ MAXZ] , seen[ MAXX ][ MAXZ ];

int lx , lz;

bignum solve( int lenx , int lenz ){

if( lenz == lz ) return bignum( 1 );

if( lenx == lx ) return bignum( 0 );

if( seen[ lenx ][ lenz ] == bignum( 1 ) ) return dp[ lenx ][ lenz ];

bignum ans = 0;

if( x[ lenx ] == z[ lenz ] ) ans = ans + solve( lenx + 1 , lenz + 1 ) + solve( lenx + 1 , lenz );

else ans = ans + solve( lenx + 1 , lenz );

seen[ lenx ][ lenz ] = 1;

return dp[ lenx ][ lenz ] = ans;

}

int main(){

int t , i, j;

scanf("%d" , &t );

while( t-- > 0 && scanf("%s%s" , &x , &z ) ){

lx = strlen( x ); lz = strlen( z );

for( i = 0 ; i <= lx ; ++i )

for( j = 0 ; j <= lz ; ++j ){

seen[ i ][ j ] = bignum( 0 );

}

printf("%s\n" , solve( 0 , 0 ).str() );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Take the land

\*\*\*ID: 10074

\*\*\*Juez: UVA

\*\*\*Tipo: DP, Maximum 2D sum

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 105

int h , w;

int a[ MAX ][ MAX ], sum , s , maxi, n;

int pr[ MAX ];

int Kadane2D(){

maxi = 0;

int resp = 0;

int k, l , x1 = 0 , x2 = 0 , y1 = 0, y2 = 0, j;

for( int z = 0 ; z < h; ++z ){

for( int i = 0 ; i < w ; ++i ) pr[ i ] = 0;

for( int x = z ; x < h ; ++x ){

s = 1<<31;

sum = 0;

for( int i = 0 ; i < w ; ++i ){

pr[ i ] += a[ x ][ i ];

sum += pr[ i ];

if( sum > s ){

s = sum;

}

if( sum < 0 ){

sum = 0;

}

}

if( s > maxi ){

maxi = s;

}

}

}

return maxi;

}

int main(){

int x;

while( scanf("%d %d" , &h , &w ) && h|w ){

for( int i = 0 ; i < h ; ++i ){

for( int j = 0 ; j < w ; ++j ){

scanf("%d" , &x );

if( x == 1 ) a[ i ][ j ] = -10050;

else a[ i ][ j ] = 1;

}

}

printf("%d\n" , Kadane2D() );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Airlines

\*\*\*ID: 10075

\*\*\*Juez: UVA

\*\*\*Tipo: Floyd Warshall ,Great Circle Distance

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#include <math.h>

#define MAX 105

int ady[ MAX ][ MAX ];

char cities[ MAX ][ 30 ];

int V , ncities;

int get\_Id( char \*s){

for( int i = 0 ; i < ncities ; ++i ){

if( strcmp( s , cities[ i ] ) == 0 )return i;

}

strcpy( cities[ ncities ] , s );

return ncities++;

}

struct City{

double lat , lon;

City( double l , double ll ): lat( l ) , lon( ll ){}

};

#define R 6378

double rad = 3.141592653589793 / 180.;

int greatCircle( double a\_lat , double a\_long , double b\_lat , double b\_long ){

double t = cos( a\_lat ) \* cos( b\_lat ) \* cos( a\_long - b\_long ) + sin( a\_lat ) \* sin( b\_lat );

if( t < -1 )t = -1;

if( t > 1 ) t = 1;

return (int)floor( acos( t ) \* R + 0.5 );

}

#define INF 999999999

void Init(){

for( int i = 0 ; i < V ; ++i ){

for( int j = 0 ; j < V ; ++j ){

ady[ i ][ j ] = INF;

}

ady[ i ][ i ] = 0;

}

}

void Floyd(){

for( int k = 0 ; k < V ; ++k ){

for( int i = 0 ; i < V ; ++i ){

for( int j = 0 ; j < V ; ++j ){

int t = ady[ i ][ k ] + ady[ k ][ j ];

if( ady[ i ][ j ] > t ){

ady[ i ][ j ] = t;

}

}

}

}

}

int main(){

int E, Q , u , v , q = 1;

char s[ 30 ] , s2[ 30 ];

double lat, lon;

while( scanf("%d %d %d" ,&V ,&E, &Q ) && ( V + E + Q ) ){

if( q > 1 )printf("\n");

ncities = 0;

double vv[ V ][ 2 ];

for( int i = 0 ; i < V ; ++i ){

scanf("%s %lf %lf" , &s , &lat , &lon );

u = get\_Id( s );

vv[ u ][ 0 ] = lat\*rad;

vv[ u ][ 1 ] = lon\*rad;

}

Init();

while( E-- ){

scanf("%s %s", &s, &s2 );

u = get\_Id( s ); v = get\_Id( s2 );

ady[ u ][ v ] = greatCircle( vv[ u ][ 0 ] , vv[ u ][ 1 ] , vv[ v ][ 0 ] , vv[ v ][ 1 ] );

}

Floyd();

printf("Case #%d\n" , q++ );

while( Q-- ){

scanf("%s %s" ,&s , &s2 );

u = get\_Id( s ); v = get\_Id( s2 );

if( ady[ u ][ v ] == INF ) printf("no route exists\n");

else printf("%d km\n" , ady[ u ][ v ] );

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Gopher II

\*\*\*ID: 10080

\*\*\*Juez: UVA

\*\*\*Tipo: Maximum Bipartite Matching, max flow

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

struct Point{

double x, y;

Point( double xx , double yy ): x(xx), y(yy){}

};

/\*\*

u: nodo origen

v: nodo destino

c: capacidad de nodo u a v

r: indice de arista inversa, ejemplo si arista act 1->2 retorno es 1<-2

\*\*/

struct Edge{

int u;

int v;

int c;

int r;

Edge( int uu, int vv, int cc, int rr): u(uu), v(vv), c(cc), r(rr){}

};

vector< Edge > E;

vector< vector<int> > ady; //posee indices de aristas

int maxflow( int source , int sink ){

for( int f = 0; ;){

///vector de enlaces previos para ver la ruta recorrida del argumenting path

///tiene indices de aristas

vector<int> prev( ady.size() , -1 );

queue<int> Q;

Q.push( source );

while( !Q.empty() ){

int u = Q.front(); Q.pop();

for( int i = 0 ; i < ady[ u ].size(); ++i ){

///indice de arista actual

int e = ady[ u ][ i ];

int v = E[ e ].v;

///si no puede pasar o si ya fue visitado

if( E[ e ].c == 0 || prev[ v ] >= 0 )continue;

Q.push( v );

prev[ v ] = e;

}

}

///si ya no hay argumenting paths

if( prev[ sink ] < 0 )return f;

///hallamos minima capacidad de argumentin path

int c = 0x7fffffff;

for( int x = sink ; x != source ; x = E[ prev[ x ] ].u ){

c <?= E[ prev[ x ] ].c;

}

///para nodos de ida 1->2 restamos capacidad minima

///para nodos de vuevla 2<-1 sumamos capacidad minima

for( int x = sink ; x != source ; x = E[ prev[ x ] ].u ){

E[ prev[ x ] ].c -= c;

E[ E[ prev[ x ] ].r ].c += c;

}

f += c;

}

}

///indices de aristas comienzan x cero

///si tengo 1->2 su indice es 0, si 1<-2 su indice es 1

///x tanto r de primero es 1, de segundo es 0

void add( int u , int v , int c , bool dirigido = true ){

Edge e1( u , v , c , E.size() + 1 );

ady[ u ].push\_back( E.size() );

E.push\_back( e1 );

///capadidad 0 si es dirigido

Edge e2( v , u , ( dirigido )? 0 : c , E.size() - 1 );

ady[ v ].push\_back( E.size() );

E.push\_back( e2 );

}

double d( Point p1, Point p2 ){

return (p1.x - p2.x)\*(p1.x - p2.x) + (p1.y - p2.y)\*(p1.y - p2.y);

}

int main(){

int g, h, s ,u , v , c, dist;

double x, y;

while( scanf("%d %d %d %d",&g, &h, &s, &v ) != EOF ){

E.clear();

ady = vector< vector<int> > ( g + h + 5 , vector<int>() );

dist = s \* s \* v \* v;

vector<Point> p;

for( int i = 0 ; i < g + h ; ++i ){

scanf("%lf %lf", &x, &y );

p.push\_back( Point( x , y ) );

}

for( int i = 0 ; i < g ; ++i ){

add( 0 , i + 1 , 1 );

for( int j = g ; j < g + h ; ++j ){

if( d( p[ i ] , p[ j ] ) <= dist ){

add( i + 1 , j + 1 , 1 );

}

}

}

for( int i = g ; i < g + h ; ++i )add( i + 1 , g + h + 1 , 1 );

printf("%d\n", g - maxflow( 0 , g + h + 1 ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Marbles

\*\*\*ID: 10090

\*\*\*Juez: UVA

\*\*\*Tipo: diophantine equation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

long long c1 , c2;

/\*

Algoritmo Extendido de Euclides

ax + by = gcd( a , b ) --- |x| + |y| es minimo

Retorna el gcd( a , b)

\*/

long long extended\_euclid( long long a , long long b , long long &x , long long &y ){

if( b == 0 ){ x = 1; y = 0; return a; }

long long d = extended\_euclid( b , a % b , y , x );

y -= a/b \* x;

return d;

}

//a%b

long long mod( long long a , long long b ){

return ( (a % b) + b ) % b;

}

//Calcula una solucion para x e y tal que ax + by = c -> x , y >=0

bool linear\_diophantine( long long a , long long b , long long c , long long &x , long long &y ){

long long d = extended\_euclid( a , b , x , y );

if( c % d ) return false;

long long x1 = x \* c/d , y1 = y \* c/d , A = a/d , B = b/d , k;

x = mod( x1 , B );

k = ( c - a \* x )/b;

long long mini = 1LL<<62 , X = -1 , Y;

if( k >= 0 && c1 \* x + k \* c2 < mini ){

mini = c1 \* x + k \* c2;

X = x; Y = k;

}

y = mod( y1 , A );

k = ( c - b \* y )/a;

if( k >= 0 && c1 \* k + y \* c2 < mini ){

X = k; Y = y;

}

x = X; y = Y;

return (x >= 0);

}

int main(){

long long a , c , b , x , y;

while( scanf("%lld" , &c ) , c ){

scanf("%lld %lld %lld %lld" , &c1 , &a , &c2 , &b );

if( linear\_diophantine( a , b , c , x , y ) )printf("%lld %lld\n" , x , y );

else puts("failed");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Problem with the Problem Setter

\*\*\*ID: 10092

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Maximum Bipartite Matching

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#include <vector>

using namespace std;

#define M 30

#define N 1005

bool graph[ N ][ M ];

bool seen[ M ];

int matchL[ N ], matchR[ M ];

int n, m;

bool bpm( int u ){

for( int v = 1; v <= m; v++ ){

if( graph[u][v] ){

if( seen[v] ) continue;

seen[v] = true;

if( matchR[ v ] > 0 ){

matchL[ u ] = v;

matchR[ v ]--;

return true;

}

for( int k = 1 ; k <= n ; ++k ){

if( matchL[ k ] == v && bpm( k ) ){

matchL[ u ] = v;

return true;

}

}

}

}

return false;

}

int main(){

int k , cat, ncat;

while( scanf("%d %d" , &m , &n ) && m|n ){

ncat = 0;

for( int i = 1 ; i <= m ; ++i ){

scanf("%d" , &matchR[ i ] );

ncat += matchR[ i ];

}

for( int i = 1 ; i <= n ; ++i ){

scanf("%d" , &k );

while( k-- ){

scanf("%d" , &cat );

graph[ i ][ cat ] = 1;

}

}

memset( matchL, -1 , sizeof( matchL ) );

int i, cnt = 0;

for( i = 1; i <= n; ++i ){

memset( seen, 0, sizeof( seen ) );

if( bpm( i ) ) cnt++;

}

if( cnt != ncat )printf("0\n");

else{

printf("1\n");

vector< int > v[ M ];

for( int i = 1 ; i <= n ; ++i ){

if( matchL[ i ] != -1 ){

v[ matchL[ i ] ].push\_back( i );

}

}

for( int i = 1 ; i <= m; ++i ){

for( int j = 0 ; j < v[ i ].size() ; ++j ){

if( j != 0 )printf(" ");

printf("%d" , v[ i ][ j ] );

}

printf("\n");

}

}

memset( graph , 0 , sizeof( graph ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Generating Fast Sorted Permutation

\*\*\*ID: 10098

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, next\_permutation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t;

scanf("%d", &t );

char ss[ 20 ];

for( int i = 0 ; i < t ; ++i ){

scanf("%s" , ss );

string s(ss);

sort( s.begin() , s.end() );

do{

printf("%s\n", s.c\_str());

}while( next\_permutation( s.begin() , s.end() ) );

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Tourist Guide

\*\*\*ID: 10099

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Floyd Warshall, Maxmin

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 101

int ady[ MAX ][ MAX ] , V;

void Init(){

for( int i = 0 ; i < V ; ++i ){

for( int j = 0 ; j < V ; ++j ){

ady[ i ][ j ] = -1;

}

ady[ i ][ i ] = 0;

}

}

int min(int a, int b){

return (a < b) ? a: b;

}

int max(int a, int b){

return (a > b) ? a: b;

}

void MaxMin(){

for( int k = 0 ; k < V ; ++k ){

for( int i = 0 ; i < V ; ++i ){

for( int j = 0 ; j < V ; ++j ){

ady[ i ][ j ] = max( ady[ i ][j ] , min( ady[ i ][k ] , ady[ k ][ j ] ) );

}

}

}

}

int main(){

int E , u , v , w, q = 1 , ans;

while( scanf( "%d %d" , &V , &E ) && V|E ){

Init();

while( E-- ){

scanf("%d %d %d" , &u , &v , &w );

u--; v--;

ady[ u ][ v ] = ady[ v ][ u ] = w;

}

scanf("%d %d %d" ,&u ,&v ,&w );

u--; v--;

MaxMin();

ady[ u ][ v ]--;

printf("Scenario #%d\n" , q++ );

ans = w/ady[ u ][ v ];

if( w % ady[ u ][ v ] != 0 )ans++;

printf("Minimum Number of Trips = %d\n\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Longest Match

\*\*\*ID: 10100

\*\*\*Juez: UVA

\*\*\*Tipo: LCS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAXN 1005

char line[ MAXN ];

int A[ MAXN ] , B[ MAXN ] , lenA , lenB , dp[ MAXN ][ MAXN ];

map< string , int > mp;

bool isLetter( char c ){ return c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z'; }

int MAX( int a , int b , int c ){ return max( a , max( b , c ) ); }

int LCS(){

memset( dp , 0 , sizeof( dp ) );

int i , j;

for( i = 1 ; i <= lenA ; ++i )

for( j = 1 ; j <= lenB ; ++j )

dp[ i ][ j ] = MAX( dp[ i - 1 ][ j ] , dp[ i ][ j - 1 ] , dp[ i - 1 ][ j - 1 ] + ( A[ i - 1 ] == B[ j - 1 ] ) );

return dp[ lenA ][ lenB ];

}

int main(){

map< string , int > :: iterator it;

int i , blank , q = 1 , len;

while( gets( line ) ){

mp.clear();

blank = len = 0;

for( i = 0 ; line[ i ] ; ++i ) if( !isLetter( line[ i ] ) && !isdigit( line[ i ] ) ) line[ i ] = ' ';

if( strlen( line ) == 0 ) blank = 1;

else{

stringstream ss( line );

lenA = 0;

while( ss>>line ){

if( !mp.count( line ) ){ mp[ line ] = len++; A[ lenA ] = len - 1; }

else A[ lenA ] = mp[ line ];

lenA++;

}

}

gets( line );

for( i = 0 ; line[ i ] ; ++i ) if( !isLetter( line[ i ] ) && !isdigit( line[ i ] ) ) line[ i ] = ' ';

if( strlen( line ) == 0 ) blank = 1;

else{

lenB = 0;

stringstream ss( line );

while( ss>>line ){

if( !mp.count( line ) ){ mp[ line ] = len++; B[ lenB ] = len - 1; }

else B[ lenB ] = mp[ line ];

lenB++;

}

}

printf("%2d. " , q++ );

if( blank )puts("Blank!");

else printf("Length of longest match: %d\n" , LCS() );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Bangla Numbers

\*\*\*ID: 10101

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int a[ 5 ] = { 10000000 , 100000 , 1000 , 100 };

char s[ 5 ][ 6 ] = { "kuti" , "lakh" , "hajar" , "shata" };

int main(){

long long n , res , div , d , r;

int q = 1 , i , j;

while( scanf("%lld" , &n ) != EOF ){

printf("%4d." , q++ );

if( !n ){

printf(" 0\n");

continue;

}

for( i = 0 ; i < 4 ; ++i ){

div = n / a[ i ];

if( div > 0 ){

while( div > 99 ){

for( j = 0 ; j < 4 ; ++j ){

d = div / a[ j ];

if( d > 0 ) printf(" %lld %s" , d , s[ j ] );

div %= a[ j ];

}

}

if( div > 0 ) printf(" %lld %s" , div , s[ i ] );

else printf(" %s" , s[ i ] );

}

n = n % a[ i ];

}

if( n > 0 ) printf(" %d" , n );

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# What is the Median

\*\*\*ID: 10107

\*\*\*Juez: UVA

\*\*\*Tipo: Sorting, mediam selection

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

int main(){

int n, l = 0;

long long num;

vector<long long> a;

vector<long long >::iterator low;

while( cin>>num ){

low=lower\_bound ( a.begin() , a.end(), num);

a.insert( low, num );

l = a.size();

if( l & 1 )cout<<a[ l/2 ]<<endl;

else cout<<( a[ l/2 ] + a[ l/2 - 1] )/2<<endl;

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Light , more light

\*\*\*ID: 10110

\*\*\*Juez: UVA

\*\*\*Tipo: Perfect Square

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int numD( long long n ){

long long i;

int cnt = 0 , ans = 1;

for( i = 2 ; i \* i <= n ; ++i ){

cnt = 0;

while( n % i == 0 ){ cnt++; n /= i;}

ans \*= ( cnt + 1 );

}

if( n > 1 ) ans \*= 2;

return ans;

}

int main(){

long long n , i;

map< long long , bool > mp;

mp[ 1 ] = 1;

for( i = 2 ; ; ++i ){

mp[ i \* i ] = 1;

if( i \* i > UINT\_MAX ) break;

}

while( scanf("%lld" , &n ) , n ){

printf("%s\n" , mp[ n ] == 1 ?"yes":"no" );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Loansome Car Buyer

\*\*\*ID: 10114

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc , simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int m , t , x , val , borrow;

double v;

double pay , loan , dep[ 105 ] , payoff;

while( scanf("%d %lf %lf %d" , &m , &pay , &loan , &t ) ,m > -1 ){

memset( dep , 0 , sizeof( dep ) );

for( int i = 0 ; i < t ; ++i ){

scanf("%d %lf" , &x , &v );

dep[ x ] = v;

}

for( int i = 1 ; i <= m ; ++i ){

if( dep[ i ] == 0 ) dep[ i ] = dep[ i - 1 ];

}

val = pay + loan;

int i = 0;

payoff = loan/(double)m;

for( ; i < m ; ++i ){

val = val - val \* dep[ i ] + 1E-9;

if( i )loan -= payoff;

if( loan < val ) break;

}

if( i == 1 )puts("1 month");

else printf("%d months\n" , i );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Automatic Editing

\*\*\*ID: 10115

\*\*\*Juez: UVA

\*\*\*Tipo: String

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int n;

char cad[ 205 ];

string s;

char find[ 12 ][ 90 ], replace[ 12 ][ 90 ];

while( scanf("%d" , &n ) , n ){

gets( cad );

for( int i = 0 ; i < n ; ++i ){

gets( find[ i ] );

gets( replace[ i ] );

}

gets( cad );

s = string( cad );

int k = 0, aux, i = 0 , j;

for( i = 0 ; i < n ; ++i ){

string f = string( find[ i ] );

k = 0;

while( k < s.length() ){

aux = k;

j = 0;

for( ; j < f.length() && aux < s.length() ; ++j ){

if( s[ aux ] == f[ j ] ) aux++;

else break;

}

if( j == f.length() ){

s.replace( k , f.length() , string(replace[ i ] ) );

k = 0;

}

else k++;

}

}

printf("%s\n" , s.c\_str() );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Robot Motion

\*\*\*ID: 10116

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, BFS, simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstdio>

#include <cstring>

#define MAX 605

char tabla[ MAX ][ MAX ];

int dist[ MAX ][ MAX ];

int main(){

int f , c , x , y , dist\_ant;

while( scanf("%d %d %d" , &f , &c , &y ) , f | c | y ){

memset( dist , 0 , sizeof( dist ) );

for( int i = 0 ; i < f ; ++i ) scanf("%s" , &tabla[ i ] );

x = 0;

dist\_ant = 0;

y--;

while( 1 ){

if( dist[ x ][ y ] ){

printf("%d step(s) before a loop of %d step(s)\n" , dist[ x ][ y ] - 1 , dist\_ant - dist[ x ][ y ] + 1 );

break;

}

//si exceo

if( x < 0 || y < 0 || x >= f || y >= c ){

printf("%d step(s) to exit\n" , dist\_ant );

break;

}

dist[ x ][ y ] = dist\_ant + 1;

dist\_ant = dist[ x ][ y ];

if( tabla[ x ][ y ] == 'N' ) x--;

else if( tabla[ x ][ y ] == 'S' ) x++;

else if( tabla[ x ][ y ] == 'E' ) y++;

else y--;

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Play on Words

\*\*\*ID: 10129

\*\*\*Juez: UVA

\*\*\*Tipo: Union Find, DFS, Euler Tour

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 26

int V;

int ady[ MAX ][ MAX ];

bool seen[ MAX ];

int in[ MAX ], out[ MAX ];

int parent[MAX],rank[MAX];

void Make\_Set(const int x){

parent[x]=x;

rank[x]=0;

}

int Find(const int x){

if(parent[x]!=x) parent[x]=Find(parent[x]);

return parent[x];

}

void Union(const int PX, const int PY){

if(rank[PX]>rank[PY]) parent[PY]=PX;

else{

parent[PX]=PY;

if(rank[PX]==rank[PY]) rank[PY]++;

}

}

void Merge(const int x,const int y){

Union(Find(x),Find(y));

}

int bosque(){

set<int> cc;

for( int i = 0; i < MAX ;i++){

if( out[ i ] + in[ i ] > 0 )cc.insert(Find(i));

}

return cc.size();

}

bool connected( ){

for( int i = 0 ; i < MAX ;++i){

if( out[ i ] + in[ i ] > 0 )Make\_Set( i );

}

for( int i = 0 ; i < MAX ;++i ){

for( int j = 0 ; j < MAX ; ++j ){

if( ady[ i ][ j ] ){

if( Find( i ) != Find( j ) ){

Merge( i , j );

}

}

}

}

if( bosque() != 1)return false;

return true;

}

int main(){

int t , ini;

scanf("%d", &t);

char s[ 1005 ];

while( t-- ){

scanf("%d", &V);

memset( in , 0 , sizeof( in ) );

memset( out , 0 , sizeof( out ) );

memset( ady , 0 , sizeof( ady ));

for( int i = 0 ; i < V ;++i ){

scanf("%s", s );

int u = s[ strlen( s ) - 1 ] - 'a';

int v = s[ 0 ] - 'a';

ady[ u ][ v ] = 1;

ady[ v ][ u ] = 1;

in[ v ]++;

out[ u ]++;

}

if( !connected( ) ){

printf("The door cannot be opened.\n");

continue;

}

int cont1 = 0, cont2 = 0, V\_alpha = 0, cont4 = 0;

bool possible = false;

for( int i = 0 ; i < MAX ;++i ){

if( out[ i ] + in[ i ] > 0 ){

V\_alpha++;

if( out[ i ] == in[ i ] + 1 )cont1++;

else if( in[ i ] == out[ i ] + 1 )cont2++;

else if( out[ i ] == in[ i ] )cont4++;

}

}

if( V\_alpha == cont4 || ( cont1 < 2 && cont2 < 2 && cont4 + cont1 + cont2 == V\_alpha ) )possible = true;

if( !possible )printf("The door cannot be opened.\n");

else printf("Ordering is possible.\n");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Play on Words

\*\*\*ID: 10130

\*\*\*Juez: UVA

\*\*\*Tipo: DP, Knapsack

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int dp[ 305 ] , w[ 1005], v[ 1005 ];

int main(){

int t, n , p, capacity, x;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n);

for( int i = 0 ; i < n ; ++i ) scanf("%d %d" , &v[ i ] , &w[ i ] );

scanf("%d" , &p );

capacity = 0;

while( p-- ){

scanf("%d" , &x );

memset( dp , 0 , sizeof( dp ) );

for( int i = 0 ; i < n ; ++i ){

for( int j = x ; j >= w[ i ] ; --j ){

dp[ j ] = max( dp[ j ] , dp[ j - w[ i ] ] + v[ i ] );

}

}

capacity += dp[ x ];

}

printf("%d\n" , capacity );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# CDVII

\*\*\*ID: 10138

\*\*\*Juez: UVA

\*\*\*Tipo: Sorting

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>>

using namespace std;

#define MAX 24

#define MAXN 10005

int a[ MAX ];

map<string , int> cost;

int len;

struct Data{

string number , time, op;

int km, month , day , hour , minute;

void convert(){

string t = "";

for( int i = 0 ; i < time.length() ; ++i ){

if( time[i] == ':' )

t += " ";

else

t += time[i];

}

stringstream ss(t);

ss>>month>>day>>hour>>minute;

}

Data(){}

}data[ MAXN ];

bool cmp( Data d1 , Data d2 ){

return d1.number < d2.number || ( d1.number == d2.number && d1.time < d2.time );

}

int main(){

int t;

scanf("%d", &t );

string line;

for( int qq = 0 ; qq < t ; ++qq ){

if( qq )

printf("\n");

for( int i = 0 ; i < 24 && scanf("%d" , &a[i]) ; ++i );

len = 0; cost.clear();

getline( cin , line );

while( getline( cin , line ) && line.length() > 0 ){

stringstream ss( line );

Data d;

ss>>d.number>>d.time>>d.op>>d.km;

d.convert();

data[ len++ ] = d;

}

sort( data , data + len , cmp );

for( int i = 0 ; i < len - 1 ; ++i ){

Data d = data[ i ] , next = data[ i + 1 ];

if( d.number == next.number && d.op == "enter" && next.op == "exit" ){

int v1 = a[ d.hour ], v2 = a[ next.hour ];

cost[ d.number ] += v1 \* abs( next.km - d.km ) + 100;

}

}

for( map<string , int> :: iterator it = cost.begin() ; it != cost.end() ; ++it ){

printf("%s $%.2lf\n" , (it -> first).c\_str() , (it -> second + 200)/100.0 );

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Factovisors

\*\*\*ID: 10139

\*\*\*Juez: UVA

\*\*\*Tipo: Prime Factors

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>>

using namespace std;

map< long long , int > getFactors( long long num ){

map< long long , int > factores;

for( long long i = 2 ; i \* i <= num ; ++i ){

while( num % i == 0 ){

factores[ i ]++;

num /= i;

}

}

if( num > 1 ) factores[ num ]++;

return factores;

}

int main(){

int s;

long long n , m , prime , aux;

bool possible;

map< long long , int >::iterator it;

map< long long , int > fM;

while( scanf("%lld %lld" , &n , &m ) != EOF ){

fM = getFactors( m );

possible = true;

for( it = fM.begin() ; it != fM.end() ; ++it ){

s = 0;

prime = it -> first;

aux = n;

while( prime <= aux ){

s += aux/prime;

prime \*= it -> first;

if( s >= it -> second ) break;

}

if( s < it -> second ){ possible = false; break;}

}

if( !possible ) printf("%lld does not divide %lld!\n" , m , n );

else printf("%lld divides %lld!\n" , m , n );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Prime Distance

\*\*\*ID: 10140

\*\*\*Juez: UVA

\*\*\*Tipo: Segmented Sieve

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define DIF 1000005 //L-U

#define MAX 46350 //sqrt( U )

int primes[ MAX ] , isPrime[ MAX ] , len , is\_Prime[ DIF ];

void sieve( int n ){

memset( isPrime , 1 , sizeof( isPrime ) );

isPrime[ 0 ] = 0;

isPrime[ 1 ] = 0;

int i , k ;

for( i = 2 ; i \* i <= n ; ++i ){

if( isPrime[ i ] ){

for( k = 2 ; k \* i <= n ; ++k){

isPrime[ k \* i ] = 0;

}

}

}

len = 0;

if( n >= 2 ) primes[ len++ ] = 2;

for( i = 3 ; i <= n ; i += 2 ){

if( isPrime[ i ] ) primes[ len++ ] = i;

}

}

long long v[ DIF ];

int sz;

void segmentedSieve( int L , int U ){

if( L < 2 ) L = 2;

memset( is\_Prime , 1 , sizeof( is\_Prime ) );

int root;

long long j , start , p , i ;

root = ( int )sqrt( U ) + 1;

for( i = 0 ; i < len ; ++i ){

p = primes[ i ];

if( p >= root ) break;

start = ( p >= L ) ? p \* 2 : ( L + ( p - L % p ) % p );

for( j = start ; j <= U ; j += p ){

is\_Prime[ j - L ] = false;

}

}

start = ( L % 2 ? L : L + 1 );

sz = 0;

if( L == 2 ) v[ sz++ ] = 2;

for( i = start ; i <= U ; i += 2 )if( is\_Prime[ i - L ] ) v[ sz++ ] = i;

}

int main(){

sieve( MAX );

int L , U , i;

long long xmini , ymini , xmaxi , ymaxi , mini , maxi;

while( scanf("%d %d" , &L , &U ) != EOF ){

segmentedSieve( L , U );

if( sz < 2 ){ puts("There are no adjacent primes."); continue; }

mini = 1LL<<50;

maxi = -1;

for( i = 1 ; i < sz ; ++i ){

if( v[ i ] - v[ i - 1 ] < mini ){

mini = v[ i ] - v[ i - 1 ];

xmini = v[ i - 1 ];

ymini = v[ i ];

}

if( v[ i ] - v[ i - 1 ] > maxi ){

maxi = v[ i ] - v[ i - 1 ];

xmaxi = v[ i - 1 ];

ymaxi = v[ i ];

}

}

printf("%lld,%lld are closest, %lld,%lld are most distant.\n" , xmini , ymini , xmaxi , ymaxi );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# ShellSort

\*\*\*ID: 10152

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t , n;

char name[ 256 ][ 128 ] , s[ 256 ];

int resp[ 256 ] , j;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

getchar();

for( int i = 0 ; i < n ; ++i ){

gets( name[ i ] );

}

for( int i = 0 ; i < n ; ++i ){

gets( s );

for( j = 0 ; j < n ; ++j ){

if( strcmp( s , name[ j ] ) == 0 ){

break;

}

}

resp[ i ] = j;

}

int i;

for( i = n - 2 ; i >= 0 && resp[ i ] < resp[ i + 1 ] ; --i );

while( i >= 0 )printf("%s\n" , name[ resp[ i-- ] ] );

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# War

\*\*\*ID: 10158

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Connectivity, Union Find

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

#define MAX\_SIZE 20005

///UNION-FIND

int parent[MAX\_SIZE],rank[MAX\_SIZE];

void Make\_Set(const int x){

parent[x]=x;

rank[x]=0;

}

int Find(const int x){

if(parent[x]!=x) parent[x]=Find(parent[x]);

return parent[x];

}

void Union(const int PX, const int PY){

if(rank[PX]>rank[PY]) parent[PY]=PX;

else{

parent[PX]=PY;

if(rank[PX]==rank[PY]) rank[PY]++;

}

}

void Merge(const int x,const int y){

Union(Find(x),Find(y));

}

///END UNION FIND

using namespace std;

int n;

///para enemigos

int other( int nodo ){

if( nodo >= n )return nodo - n ;

return n + nodo;

}

int main(){

int c , x, y;

while ( scanf("%d", &n) != EOF ){

for( int i = 0 ; i < MAX\_SIZE ; ++i )Make\_Set(i);

while( scanf("%d %d %d", &c ,&x ,&y ) ){

if( c == 0 && x == 0 && y == 0)break;

if( c == 1 ){

///si son enemigos no tienen xq ser amigos

if( Find( x ) == Find( other(y) ) || Find( y ) == Find( other(x) ) ){

printf("-1\n");

}

else{

Merge( x , y );

Merge( other( x ) , other( y ) );

}

}

else if( c == 2 ){

///si son amigos no tienen xq ser enemigos

if( Find( other( x ) ) == Find( other( y ) ) || Find( x ) == Find( y ) ){

printf("-1\n");

}

else{

Merge( x , other(y) );

Merge( y , other(x) );

}

}

else if( c == 3 ){

if( Find( x ) == Find( y ) || Find( other( x ) ) == Find( other( y ) ) ){

printf("1\n");

}

else printf("0\n");

}

else if( c == 4 ){

if( Find( x ) == Find( other( y ) ) || Find( y ) == Find( other( x ) ) ){

printf("1\n");

}

else printf("0\n");

}

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Ant on a Chessboard

\*\*\*ID: 10161

\*\*\*Juez: UVA

\*\*\*Tipo: Recurrences

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

long long n , pos , x , y;

while( scanf("%lld" , &n ) , n ){

pos = ( long long ) ceil( sqrt( n ) ) - 1;

if( pos & 1 ){

if( pos \* pos + pos + 1 >= n ){

y = pos + 1;

x = n - pos \* pos;

}

else{

x = pos + 1;

y = ( pos + 1 ) - ( n - ( pos \* pos + pos + 1 ) );

}

}

else{

if( pos \* pos + pos + 1 >= n ){

x = pos + 1;

y = n - pos \* pos;

}

else{

y = pos + 1;

x = ( pos + 1 )- ( n - ( pos \* pos + pos + 1 ) ) ;

}

}

printf("%lld %lld\n" , x , y );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Meeting Prof. Miguel

\*\*\*ID: 10171

\*\*\*Juez: UVA

\*\*\*Tipo: Floyd Warshall, Shortest Path

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 26

#define INF 9999999

int ady[ MAX ][ MAX ];

char who[ MAX ][ MAX ];

int young[ MAX ][ MAX ];

void Init(){

for( int i = 0 ; i < MAX ; ++i ){

for( int j = 0 ; j < MAX ; ++j ){

ady[ i ][ j ] = INF;

young[ i ][ j ] = INF;

}

ady[ i ][ i ] = 0;

young[ i ][ i ] = 0;

}

}

void FloydYoung( ){

for( int k = 0 ; k < MAX ; ++k ){

for( int i = 0 ; i < MAX ; ++i ){

for( int j = 0 ; j < MAX ; ++j ){

int t = young[ i ][ k ] + young[ k ][ j ];

if( young[ i ][ j ] > t ){

young[ i ][ j ] = t;

}

}

}

}

}

void FloydOld( ){

for( int k = 0 ; k < MAX ; ++k ){

for( int i = 0 ; i < MAX ; ++i ){

for( int j = 0 ; j < MAX ; ++j ){

int t = ady[ i ][ k ] + ady[ k ][ j ];

if( ady[ i ][ j ] > t ){

ady[ i ][ j ] = t;

}

}

}

}

}

int main(){

int E , d;

char road, dir, u , v;

while( scanf("%d" ,&E) && E ){

Init();

while( E-- ){

scanf("\n%c %c %c %c %d" , &road , &dir , &u , &v , &d );

if( road == 'Y' ){

if( d < young[ u - 'A' ][ v - 'A' ] )

young[ u - 'A' ][ v - 'A' ] = d;

if( dir == 'B' ){

if( d < young[ v -'A' ][ u-'A' ] )

young[ v -'A' ][ u-'A' ] = d;

}

}

else{

if( d < ady[ u - 'A' ][ v - 'A' ] )

ady[ u - 'A' ][ v - 'A' ] = d;

if( dir == 'B' ){

if( d < ady[ v-'A' ][ u-'A' ] )

ady[ v-'A' ][ u-'A' ] = d;

}

}

}

FloydYoung();

FloydOld();

scanf("\n%c %c" , &u , &v );

int min = INF;

for( int i = 0 ; i < MAX ;++i ){

int sum = ady[ v -'A' ] [ i ] + young[ u - 'A' ][ i ];

if( min > sum) min = sum;

}

if( min == INF )printf("You will never meet.\n");

else{

printf("%d",min);

for( int i = 0 ; i < MAX ; ++i){

if( ady[ v -'A' ] [ i ] + young[ u - 'A' ][ i ] == min ){

printf(" %c" , i + 'A');

}

}

printf("\n");

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Ocean Deep! Make it shallow!!

\*\*\*ID: 10176

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory, Modulo, bitwise

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MOD 131071

int main(){

int c, r;

for(r = 0; ( c = getchar() ) !=EOF; ){

if( c == '0' ){

r = ( r << 1 ) % MOD;

}

else if( c == '1' ){

r = ( ( r << 1 ) | 1 ) % MOD;

}

else if( c == '#'){

printf( r ? " NO\n" : "YES\n");

r = 0;

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Count the faces

\*\*\*ID: 10178

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Euler Formula

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 70

int uf[ MAX ];

int Find( int x ){

return ( uf[ x ] == x ) ? x : uf[ x ] = Find( uf[ x ] );

}

void Union( int x , int y ){

uf[ Find( x ) ] = Find( y );

}

char cities[ MAX ][ 2 ];

int numcities;

int get\_Id( char \*s ){

for( int i = 0 ; i < numcities ; ++i ){

if( strcmp( s , cities[ i ] ) == 0 ) return i;

}

strcpy( cities[ numcities ] ,s );

return numcities++;

}

int main(){

int V , E , components;

char u[ 2 ] , v[ 2 ];

while( scanf("%d %d" , & V , &E ) != EOF ){

components = 0;

numcities = 0;

for( int i = 0 ; i < V ; ++i ) uf[ i ] = i;

for( int i = 0 ; i < E ; ++i ){

scanf("%s %s" , &u , &v );

int x = get\_Id( u ), y = get\_Id( v );

Union( x , y );

}

for( int i = 0 ; i < V ; ++i ){

if( Find( i ) == i ){

components++;

}

}

printf("%d\n" , 1 + E - V + components );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Bee Maja

\*\*\*ID: 10182

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, math, brute force

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 100005

int main(){

int n , pos , aux , i , cnt , x , y;

while( scanf("%d" , &n ) != EOF ){

if( n == 1 ){

puts("0 0");

continue;

}

cnt = 1;

for( pos = 1 ; ; ++pos ){

cnt += 6 \* pos;

if( cnt >= n ) break;

}

cnt -= 6 \* pos;

aux = ( n - 1 - cnt )/pos;

if( aux == 0 ){

y = n - cnt;

x = pos - ( n - cnt );

}

else if( aux == 1 ){

cnt += pos;

x = -1 \* ( n - cnt );

y = pos;

}

else if( aux == 2 ){

cnt += aux \* pos;

x = -1 \* pos;

y = pos - ( n - cnt );

}

else if( aux == 3 ){

cnt += aux \* pos;

y = -1 \* ( n - cnt );

x = ( n - cnt ) - pos;

}

else if( aux == 4 ){

cnt += aux \* pos;

x = ( n - cnt );

y = -pos;

}

else{

cnt += aux \* pos;

x = pos;

y = ( n - cnt - pos);

}

printf("%d %d\n" , x , y );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Automated Judge Script

\*\*\*ID: 10188

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

char judge[ 101 ][ 122 ] , team[ 101 ][ 122 ];

int n1 , n2 , state , q = 1 , l1 , l2;

string d1 , d2;

while( scanf("%d" , &n1 ) , n1 ){

getchar();

for( int i = 0 ; i < n1 ; ++i ) gets( judge[ i ] );

scanf("%d" , &n2 );

getchar();

for( int i = 0 ; i < n2 ; ++i ) gets( team[ i ] );

state = 1;

d1 = d2 = "";

for( int i = 0 ; i < n1 ; ++i ){

l1 = strlen( judge[ i ] );

for( int j = 0 ; j < l1 ; ++j ){

if( '0' <=judge[ i ][ j ] && judge[ i ][ j ] <= '9')d1 += judge[ i ][ j ];

}

}

for( int i = 0 ; i < n2 ; ++i ){

l2 = strlen( team[ i ] );

for( int j = 0 ; j < l2 ; ++j ){

if( '0' <=team[ i ][ j ] && team[ i ][ j ] <= '9') d2 += team[ i ][ j ];

}

}

if( d1 != d2 ) state = 3;

if(state == 1 ){

if( n1 != n2 ){

state = 2;

goto finish;

}

for( int i = 0 ; i < n1 ; ++i ){

l2 = strlen( team[ i ] );

l1 = strlen( judge[ i ] );

if( l1 != l2 ){

state = 2;

break;

}

for( int j = 0 ; j < l1 ; ++j ){

if( team[ i ][ j ] != judge[ i ][ j ] ){

state = 2;

break;

}

}

if( state == 2 )break;

}

}

finish:

printf("Run #%d: " , q++ );

if( state == 1 ) puts("Accepted");

else if( state == 2 ) puts("Presentation Error");

else puts("Wrong Answer");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Check the Check

\*\*\*ID: 10196

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc Chess

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

char ady[ 10 ][ 10 ];

int cabx[ 8 ]={ -2 , -2 , -1 , -1 , 1 , 1 , 2 , 2 };

int caby[ 8 ]={ -1 , 1 , -2 , 2 , -2 , 2 , -1 , 1 };

int rx[ 4 ] = { 1 , -1 , 0 , 0 };

int ry[ 4 ] = { 0 , 0 , 1 , -1 };

int qx[ 8 ] = { 1 , -1 , 0 , 0 , 1 , 1 , -1 , -1 };

int qy[ 8 ] = { 0 , 0 , 1 , -1 , 1 , -1 , 1 , -1 };

int bx[ 4 ] = { 1 , 1 , -1 , -1 };

int by[ 4 ] = { 1 , -1 , 1 , -1 };

int main(){

bool f;

int C;

int c;

int q = 1, xw , yw , xb ,yb ,x , y;

while( 1 ){

for( int i = 0 ; i < 8 ; ++i ) gets( ady[ i ] );

f = false;

for( int i = 0 ; i < 8 ; ++i ){

for( int j = 0 ; j < 8 ; ++j ){

if( ady[ i ][ j ] != '.')f = true;

if( ady[ i ][ j ] == 'k' ) { xb = i; yb = j; }

if( ady[ i ][ j ] == 'K' ) { xw = i; yw = j; }

}

}

if( !f)break;

C = 'N';

//8 movimientos

//reina

for( int i = 0 ; i < 8 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = k \* qx[ i ] + xb, ny = k \* qy[ i ]+ yb;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == '.' )continue;

if( ady[ nx ][ ny ] == 'Q' ){

C = 'B';

break;

}

else break;

}

else break;

}

}

if( C == 'N'){

//caballo

for( int i = 0 ; i < 8 ; ++i ){

int nx = cabx[ i ] + xb, ny = caby[ i ]+ yb;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == 'N' ){

C = 'B';

break;

}

}

}

}

if( C == 'N'){

//bishop

for( int i = 0 ; i < 4 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = k \* bx[ i ] + xb, ny = k \* by[ i ]+ yb;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == '.')continue;

if( ady[ nx ][ ny ] == 'B' ){

C = 'B';

break;

}

else break;

}

else break;

}

}

}

if( C == 'N'){

//rook

for( int i = 0 ; i < 4 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = k \* rx[ i ] + xb, ny = k \* ry[ i ]+ yb;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == '.')continue;

if( ady[ nx ][ ny ] == 'R' ){

C = 'B';

break;

}

else break;

}

else break;

}

}

}

if( C == 'N' ){

//peon

for( int i = 0 ; i < 2 ; ++i ){

int nx = bx[ i ] + xb, ny = by[ i ]+ yb;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == 'P' ){

C = 'B';

break;

}

}

}

}

if( C == 'N' ){

//queen

for( int i = 0 ; i < 8 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = k \* qx[ i ] + xw, ny = k \* qy[ i ]+ yw;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == '.')continue;

if( ady[ nx ][ ny ] == 'q' ){

C = 'W';

break;

}

else break;

}

else break;

}

}

if( C == 'N' ){

//caballo

for( int i = 0 ; i < 8 ; ++i ){

int nx =cabx[ i ] + xw, ny = caby[ i ]+ yw;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == 'n' ){

C = 'W';

break;

}

}

}

}

if( C == 'N'){

//bishop

for( int i = 0 ; i < 4 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = k \* bx[ i ] + xw, ny = k \* by[ i ]+ yw;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == '.')continue;

if( ady[ nx ][ ny ] == 'b' ){

C = 'W';

break;

}

else break;

}

else break;

}

}

}

if( C == 'N'){

//rook

for( int i = 0 ; i < 4 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = k \* rx[ i ] + xw, ny = k \* ry[ i ]+ yw;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == '.')continue;

if( ady[ nx ][ ny ] == 'r' ){

C = 'W';

break;

}

else break;

}

else break;

}

}

}

if( C == 'N' ){

//peon

for( int i = 2 ; i < 4 ; ++i ){

int nx = bx[ i ] + xw, ny = by[ i ]+ yw;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 ){

if( ady[ nx ][ ny ] == 'p' ){

C = 'W';

break;

}

}

}

}

}

printf("Game #%d: " , q++ );

if( C == 'N')puts("no king is in check.");

else if( C == 'W') puts("white king is in check.");

else if( C == 'B' ) puts("black king is in check.");

gets( ady[ 0 ] );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Adventures in Moving

\*\*\*ID: 10201

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

#define MAXD 10005

#define INF 1<<30

int \_end , n , dist[ MAX ] , cost[ MAX ];

long long dp[ MAX ][ 205 ];

long long solve( int cur , int gas ){

if( gas < 0 ) return INF;

if( dist[ cur ] == \_end ){

if( dist[ cur ] + gas - \_end >= 100 ) return 0;

return INF;

}

int i;

if( dp[ cur ][ gas ] != -1 ) return dp[ cur ][ gas ];

long long ans = INF , costo;

int j, novoGas , k;

for( i = cur + 1 ; i < n ; ++i ){

//si es posible llegar

novoGas = dist[ i ] - dist[ cur ] - gas;

for( j = gas , k = 0 ; j <= 200 ; ++j , ++k ){

costo = k \* cost[ cur ];

if( j + dist[ cur ] >= dist[ i ] )

ans = min( ans , costo + solve( i , j - ( dist[ i ] - dist[ cur ] ) ) );

}

}

return dp[ cur ][ gas ] = ans;

}

char line[ 105 ];

int main(){

int t , mini , i , q;

scanf("%d" , &t );

long long ans;

bool correct;

for( q = 0 ; q < t ; ++q ){

if( q ) printf("\n");

while( gets( line ) != NULL && sscanf( line , "%d", &\_end ) != 1);

n = 0;

correct = false;

mini = 1<<30;

while( gets( line ) ){

if( strlen( line ) == 0 ) break;

sscanf( line , "%d %d" , &dist[ n ] , &cost[ n ] );

if( dist[ n ] > \_end ){

dist[ n + 1 ] = \_end;

cost[ n + 1 ] = INF;

swap( dist[ n ] , dist[ n + 1 ] );

swap( cost[ n ] , cost[ n + 1 ] );

n++;

}

n++;

}

if( dist[ n - 1 ] <= \_end ){

dist[ n ] = \_end;

cost[ n++ ] = INF;

}

memset( dp , -1 , sizeof( dp ) );

ans = INF;

for( i = 0 ; i < n - 1 ; ++i )

if( dist[ i ] <= 100 ) ans = min( ans , solve( i , 100 - dist[ i ] ) );

if( ans == INF )puts("Impossible");

else printf("%lld\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Snow Clearing

\*\*\*ID: 10203

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

double dist( int x1 , int y1 , int x2 , int y2 ){

return sqrt( ( x1 - x2 ) \* ( x1 - x2 ) + ( y1 - y2 ) \* ( y1 - y2 ) );

}

#define MAX 300

char line[ MAX ];

int main(){

int t ,x ,y ,x1 , y1 , x2 , y2 , time;

double d , D;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d %d" , &x , &y ) == 2 ){

d = 0.0;

gets( line );

while( gets( line ) && sscanf( line , "%d %d %d %d" , &x1 , &y1 , &x2 , &y2 ) == 4 ){

D = dist( x1 , y1 , x2 , y2 );

d += D/10000.0;

}

time = ( int )( d \* 60 + 0.5 ) ;

printf("%d:%02d\n" , time/60 , time % 60 );

if( t ) printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Stack 'em Up

\*\*\*ID: 10205

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc - Game, simulationa

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

string d[ 4 ] = {"Clubs", "Diamonds", "Hearts", "Spades"};

string aux[ 13 ] = {"2" , "3" , "4" , "5" , "6" , "7" , "8" , "9" , "10" , "Jack" , "Queen" , "King" , "Ace"};

int main(){

int t, shuffles, pos, shuffle[ 105 ][ 53 ];

scanf("%d" , &t );

vector<string> deck[ 105 ];

for( int i = 0; i < 4 ; ++i ){

for( int j = 0 ; j < 13 ; ++j ){

deck[ 0 ].push\_back( aux[ j ] + " of " + d[ i ] );

}

}

char kk[ 5 ];

int k;

for( int q = 0 ; q < t ; ++q ){

if( q )printf("\n");

scanf("%d" , &shuffles );

for( int sh = 0 ; sh < shuffles ; ++sh ){

deck[ sh + 1 ] = vector<string>( 53 );

for( int i = 0 ; i < 52 ; ++i ){

scanf("%d" , &shuffle[ sh + 1 ][ i ] );

}

}

gets( kk );

int sh = 0;

while( gets( kk ) ){

if( strlen( kk ) == 0 )break;

k = atoi( kk );

for( int i = 0 ; i < 52 ; ++i )

deck[ sh + 1 ][ i ] = deck[ sh ][ shuffle[ k ][ i ] - 1 ];

sh++;

}

for( int i = 0 ; i < 52 ; ++i )

printf("%s\n" , deck[ sh ][ i ].c\_str() );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Is This Integration ?

\*\*\*ID: 10209

\*\*\*Juez: UVA

\*\*\*Tipo: Geometry, sectors, angles

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cmath>

#define EPS 1e-9

const long double PI = 2 \* acos(.0);

int main(){

double r;

while( scanf("%lf" , &r ) != EOF ){

double z = r \* r \* ( 1 - PI/6.0 - sqrt( 3.0 )/4.0 );

double y = r \* r \* ( 1 - PI/4.0 ) - 2 \* z;

double x = r \* r - 4 \* z - 4 \* y;

printf("%.3lf %.3lf %.3lf\n" , x , 4 \* y , 4 \* z );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Last Non-zero Digit

\*\*\*ID: 10212

\*\*\*Juez: UVA

\*\*\*Tipo: Prime Factorization

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

long long n , m , i , dos , cinco , aux , MOD , ceros , ans;

while( scanf("%lld %lld" , &n , &m ) != EOF ){

dos = cinco = 0;

m = n - m + 1;

ans = 1;

for( i = m ; i <= n ; ++i ){

aux = i;

while( aux % 2 == 0 ){

dos++;

aux /= 2;

}

while( aux % 5 == 0 ){

cinco++;

aux /= 5;

}

ans = ( ans % 10 \* aux % 10 ) % 10;

}

if( dos < cinco ) for( i = dos ; i < cinco ; ++i ) ans = ( ans % 10 \* 5 ) %10 ;

else for( i = cinco ; i < dos ; ++i ) ans = ( ans % 10 \* 2 ) %10 ;

printf("%lld\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Satellites

\*\*\*ID: 10221

\*\*\*Juez: UVA

\*\*\*Tipo: Geometry

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cmath>

#define MAX 4

#define EPS 1e-9

const double PI = 2 \* acos(.0);

char s[ MAX ];

double Deg\_to\_Rad( double angle ){

return angle \* PI/180.0;

}

int main(){

double radius , angle;

while( scanf("%lf %lf %s" , &radius , &angle , s ) != EOF ){

radius += 6440;

if( s[ 0 ] == 'm' )

angle \*= 1/60.0;

if( angle > 180 )

angle = 360.0 - angle;

printf("%.6lf %.6lf\n" , radius \* PI \* 2.0 \* angle/360.0 ,

sqrt( 2 \* radius \* radius \* ( 1 - cos( Deg\_to\_Rad(angle) ) ) ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Hardwood Species

\*\*\*ID: 10226

\*\*\*Juez: UVA

\*\*\*Tipo: Data Structure, Trie

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t,tot;

scanf("%d",&t);

char line[40];

gets(line);

gets(line);

map<string,int>::iterator it;

for(int q = 0; q<t; q++){

if(q)putchar('\n');

map<string,int> mp;

tot=0;

while(gets(line)){

if(strcmp(line,"") == 0)break;

mp[string(line)]++;

++tot;

}

for(it = mp.begin();it!= mp.end(); it++){

printf("%s %.4lf\n",(\*it).first.c\_str(),((\*it).second/(double)tot)\*100.0);

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Forest

\*\*\*ID: 10227

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

vector<int> ady[ MAX ];

int main(){

int t , p , q, x , y, ans;

scanf("%d" , &q );

char line[256];

bool visit[ MAX ][ MAX ];

int aux[ MAX ], cnt[ MAX ], seen[ MAX ];

for( int qq = 0 ; qq < q ; ++qq ){

if( qq )printf("\n");

scanf("%d %d" , &p , &t );

memset( visit , 0 , sizeof( visit ) );

gets( line );

while( gets( line ) ){

int l = strlen( line );

if( l == 0 )break;

sscanf( line , "%d %d" , &x , &y );

if( !visit[ x ][ y ] )ady[ x ].push\_back( y );

visit[ x ][ y ] = 1;

}

ans = 0;

memset( seen , 0 , sizeof( seen ) );

for( int i = 1 ; i <= p ; ++i ){

if( seen[ i ] != 0 || ady[ i ].size() == 0 ) continue;

memset( cnt , 0 , sizeof( cnt ) );

for( int j = 0 ; j < ady[ i ].size() ; ++j ) cnt[ ady[ i ][ j ] ]++;

seen[ i ] = i;

for( int j = i + 1 ; j <= p ; ++j ){

if( seen[ j ] != 0 || ady[ j].size() == 0 ) continue;

memset( aux , 0 , sizeof( aux ) );

for( int k = 0 ; k < ady[ j ].size() ; ++k ) aux[ ady[ j ][ k ] ]++;

int k = 0;

for( ; k < MAX ; ++k ){

if( cnt[ k ] != aux[ k ] ){

break;

}

}

if( k == MAX ){

seen[ j ] = i;

}

}

}

set<int> s;

for( int i = 1 ; i <= p ; ++i ){

s.insert( seen[ i ] );

}

printf("%d\n" , s.size() );

for( int i = 0 ; i < MAX ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Throw the Dice

\*\*\*ID: 10238

\*\*\*Juez: UVA

\*\*\*Tipo: Probability Basic + DP + Bignum

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

const int MAXD = 105, DIG = 9, BASE = 1000000000;

const unsigned long long BOUND = numeric\_limits <unsigned long long> :: max () - (unsigned long long) BASE \* BASE;

struct bignum

{

int D, digits [MAXD / DIG + 2];

inline void trim ()

{

while (D > 1 && digits [D - 1] == 0)

D--;

}

inline void init (long long x)

{

memset (digits, 0, sizeof (digits));

D = 0;

do

{

digits [D++] = x % BASE;

x /= BASE;

}

while (x > 0);

}

inline bignum (long long x)

{

init (x);

}

inline bignum (int x = 0)

{

init (x);

}

inline bignum (char \*s)

{

memset (digits, 0, sizeof (digits));

int len = strlen (s), first = (len + DIG - 1) % DIG + 1;

D = (len + DIG - 1) / DIG;

for (int i = 0; i < first; i++)

digits [D - 1] = digits [D - 1] \* 10 + s [i] - '0';

for (int i = first, d = D - 2; i < len; i += DIG, d--)

for (int j = i; j < i + DIG; j++)

digits [d] = digits [d] \* 10 + s [j] - '0';

trim ();

}

inline char \*str ()

{

trim ();

char \*buf = new char [DIG \* D + 1];

int pos = 0, d = digits [D - 1];

do

{

buf [pos++] = d % 10 + '0';

d /= 10;

}

while (d > 0);

reverse (buf, buf + pos);

for (int i = D - 2; i >= 0; i--, pos += DIG)

for (int j = DIG - 1, t = digits [i]; j >= 0; j--)

{

buf [pos + j] = t % 10 + '0';

t /= 10;

}

buf [pos] = '\0';

return buf;

}

inline bool operator < (const bignum &o) const

{

if (D != o.D)

return D < o.D;

for (int i = D - 1; i >= 0; i--)

if (digits [i] != o.digits [i])

return digits [i] < o.digits [i];

return false;

}

inline bool operator == (const bignum &o) const

{

if (D != o.D)

return false;

for (int i = 0; i < D; i++)

if (digits [i] != o.digits [i])

return false;

return true;

}

inline bignum operator << (int p) const

{

bignum temp;

temp.D = D + p;

for (int i = 0; i < D; i++)

temp.digits [i + p] = digits [i];

for (int i = 0; i < p; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum operator >> (int p) const

{

bignum temp;

temp.D = D - p;

for (int i = 0; i < D - p; i++)

temp.digits [i] = digits [i + p];

for (int i = D - p; i < D; i++)

temp.digits [i] = 0;

return temp;

}

inline bignum range (int a, int b) const

{

bignum temp = 0;

temp.D = b - a;

for (int i = 0; i < temp.D; i++)

temp.digits [i] = digits [i + a];

return temp;

}

inline bignum operator + (const bignum &o) const

{

bignum sum = o;

int carry = 0;

for (sum.D = 0; sum.D < D || carry > 0; sum.D++)

{

sum.digits [sum.D] += (sum.D < D ? digits [sum.D] : 0) + carry;

if (sum.digits [sum.D] >= BASE)

{

sum.digits [sum.D] -= BASE;

carry = 1;

}

else

carry = 0;

}

sum.D = max (sum.D, o.D);

sum.trim ();

return sum;

}

inline bignum operator - (const bignum &o) const

{

bignum diff = \*this;

for (int i = 0, carry = 0; i < o.D || carry > 0; i++)

{

diff.digits [i] -= (i < o.D ? o.digits [i] : 0) + carry;

if (diff.digits [i] < 0)

{

diff.digits [i] += BASE;

carry = 1;

}

else

carry = 0;

}

diff.trim ();

return diff;

}

inline bignum operator \* (const bignum &o) const

{

bignum prod = 0;

unsigned long long sum = 0, carry = 0;

for (prod.D = 0; prod.D < D + o.D - 1 || carry > 0; prod.D++)

{

sum = carry % BASE;

carry /= BASE;

for (int j = max (prod.D - o.D + 1, 0); j <= min (D - 1, prod.D); j++)

{

sum += (unsigned long long) digits [j] \* o.digits [prod.D - j];

if (sum >= BOUND)

{

carry += sum / BASE;

sum %= BASE;

}

}

carry += sum / BASE;

prod.digits [prod.D] = sum % BASE;

}

prod.trim ();

return prod;

}

inline double double\_div (const bignum &o) const

{

double val = 0, oval = 0;

int num = 0, onum = 0;

for (int i = D - 1; i >= max (D - 3, 0); i--, num++)

val = val \* BASE + digits [i];

for (int i = o.D - 1; i >= max (o.D - 3, 0); i--, onum++)

oval = oval \* BASE + o.digits [i];

return val / oval \* (D - num > o.D - onum ? BASE : 1);

}

inline pair <bignum, bignum> divmod (const bignum &o) const

{

bignum quot = 0, rem = \*this, temp;

for (int i = D - o.D; i >= 0; i--)

{

temp = rem.range (i, rem.D);

int div = (int) temp.double\_div (o);

bignum mult = o \* div;

while (div > 0 && temp < mult)

{

mult = mult - o;

div--;

}

while (div + 1 < BASE && !(temp < mult + o))

{

mult = mult + o;

div++;

}

rem = rem - (o \* div << i);

if (div > 0)

{

quot.digits [i] = div;

quot.D = max (quot.D, i + 1);

}

}

quot.trim ();

rem.trim ();

return make\_pair (quot, rem);

}

inline bignum operator / (const bignum &o) const

{

return divmod (o).first;

}

inline bignum operator % (const bignum &o) const

{

return divmod (o).second;

}

inline bignum power (int exp) const

{

bignum p = 1, temp = \*this;

while (exp > 0)

{

if (exp & 1) p = p \* temp;

if (exp > 1) temp = temp \* temp;

exp >>= 1;

}

return p;

}

};

#define MAX 55

bignum dp[ MAX ][ 4005 ];

bool seen[ MAX ][ 4005 ];

int n , face;

bignum solve( int len , int sum ){

if( len == n ){

if( sum == 0 ) return 1;

return 0;

}

if( sum < 0 ) return 0;

if( seen[ len ][ sum ] ) return dp[ len ][ sum ];

bignum ans = 0;

for( int i = 1 ; i <= face ; ++i ){

ans = ans + solve( len + 1 , sum - i );

}

seen[ len ][ sum ] = 1;

return dp[ len ][ sum ] = ans;

}

int main(){

int k , i;

while( scanf("%d %d %d" , &face , &n , &k ) != EOF ){

bignum denom = 1, num;

memset( seen , 0 , sizeof( seen ) );

for( i = 0 ; i < n ; ++i ){

denom = denom \* face;

}

printf("%s/%s\n" , solve( 0 , k ).str() , denom.str() );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Fire! Fire!! Fire!!!

\*\*\*ID: 10243

\*\*\*Juez: UVA

\*\*\*Tipo: DP Minimum Vertex Cover

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#include <vector>

using namespace std;

#define MAX 1005

#define INF 1<<30

int V , dp[ MAX ][ 2 ];

vector< int > ady[ MAX ];

int min( int a , int b ){ return ( a < b )? a : b; }

int solve( int cur , int color , int parent ){

if( dp[ cur ][ color ] != -1 ) return dp[ cur ][ color ];

int ans = INF , i , y;

if( color ){

ans = 1;

for( i = 0 ; i < ady[ cur ].size() ; ++i ){

y = ady[ cur ][ i ];

if( y != parent )

ans += min( solve( y , 0 , cur ) , solve( y , 1 , cur ) );

}

}

else{

ans = 0;

for( i = 0 ; i < ady[ cur ].size() ; ++i ){

y = ady[ cur ][ i ];

if( y != parent )

ans += solve( y , 1 , cur );

}

}

return dp[ cur ][ color ] = ans;

}

int main(){

int i , k , v , ans;

while( scanf("%d" , &V ) , V ){

for( i = 0 ; i < V && scanf("%d" , &k ) == 1 ; ++i ){

while( k-- ){

scanf("%d" , &v );

ady[ i ].push\_back( v - 1 );

}

}

memset( dp , -1 , sizeof( dp ) );

ans = solve( 0 , 1 , -1 );

if( ans > 1 ) ans = min( ans , solve( 0 , 0 , -1 ) );

printf( "%d\n" , ans );

for( i = 0 ; i < V ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Grand Dinner

\*\*\*ID: 10249

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

struct Team{

int sz , id;

Team( int s , int i ): sz( s ) , id( i ){}

Team(){}

bool operator <( const Team t1 ) const{

return sz > t1.sz || ( sz == t1.sz && id < t1.id );

}

}team[ MAX ] , table[ MAX ];

vector<int> ady[ MAX ];

int T[ MAX ];

int main(){

int m , n, alu , mesa;

bool first;

while( scanf("%d %d" , &m , &n ) , m | n ){

for( int i = 0 ; i < MAX ; ++i ) ady[ i ].clear();

for( int i = 0 ; i < m ; ++i ){

scanf("%d" , &team[ i ].sz );

team[ i ].id = i + 1;

T[ i ] = team[ i ].sz;

}

for( int i = 0 ; i < n ; ++i ){

scanf("%d" , &table[ i ].sz );

table[ i ].id = i + 1;

}

sort( team , team + m );

for( int i = 0 ; i < m ; ++i ){

for( int j = 0 , k = team[ i ].sz ; j < n && k > 0 ; ++j ){

if( table[ j ].sz > 0 ){

table[ j ].sz--;

k--;

ady[ team[ i ].id ].push\_back( table[ j ].id );

}

}

}

int i = 0;

for( ; i < m ; ++i ){

if( ady[ i + 1 ].size() != T[ i ] ){

puts("0");

break;

}

}

if( i != m ){

continue;

}

puts("1");

for( i = 1 ; i <= m ; ++i ){

for( int j = 0 ; j < ady[ i ].size() ; ++j ){

if( j ) printf(" ");

printf("%d" , ady[ i ][ j ] );

}

printf("\n");

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Other Two Trees

\*\*\*ID: 10250

\*\*\*Juez: UVA

\*\*\*Tipo: Point Rotation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define EPS 1e-9

const long double PI = 2 \* acos(.0);

struct Point{

double x , y;

Point( double X , double Y ):x(X) , y(Y){}

bool operator <( Point other ) const{

if( fabs( x - other.x ) < EPS ){

return y + EPS < other.y;

}

return x + EPS < other.x;

}

bool operator ==( Point other )const{

return ( fabs( x - other.x ) < EPS && fabs( y - other.y ) < EPS );

}

Point operator -(Point other)const{

return Point( x - other.x , y - other.y );

}

Point(){}

};

//Distance between two points

double dist( Point p1 , Point p2 ){

return hypot( p1.x - p2.x , p1.y - p2.y );

}

//Convert angle from degrees to radians

double Deg\_to\_Rad( double angle ){

return angle \* PI/180.0;

}

//Convert angle from radians to degrees

double Rad\_to\_Deg( double angle ){

return angle \* 180.0/PI;

}

//Rotate ccw

Point rotate( Point p , double angle ){

double rad = Deg\_to\_Rad( angle );

return Point( p.x \* cos( rad ) - p.y \* sin( rad ) , p.x \* sin( rad ) + p.y \* cos( rad ) );

}

int main(){

double x1 ,y1 , x2 , y2;

while( scanf("%lf %lf %lf %lf" , &x1 , &y1 , &x2 , &y2 ) != EOF ){

if( fabs( x1 ) > EPS || fabs( y1 ) > EPS || fabs( x2 ) > EPS || fabs( y2 ) > EPS ){

Point mid( (x1 + x2)/2 , (y1+y2)/2) ;

Point A = rotate( Point( x1 - mid.x , y1 - mid.y ) , 90 ),

B = rotate( Point( x2 - mid.x , y2 - mid.y ) , 90 );

printf("%.10lf %.10lf %.10lf %.10lf\n" , A.x + mid.x , A.y + mid.y , B.x + mid.x , B.y + mid.y );

}

else break;

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Common Permutation

\*\*\*ID: 10252

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

using namespace std;

int main(){

char s1[4005], s2[4005];

int v;

while( gets(s1) && gets(s2)){

int a[26] = {0}, b[26] = {0};

for( int i = 0 ; s1[i] ; ++i ){

if( s1[i] >= 'a' && s1[i] <= 'z') a[ s1[i] - 'a']++;

}

for( int i = 0 ; s2[i] ; ++i ){

if( s2[i] >= 'a' && s2[i] <= 'z') b[ s2[i] - 'a']++;

}

for(int i = 0 ; i < 26 ; ++i ){

v = ( a[i] < b[i] ) ? a[i] : b[i];

while( v-- ) putchar(i+'a');

}

putchar('\n');

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Adventure of Super Mario

\*\*\*ID: 10269

\*\*\*Juez: UVA

\*\*\*Tipo: Dijkstra + Floyd

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstdio>

#include <cstring>

#include <queue>

using namespace std;

#define INF 1<<30

#define MAX 200

#define MAXL 105

#define MAXK 14

int A , B , L , K , dist[ MAX ][ MAXK ];

int ady[ MAX ][ MAX ];

void init(){

int i , j;

for( i = 0 ; i <= A + B ; ++i ){

for( j = 0 ; j <= A + B ; ++j ){

( i == j ) ? ady[ i ][ j ] = 0 : ady[ i ][ j ] = INF;

}

}

}

void Floyd(){

int i , j , k;

for( k = 0 ; k < A ; ++k ){

for( i = 0 ; i < A + B ; ++i ){

if( ady[ i ][ k ] == INF ) continue;

for( j = 0 ; j < A + B ; ++j ){

if( ady[ k ][ j ] != INF && ady[ i ][ k ] + ady[ k ][ j ] < ady[ i ][ j ] ){

ady[ i ][ j ] = ady[ i ][ k ] + ady[ k ][ j ];

}

}

}

}

}

struct Node{

int x , d , l , k;

Node( int xx , int dd , int ll , int kk): x( xx ) , d( dd ) , l( ll ), k( kk ){}

};

struct cmp{

bool operator() ( Node e1 , Node e2 ){

return e1.d > e2.d;

}

};

int dijkstra( int inicial ){

int i , v , w , l , k , u , j , c;

priority\_queue< Node , vector< Node > , cmp > Q;

for( i = 0 ; i <= A + B ; ++i )

for( k = 0 ; k <= K ; ++k )

dist[ i ][ k ] = INF;

for( k = 0 ; k <= K ; ++k ) dist[ inicial ][ k ] = 0;

Q.push( Node( inicial , 0 , L , K ) );

while( !Q.empty() ){

Node act = Q.top(); Q.pop();

u = act.x;

for( v = 0 ; v < A + B ; ++v ){

w = ady[ u ][ v ];

l = act.l;

k = act.k;

if( ady[ u ][ v ] != INF ){

if( dist[ v ][ k ] > dist[ u ][ k ] + w ){

dist[ v ][ k ] = dist[ u ][ k ] + w;

Q.push( Node( v , dist[ v ][ k ] , l , k ) );

}

if( k > 0 && l >= w && dist[ v ][ k - 1 ] > dist[ u ][ k ] ){

dist[ v ][ k - 1 ] = dist[ u ][ k ];

Q.push( Node( v , dist[ v ][ k - 1 ] , l , k - 1 ) );

}

}

}

}

int ans = INF;

for( k = 0 ; k <= K ; ++k ) ans = min( ans , dist[ 0 ][ k ] );

return ans;

}

int main(){

int t , E , u , v , w , i;

scanf( "%d" , &t );

while( t-- > 0 && scanf("%d %d %d %d %d" , &A , &B , &E , &L , &K ) ){

init();

while( E-- > 0 && scanf("%d %d %d" , &u , &v , &w ) ){

ady[ u - 1 ][ v - 1 ] = ady[ v - 1 ][ u - 1 ] = w;

}

Floyd();

printf("%d\n" , dijkstra( A + B - 1 ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Hanoi Tower Troubles Again!

\*\*\*ID: 10276

\*\*\*Juez: UVA

\*\*\*Tipo: Brute force, Backtracking

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#include <cmath>

#define MAX 51

int a[ MAX ];

bool isInt( double n ){

if( (int) n == n ){

return true;

}

return false;

}

int main(){

int i, j, n, t;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d", &n) == 1 ) {

memset( a , 0 , sizeof( a ) );

for( i = 1 ; ; ++i ) {

for( j = 0 ; j < n ; ++j )

if( a[ j ] == 0 || isInt( sqrt( a[ j ] + i ) ) ) break;

if ( j < n ) a[ j ] = i;

else break;

}

printf("%d\n", i - 1);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Boastin' Red Socks

\*\*\*ID: 10277

\*\*\*Juez: UVA

\*\*\*Tipo: Probability + Binary Search

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

//probabilidad condicional

long long num( long long R ){

return R \* ( R - 1 );

}

long long denom( long long R , long long A ){

return ( R + A ) \* ( R + A - 1 );

}

long long gcd( long long a , long long b ){

if( b == 0 )return a;

return gcd( b , a % b );

}

int main(){

int i;

long long p , q , left , right , mid , A , R , n , d , gc;

while( scanf("%lld %lld" , &p , &q ) , q | p ){

if( p == q ){ puts("2 0"); continue; }

if( p == 0 ){ puts("0 2"); continue; }

gc = gcd( p , q );

p /= gc; q /= gc;

R = A = -1;

for( i = 2 ; i <= 50000 ; ++i ){

left = 1; right = i;

A = -1;

while( left < right ){

mid = ( left + right )>>1;

n = num( i - mid ); d = denom( i - mid , mid );

if( n \* q == p \* d ){

A = mid;

R = i - mid;

}

if( n \* q <= p \* d ){

right = mid;

}

else left = mid + 1;

}

if( A != -1 ) break;

}

if( A == -1 ) puts("impossible");

else printf("%lld %lld\n" , R , A );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# BabelFish

\*\*\*ID: 10282

\*\*\*Juez: UVA

\*\*\*Tipo: Trie, map

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstdio>

#include <vector>

#include <string>

#include <cstring>

using namespace std;

const int ALPH\_SIZE = 26;

struct Node{

vector<Node\*> links;

bool used;

string val;

Node();

};

Node::Node(){

links.resize(ALPH\_SIZE,NULL);

used = false;

};

class Trie{

public :

Trie();

void printValue(const string &s) const;

void insert(const string &s, const string &val);

private :

Node\* myRoot;

};

Trie::Trie(){

myRoot = new Node();

}

void Trie::printValue(const string &s) const{

Node \*t = myRoot;

int len = s.size();

for(int i=0;i<len;++i){

if(t->links[s[i]-'a']==NULL){

t = NULL;

break;

}

t = t->links[s[i]-'a'];

}

if(t==NULL || (t->used)==false) fputs("eh\n",stdout);

else{

fputs((t->val).c\_str(),stdout);

putc('\n',stdout);

}

}

void Trie::insert(const string &s, const string &val){

Node \*t = myRoot;

int len = s.size();

for(int i=0;i<len;++i){

if(t->links[s[i]-'a']==NULL) t->links[s[i]-'a'] = new Node();

t = t->links[s[i]-'a'];

}

t->val = val;

t->used = true;

}

int main(){

char line[22],s1[11],s2[11];

string key,val;

Trie \*myTrie;

myTrie = new Trie();

while(true){

fgets(line,3300001,stdin);

if(strlen(line)==1) break;

sscanf(line,"%s%s",s1,s2);

key = string(s2);

val = string(s1);

myTrie->insert(key,val);

}

while(scanf("%s",s1)==1){

key = string(s1);

myTrie->printValue(key);

}

delete myTrie;

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Kissing Circles

\*\*\*ID: 10283

\*\*\*Juez: UVA

\*\*\*Tipo: Geometry, Circles

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cmath>

const double PI = 2 \* acos(.0);

//Convert angle from degrees to radians

double Deg\_to\_Rad( double angle ){

return angle \* PI/180.0;

}

int main(){

double R , N , angle , r , I , E , l , angle\_mini\_circle;

while( scanf("%lf %lf" , &R , &N ) != EOF ){

if( N == 1 ){

r = R; I = E = 0;

}

else{

angle = 360/N;

r = R \* sin( Deg\_to\_Rad(angle/2.0) )/( 1.0 + sin(Deg\_to\_Rad(angle/2.0)) );

l = sqrt( ( R - r ) \* ( R - r ) - r \* r);

angle\_mini\_circle = 90.0 - angle/2.0;

double area\_sector = PI \* r \* r \* angle\_mini\_circle/360.0;

double area\_triangle = r \* l/2.0;

I = 2.0 \* N \* ( area\_triangle - area\_sector );

E = PI \* R \* R - N \* PI \* r \* r - I;

}

printf("%.10lf %.10lf %.10lf\n" , r , I , E );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Chessboard in FEN

\*\*\*ID: 10284

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 105

char s[ MAX ];

int ady[ 8 ][ 8 ];

int n = 8;

int cabx[ 8 ]={ -2 , -2 , -1 , -1 , 1 , 1 , 2 , 2 };

int caby[ 8 ]={ -1 , 1 , -2 , 2 , -2 , 2 , -1 , 1 };

int rx[ 4 ] = { 1 , -1 , 0 , 0 };

int ry[ 4 ] = { 0 , 0 , 1 , -1 };

int qx[ 8 ] = { 1 , -1 , 0 , 0 , 1 , 1 , -1 , -1 };

int qy[ 8 ] = { 0 , 0 , 1 , -1 , 1 , -1 , 1 , -1 };

int bx[ 4 ] = { 1 , 1 , -1 , -1 };

int by[ 4 ] = { 1 , -1 , 1 , -1 };

void queen( int x , int y ){

for( int i = 0 ; i < 8 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = k \* qx[ i ] + x, ny = k \* qy[ i ]+ y;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 && ( ady[ nx ][ ny ] == '.' || ady[ nx ][ ny ] == 1 ) ){

ady[ nx ][ ny ] = 1;

}

else break;

}

}

}

void knight( int x , int y ){

for( int i = 0 ; i < 8 ; ++i ){

int nx = cabx[ i ] + x, ny = caby[ i ]+ y;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 && ady[ nx ][ ny ] == '.' ){

ady[ nx ][ ny ] = 1;

}

}

}

void bishop( int x , int y ){

for( int i = 0 ; i < 4 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = k \* bx[ i ] + x, ny = k \* by[ i ]+ y;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 && ( ady[ nx ][ ny ] == '.' || ady[ nx ][ ny ] == 1 )){

ady[ nx ][ ny ] = 1;

}

else break;

}

}

}

void rook( int x , int y ){

for( int i = 0 ; i < 4 ; ++i ){

int k = 0;

while( 1 ){

k++;

int nx = k \* rx[ i ] + x, ny = k \* ry[ i ]+ y;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 && ( ady[ nx ][ ny ] == '.' || ady[ nx ][ ny ] == 1 )){

ady[ nx ][ ny ] = 1;

}

else break;

}

}

}

void peon( int x , int y , bool color ){

if( color ){ // si negro

for( int i = 0 ; i < 2 ; ++i ){

int nx = bx[ i ] + x, ny = by[ i ]+ y;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 && ady[ nx ][ ny ] == '.' ){

ady[ nx ][ ny ] = 1;

}

}

}

else{

for( int i = 2 ; i < 4 ; ++i ){

int nx = bx[ i ] + x, ny = by[ i ]+ y;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 && ady[ nx ][ ny ] == '.' ){

ady[ nx ][ ny ] = 1;

}

}

}

}

void king( int x , int y ){

for( int i = 0 ; i < 8 ; ++i ){

int nx = qx[ i ] + x, ny = qy[ i ] + y;

if( nx >= 0 && ny >= 0 && nx < 8 && ny < 8 && ady[ nx ][ ny ] == '.' ){

ady[ nx ][ ny ] = 1;

}

}

}

int main(){

int l, fila , col;

while( scanf("%s" , &s ) != EOF ){

l = strlen( s );

fila = col = 0;

for( int i = 0 ; i < l ; ++i ){

if( s[ i ] == '/' ){

col = 0;

fila++;

continue;

}

if( '0' <= s[ i ] && s[ i ] <= '8' ){

for( int j = 0 ; j < ( s[ i ] - '0' ) ; ++j ) ady[ fila ][ col++ ] = '.';

}

else ady[ fila ][ col++ ] = s[ i ];

}

for( int i = 0 ; i < 8 ; ++i ){

for( int j = 0 ; j < 8 ; ++ j ){

if( ady[ i ][ j ] == 'k' || ady[ i ][ j ] == 'K' )king( i , j );

else if( ady[ i ][ j ] == 'n' || ady[ i ][ j ] == 'N' )knight( i , j );

else if( ady[ i ][ j ] == 'p' )peon( i , j , 1 );

else if( ady[ i ][ j ] == 'P' )peon( i , j , 0 );

else if( ady[ i ][ j ] == 'r' || ady[ i ][ j ] == 'R' )rook( i , j );

else if( ady[ i ][ j ] == 'q' || ady[ i ][ j ] == 'Q' )queen( i , j );

else if( ady[ i ][ j ] == 'b' || ady[ i ][ j ] == 'B' )bishop( i , j );

}

}

int ans = 0;

for( int i = 0 ; i < 8 ; ++i ){

for( int j = 0 ; j < 8 ; ++ j ){

if( ady[ i ][ j ] == '.')ans++;

}

}

printf("%d\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Longest Run on a Snowboard

\*\*\*ID: 10285

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking, DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 105

int max( int a , int b ){ return ( a > b )? a : b; }

int r , c , ady[ MAX ][ MAX ] , maxi , dx[ 4 ] = { 0 , 0 , 1 , -1 } , dy[ 4 ] = { 1 , -1 , 0 , 0 } , dist;

bool seen[ MAX ][ MAX ];

int dp[ MAX ][ MAX ];

void dfs( int x , int y ){

dist++;

maxi = max( maxi , dist );

seen[ x ][ y ] = 1;

for( int i = 0 ; i < 4 ; ++i ){

int nx = x + dx[ i ], ny = y + dy[ i ];

if( nx >= 0 && ny >= 0 && nx < r && ny < c && seen[ nx ][ ny ] == 0 && ady[ nx ][ ny ] < ady[ x ][ y ] ){

dfs( nx , ny );

dist--;

seen[ nx ][ ny ] = 0;

}

}

}

int main(){

int t;

char s[ 205 ];

scanf("%d" , &t );

while( t-- ){

scanf("%s %d %d" , &s , &r , &c );

for( int i = 0 ; i < r ; ++i ){

for( int j = 0 ; j < c ; ++j ){

scanf("%d" , &ady[ i ][ j ] );

}

}

maxi = 0;

dist = 0;

for( int i = 0 ; i < r ; ++i ){

for( int j = 0 ; j < c ; ++j ){

memset( seen , 0 , sizeof( seen ) );

dist = 0;

dfs( i , j );

}

}

printf("%s: %d\n" , s , maxi );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Hay Points

\*\*\*ID: 10295

\*\*\*Juez: UVA

\*\*\*Tipo: Map

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int m ,n , val , l;

char s[ 1005 ];

map<string , int > mp;

scanf("%d %d" , &m , &n );

while( m-- ){

scanf("%s %d" , &s , &val );

mp[ s ] = val;

}

val = 0;

bool point;

while( n && gets( s ) ){

l = strlen( s );

point = false;

if( s[ l - 1 ] == '.') point = true;

s[ l++ ] = ' ';

s[ l ] = '\0';

char \*p = strtok( s , " ");

while( p != NULL ){

val += mp[ p];

p = strtok( NULL , " ");

}

if( point ){ n--; printf("%d\n" , val ); val = 0;}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Jogging Trails

\*\*\*ID: 10296

\*\*\*Juez: UVA

\*\*\*Tipo: DP Bitmask , Shortest Path , Euler Cycle

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 16

#define INF 1<<30

int min( int a , int b ){ return ( a < b )?a : b ;}

int degree[ MAX ] , V , dist[ MAX ] , prev[ MAX ] , seen[ MAX ];

int ady[ MAX ][ MAX ];

void init(){

int i ,j;

for( i = 0 ; i < V ; ++i )

for( j = 0 ; j < V ; ++j )

ady[ i ][ j ] = ( i == j )?0:INF;

}

void floyd(){

int i , j , k;

for( k = 0 ; k < V ; ++k ){

for( i = 0 ; i < V ; ++i ){

if( ady[ i ][ k ] == INF ) continue;

for( j = 0 ; j < V ; ++j ){

if( ady[ k ][ j ] != INF && ady[ i ][ j ] > ady[ i ][ k ] + ady[ k ][ j ] ){

ady[ i ][ j ] = ady[ i ][ k ] + ady[ k ][ j ];

}

}

}

}

}

int impar[ MAX ] , len;

int bitcount( int n ){

int cnt = 0;

while( n ){

n = n & ( n - 1 );

cnt++;

}

return cnt;

}

int dp[ 1<<16 ];

int solve( int mask ){

if( bitcount( mask ) == len ) return 0;

if( dp[ mask ] != -1 ) return dp[ mask ];

int mini = INF , i , j;

for( i = 0 ; i < len ; ++i ){

if( mask & 1<<i ) continue;

for( j = i + 1 ; j < len ; ++j ){

if( mask & 1<<j ) continue;

mini = min( mini , ady[ impar[ i ] ][ impar[ j ] ] + solve( mask | 1<<i | 1<<j ) );

}

}

return dp[ mask ] = mini;

}

int main(){

int t , q , ans , E , i , j , u , v , w , DP;

while( scanf("%d" , &V ) , V ){

scanf("%d" , &E );

memset( degree , 0 , sizeof( degree ) );

init();

ans = 0;

while( E-- > 0 && scanf("%d %d %d" , &u , &v , &w ) ){

ady[ u - 1 ][ v - 1 ] = ady[ v - 1 ][ u - 1 ] = min( ady[ u - 1 ][ v - 1 ] , w );

ans += w;

degree[ u - 1 ]++;

degree[ v - 1 ]++;

}

//Floyd para rutas mas cortas

floyd();

len = 0;

DP = 0;

memset( dp , -1 , sizeof( dp ) );

for( i = 0 ; i < V ; ++i ){

if( degree[ i ] & 1 ){

//Solo es necesario impares para crear nodos falsos

impar[ len++ ] = i;

}

}

//Si todos son de ciclo par esa es la respuesta

//Sino hallo la minima combinacion de crear aristas falsas

if( len > 0 ) DP = solve( 0 );

printf("%d\n" , ans + DP );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Power Strings

\*\*\*ID: 10298

\*\*\*Juez: UVA

\*\*\*Tipo: KMP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1000005

char s[ MAX ];

int len , b[ MAX ];

void preprocess(){

int i = 0 , j = -1; b[ 0 ] = -1;

while( i < len ){

while( j >= 0 && s[ i ] != s[ j ] ) j = b[ j ];

i++; j++;

b[ i ] = j;

}

}

int main(){

int t , periodic;

while( scanf("%s" , &s ) , s[ 0 ] != '.' ){

len = strlen( s );

preprocess();

periodic = len - b[ len ];

printf("%d\n" , len/periodic );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Relatives

\*\*\*ID: 10299

\*\*\*Juez: UVA

\*\*\*Tipo: Number theory, totient function

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int countRelativelyPrime(int n) {

int sum = n, i;

for(i = 2; i\*i<=n; i++) {

if(n % i == 0) {

sum -= sum/i;

}

while(n%i == 0) n /= i;

}

if(n > 1) sum -= sum/n;

return sum;

}

int main(){

int n;

while(scanf("%d",&n) && n){

if( n == 0 )break;

if( n == 1 )printf("0\n");

else printf("%d\n",countRelativelyPrime(n));

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Rings and Glue

\*\*\*ID: 10301

\*\*\*Juez: UVA

\*\*\*Tipo: Geometry + Union Find

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>>

using namespace std;

#define MAX 105

int uf[ MAX ];

int Find( int x ){

return ( uf[ x ] == x ) ? x : uf[ x ] = Find( uf[ x ] );

}

void Union( int x , int y ){

uf[ Find( x ) ] = Find( y );

}

double dist( int x1 , int y1 , int x2 , int y2 ){

return sqrt( ( x1 - x2 )\*( x1 - x2 ) + ( y1 - y2 ) \*( y1 - y2 ) );

}

bool CircleIntersection( int x1 , int y1 , int x2 , int y2, double r1, double r2){

double d = dist( x1 , y1 , x2 , y2 );

if( d < r1 - r2 ) return false;

else if( d < r2 - r1 ) return false;

else if( d > (r1 + r2) ) return false;

return true;

}

int main(){

double x[ MAX ] , y[ MAX ], r[ MAX ];

int cnt[ MAX ];

int n, maxi;

while( scanf("%d" , &n ) , n != -1 ){

for( int i = 0 ; i < n ; ++i ){

uf[ i ] = i;

cnt[ i ] = 0;

scanf("%lf %lf %lf" , &x[ i ] ,&y[ i ] , &r[ i ] );

}

for( int i = 0 ; i < n ; ++i ){

for( int j = i + 1 ; j < n ; ++j ){

if( CircleIntersection( x[ i ] , y[ i ] , x[ j ] , y[ j ] , r[ i ] , r[ j ] ) ){

Union( i , j );

}

}

}

maxi = 0;

for( int i = 0 ; i < n ; ++i ) cnt[ Find( i ) ]++;

for( int i = 0 ; i < n ; ++i ) maxi = ( maxi < cnt[ i ] )?cnt[ i ]: maxi;

if( maxi == 1 )printf("The largest component contains %d ring.\n" , maxi );

else printf("The largest component contains %d rings.\n" , maxi );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# e-Coins

\*\*\*ID: 10306

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 45

#define MAXN 305

pair< int , int > a[ MAX ];

int dp[ MAXN ][ MAXN ];

int main(){

int t , m ,s , x , y;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &m , &s );

for( int i = 0 ; i < m ; ++i ){

scanf("%d %d" , &x , &y );

a[ i ] = make\_pair( x, y );

}

sort( a , a + m );

for( int i = 0 ; i <= s ; ++i )

for( int j = 0 ; j <= s ; ++j ) dp[ i ][ j ] = 1<<20;

dp[ 0 ][ 0 ] = 0;

for( int i = 0 ; i < m ; ++i )

{

for( int j = a[ i ].first ; j <= s ; ++j ){

for( int k = a[ i ].second ; k <= s ; ++k ){

dp[ j ][ k ] = min(dp[ j - a[ i ].first ][ k - a[ i ].second ] + 1, dp[ j ][ k ]);

}

}

}

int ans = 1<<20;

for( int i = 0 ; i <= s ; ++i ){

for( int j = 0 ; j <= s ; ++j ){

if( i \* i + j \* j == s \* s ){

ans = min( ans , dp[ i ][ j ] );

}

}

}

if( ans == 1<<20 ) puts("not possible");

else printf("%d\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Killing Aliens in Borg Maze

\*\*\*ID: 10307

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, BFS, MST

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX\_SIZE 101

struct Arista{

int origen;

int destino;

int peso;

//cambiar a > para arbol de recubrimiento de maxima suma

bool operator<(const Arista &a) const{

return peso<a.peso;

}

};

///UNION-FIND

int parent[MAX\_SIZE],rank[MAX\_SIZE];

void Make\_Set(const int x){

parent[x]=x;

rank[x]=0;

}

int Find(const int x){

if(parent[x]!=x) parent[x]=Find(parent[x]);

return parent[x];

}

void Union(const int PX, const int PY){

//int PX=Find(x),PY=Find(y);

if(rank[PX]>rank[PY]) parent[PY]=PX;

else{

parent[PX]=PY;

if(rank[PX]==rank[PY]) rank[PY]++;

}

}

void Merge(const int x,const int y){

Union(Find(x),Find(y));

}

///END UNION FIND

#define MAX 60

int f , c;

int posX[ MAX\_SIZE ] , posY[ MAX\_SIZE ];

char ady[ MAX ][ MAX ];

int dx[ 4 ] = { 1 , -1 , 0 , 0 };

int dy[ 4 ] = { 0 , 0 , 1 , -1 };

#define mp make\_pair

int dist[ MAX ][ MAX ];

void bfs( int x , int y ){

queue< pair<int , int> > Q;

Q.push( mp( x , y ) );

dist[ x ][ y ] = 0;

while( !Q.empty() ){

pair<int , int > act = Q.front(); Q.pop();

for( int i = 0 ; i < 4 ; ++i ){

int nx = act.first + dx[ i ];

int ny = act.second + dy[ i ];

if( nx >= 0 && nx < f && ny >= 0 && ny < c && ady[ nx ][ ny ] != '#'){

if( dist[ act.first ][ act.second ] + 1 < dist[ nx ][ ny ] ){

dist[ nx ][ ny ] = dist[ act.first ][ act.second ] + 1;

Q.push( mp( nx , ny ) );

}

}

}

}

}

int N;

void Kruskal( vector<Arista> v ){

int total = 0;

sort( v.begin() , v.end() );

for( int i = 0 ; i < N ; ++i ) Make\_Set( i );

for( int i = 0 ; i < v.size() ; ++i ){

int xi = v[ i ].origen, xf = v[ i ].destino, w = v[ i ].peso;

if( Find( xi ) != Find( xf ) ){

total += w;

Merge( xi , xf );

}

}

printf("%d\n" , total );

}

int main(){

int t ;

scanf("%d" ,&t );

while( t-- ){

scanf( "%d %d" , &c, &f );

N = 0;

memset( ady , '#' , sizeof( ady ) );

gets( ady[ 0 ] );

for( int i = 0 ; i < f ; ++i ){

gets( ady[ i ] );

int l = strlen( ady[ i ] );

for( int j = 0 ; j < l ; ++j ){

if( ady[ i ][ j ] == 'S' || ady[ i ][ j ] == 'A' ){

posX[ N ] = i;

posY[ N++ ] = j;

}

}

}

vector< Arista > v;

///hacemos bfs para hallar distancias entre nodos para hallar MST

for( int i = 0 ; i < N ; ++i ){

memset( dist , 0x3f , sizeof( dist ) );

bfs( posX[ i ] , posY[ i ] );

for( int j = 0 ; j < N ; ++j ){

Arista a;

a.origen = i; a.destino = j ; a.peso = dist[ posX[ j ] ][ posY[ j ] ];

v.push\_back( a );

}

}

Kruskal( v );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Roads in the North

\*\*\*ID: 10308

\*\*\*Juez: UVA

\*\*\*Tipo: Trees, diameter of a tree

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <queue>

#include <iostream>

using namespace std;

#define MAX 10005

struct Edge{

int v, w;

Edge( int vv , int ww) :v(vv) , w( ww ){}

};

vector<Edge> ady[ MAX ];

struct Estado{

int nodo, d;

Estado( int n , int dd ):nodo( n ) , d( dd ){}

};

bool seen[ MAX ];

//primero tiene distancia mas larga y segundo tiene nodo mas largo

pair<int , int > bfs( int x ){

memset( seen , 0 , sizeof( seen ) );

queue<Estado> Q;

Q.push( Estado( x , 0 ) );

seen[ x ] = true;

int large = 0, dist = 0;

while( !Q.empty() ){

Estado act = Q.front(); Q.pop();

if( dist < act.d ){

dist = act.d;

large = act.nodo;

}

for( int i = 0 ; i < ady[ act.nodo ].size(); ++i ){

Edge e = ady[ act.nodo ][ i ];

if( !seen[ e.v ] ){

seen[ e.v ] = true;

Q.push( Estado( e.v , e.w + act.d ) );

}

}

}

pair<int , int > ans( dist , large );

return ans;

}

int main(){

int u , v , w, V, i;

char line[ 2005];

while( 1 ){

V = 0;

for (i = 0; gets( line ); ) {

i++;

if( sscanf( line , "%d %d %d" ,&u ,&v ,&w ) != 3 )break;

ady[ u ].push\_back( Edge( v , w ) );

ady[ v ].push\_back( Edge( u , w ) );

}

if( i == 0 )break;

printf( "%d\n" , bfs( bfs( 1 ).second ).first );

for( i = 0 ; i < MAX ; ++i )ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Goldbatch and Euler

\*\*\*ID: 10311

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory, sieve, goldbatch conjecture

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/// MAX la raiz del limite maximo porque 10000\* 10000 nos da el total

/// y para saber si es primo es necesario solo iterar hasta i \* i <= MAX

#include <cmath>

#include <stdio.h>

#define MAX 10001

bool used[MAX];

int N;

int prime[2000],count\_p=0;

void sieve(){

prime[ count\_p++ ] = 2;

for(int i = 3 ; i <= (MAX) ; i += 2){

if( !used[ i ] ){

prime[ count\_p++ ] = i;

for(int j = i ; j <= MAX/i ; j += 2 )

used[ (j \* i) ] = true;

}

}

}

bool is\_prime( int n ){

if( n < MAX ) return !used[ n ];

int len = (int)sqrt( n );

for( int i = 0 ; i < count\_p && prime[ i ] <= sqrt( n ) ; ++i ){

if( n % prime[ i ] == 0 )return false;

}

return true;

}

int main(){

int n;

sieve();

while ( scanf( "%d", &n ) != EOF ){

if( n < 3 ){

printf("%d is not the sum of two primes!\n",n);

continue;

}

if( n&1 && n > 4){

if( is\_prime( n - 2 ) ){

printf("%d is the sum of 2 and %d.\n", n , n-2);

}

else printf("%d is not the sum of two primes!\n",n);

continue;

}

int start = n/2 - 1;

///si es par obviamente no es primo x eso le resto uno

start -= ( start & 1 ) < 1;

///start es impar por lo q si resto en unidades tendre par q nunca sera primo

///por eso resto de dos en dos

bool b = false;

int i = start;

for( ; i >=3 ; i -= 2 ){

if( is\_prime( i ) && is\_prime( n - i ) ){

b = true;

break;

}

}

if( b )printf("%d is the sum of %d and %d.\n",n, i , n - i );

else printf("%d is not the sum of two primes!\n",n);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Pay the Price

\*\*\*ID: 10313

\*\*\*Juez: UVA

\*\*\*Tipo: DP - Coin Change

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAXN 305

#define MAX 305

long long dp[ MAXN ][ MAXN ];

int \_left , \_right , n;

void solve( ){

memset( dp , 0 , sizeof(dp));

for( int i = 0 ; i < 301 ; ++i ){

dp[ 0 ][ i ]= 1;

}

for( int i = 1 ; i < 301 ; ++i ){

for( int j = 1 ; j < 301 ; ++j ){

dp[ i ][ j ] = dp[ i ][ j - 1 ];

if( i - j >= 0 )

dp[ i ][ j ] += dp[ i - j ][ j ];

}

}

}

int main(){

string s;

int N ,L1 ,L2;

solve();

while( getline( cin , s ) ){

switch( sscanf( s.c\_str() , "%d %d %d", &N , &L1, &L2 ) ){

case 1:

printf( "%lld\n" , dp[ N ][ N ] );

break;

case 2:

if( L1 > N ) L1 = N;

printf("%lld\n" , dp[ N ][ L1 ] );

break;

case 3:

if( L1 > N && L2 > N ){

printf("0\n");

break;

}

if( L1 > N ) L1 = N;

if( L2 > N ) L2 = N;

printf("%lld\n" , dp[ N ][ L2 ] - dp[ N ][ L1 - 1 ] );

break;

}

}

return 0;

}

/\*

300 2

100 10 10

100

200

200 30 75

200 111 199

300

299 88 1000

47 7 709

7 6 8

7 8 6

0 0 1

80 300

90 300

300 299

6 0 6

300 15 405

300 100 100

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Power Transmision

\*\*\*ID: 10330

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, max flow, vertex capacity

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

struct Edge{

int u , v , c, r;

Edge( int uu, int vv , int cc ,int rr): u(uu), v(vv), c(cc), r(rr){}

};

vector< Edge > E;

vector< vector<int > >ady;

int maxflow( int source , int sink ){

for( int flow = 0; ; ){

vector<int> prev( ady.size() , -1 );

queue<int> Q;

Q.push( source );

prev[ source ] = 0;

while( !Q.empty() ){

int act = Q.front(); Q.pop();

for( int i = 0 ; i < ady[ act ].size(); ++i ){

int e = ady[ act ][ i ], v = E[ e ].v;

if( prev[ v ] >= 0 || E[ e ].c == 0 ) continue;

prev[ v ] = e;

Q.push( v );

}

}

if( prev[ sink ] < 0 )return flow;

int c = 0x7ffffff;

for( int x = sink ; x != source ; x = E[ prev[ x ] ].u )

c <?= E[ prev[ x ] ].c;

for( int x = sink ; x != source ; x = E[ prev[ x ] ].u ){

E[ prev[ x ] ].c -= c;

E[ E[ prev[ x ] ].r ].c += c;

}

flow += c;

}

}

void add( int u , int v , int c ){

Edge e ( u , v , c , E.size() + 1 );

ady[ u ].push\_back( E.size() );

E.push\_back( e );

Edge e2( v, u , 0 , E.size() - 1 );

ady[ v ].push\_back( E.size() );

E.push\_back( e2 );

}

#define INF INT\_MAX

int main(){

int N, u , v , c, m, left, right;

while( scanf("%d", &N) != EOF ){

E.clear();

ady = vector< vector<int> > ( N + N + 5 , vector<int>() );

for( int i = 1 ; i <= N ;++i ){

scanf("%d", &c );

add( i , i + N , c );

}

scanf("%d",&m);

for( int i = 0 ; i < m ;++i ){

scanf("%d %d %d", &u, &v ,&c );

add( u + N , v , c );

}

scanf("%d %d", &left , &right );

for( int i = 0 ; i < left ; ++i ){

scanf("%d", &v );

add( 0 , v , INF );

}

for( int i = 0 ; i < right ; ++i ){

scanf("%d", &v );

add( v + N , N + N + 2 , INF );

}

printf("%d\n", maxflow( 0 , N + N + 2) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Mischievous Children

\*\*\*ID: 1033

\*\*\*Juez: UVA

\*\*\*Tipo: Factorial

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 21

unsigned long long fact[ MAX ];

void gen(){

fact[ 0 ] = 1;

for( int i = 1 ; i < MAX ; ++i ) fact[ i ] = i \* fact[ i - 1 ];

}

char s[ MAX ];

int cnt[ 30 ];

int main(){

gen();

int t , i , q;

scanf("%d" , &t );

unsigned long long ans;

for( q = 1 ; q <= t && scanf("%s" , &s ) ; ++q ){

memset( cnt , 0 , sizeof( cnt ) );

for( i = 0 ; s[ i ] ; ++i ) cnt[ s[ i ] - 'A' ]++;

ans = fact[ i ];

for( i = 0 ; s[ i ] ; ++i ){

ans /= cnt[ s[ i ] - 'A' ];

cnt[ s[ i ] - 'A' ]--;

}

printf("Data set %d: %llu\n" , q , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 23 Out of 5

\*\*\*ID: 10344

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

bool seen[ 6 ];

int a[ 6 ];

bool finish;

void solve( int len , int sum ){

if( finish ) return;

if( len == 5 ){

if( sum == 23 ){ finish = true; return;}

return;

}

for( int i = 0 ; i < 5 ; ++i ){

if( !seen[ i ] ){

seen[ i ] = 1;

solve( len + 1 , sum + a[ i ] );

solve( len + 1 , sum \* a[ i ] );

solve( len + 1 , sum - a[ i ] );

seen[ i ] = 0;

}

}

}

int main(){

while(scanf("%d %d %d %d %d" , &a[0], &a[1], &a[2], &a[3], &a[4]), a[0]|a[1]|a[2]|a[3]|a[4] ){

finish = false;

for( int i = 0 ; i < 5 ; ++i ){

memset( seen , 0 , sizeof( seen ) );

seen[ i ] = 1;

solve( 1 , a[ i ] );

if( finish ) break;

}

if( finish ) puts("Possible");

else puts("Impossible");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Antenna Placement

\*\*\*ID: 10349

\*\*\*Juez: UVA

\*\*\*Tipo: Max Independent Set - BPM

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAXL 405

#define MAXR 405

int seen[ MAXL ], matchL[ MAXL ] , matchR[ MAXR ];

vector< int > ady[ MAXL ];

int bpm( int left ){

if( seen[ left ] ) return 0;

seen[ left ] = 1;

int right , i;

for( i = 0 ; i < ady[ left ].size() ; ++i ){

right = ady[ left ][ i ];

if( matchR[ right ] == -1 || bpm( matchR[ right ] ) ){

matchR[ right ] = left;

matchL[ left ] = right;

return 1;

}

}

return 0;

}

#define MAX 50

char table[ MAX ][ MAX ];

int a[ MAX ][ MAX ] , dx[ ] = { 0 , 0 , 1 , -1 } , dy[ ] = { 1 , -1 , 0 , 0 };

int main(){

int t , q , h , w , white , black , i, j , k , nx , ny , V , cnt;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d %d" , &h , &w ) ){

white = black = 0;

for( i = 0 ; i < h && scanf("%s" , &table[ i ] ) ; ++i ){

for( j = 0 ; j < w ; ++j ){

if( table[ i ][ j ] == '\*' ){

if( ( i + j ) & 1 ) a[ i ][ j ] = white++;

else a[ i ][ j ] = black++;

}

}

}

for( i = 0 ; i < h ; ++i ){

for( j = 0 ; j < w ; ++j ){

if( table[ i ][ j ] == '\*' && ( ( i + j ) & 1 ) ){

for( k = 0 ; k < 4 ; ++k ){

nx = dx[ k ] + i;

ny = dy[ k ] + j;

if( nx >= 0 && ny >= 0 && nx < h && ny < w && table[ nx ][ ny ] == '\*' ){

ady[ a[ i ][ j ] ].push\_back( a[ nx ][ ny ] );

}

}

}

}

}

memset( matchL , -1 , sizeof( matchL ) );

memset( matchR , -1 , sizeof( matchR ) );

V = white + black;

cnt = 0;

for( i = 0 ; i < white ; ++i ){

memset( seen , 0 , sizeof( seen ) );

if( bpm( i ) ) cnt++;

}

printf("%d\n" ,V - cnt );

for( i = 0 ; i <= white ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Liftless EME

\*\*\*ID: 10350

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 2005

char s[ 15 ];

int m , n;

int getId( int x , int y ){ return x \* m + y; }

struct Edge{

int v , w;

Edge( int vv , int ww ): v( vv ) , w( ww ){}

};

vector< Edge > ady[ MAX ];

int dist[ MAX ];

#define INF 1<<30

int bfs(){

queue< int > Q;

int i , u , v , w;

for( i = 0 ; i < m \* n ; ++i ) dist[ i ] = INF;

for( i = 0 ; i < m ; ++i ){

Q.push( i );

dist[ i ] = 0;

}

while( !Q.empty() ){

u = Q.front(); Q.pop();

for( i = 0 ; i < ady[ u ].size() ; ++i ){

v = ady[ u ][ i ].v;

w = ady[ u ][ i ].w;

if( dist[ v ] > dist[ u ] + w ){

dist[ v ] = dist[ u ] + w;

Q.push( v );

}

}

}

int ans = INF;

for( i = 0 ; i < m ; ++i ){

ans = min( ans , dist[ getId( n - 1 , i ) ] );

}

return ans;

}

int main(){

int ans , i , t , hole , j , k;

while( scanf("%s" , &s ) != EOF ){

scanf("%d %d" , &n , &m );

ans = 2 \* ( n - 1 );

for( i = 0 ; i < n - 1 ; ++i ){

for( j = 0 ; j < m ; ++j , ++hole ){

for( k = 0 ; k < m && scanf("%d" , &t ) ; ++k ){

ady[ getId( i , j ) ].push\_back( Edge( getId( i + 1 , k ) , t ) );

}

}

}

printf("%s\n%d\n" , s , ans + bfs() );

for( i = 0 ; i < n \* m ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Rat Attack

\*\*\*ID: 10360

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 1025

int ady[ MAX ][ MAX ];

int main(){

int t , d , n , x , y , c , maxi ;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &d ,&n );

memset( ady , 0 , sizeof( ady ) );

maxi = 0;

while( n-- ){

scanf("%d %d %d" , &x , &y , &c );

for( int i = x - d ; i <= x + d ; ++i ){

if( i < 0 )continue;

if( i >= MAX ) break;

for( int j = y - d ; j <= y + d ; ++j ){

if( j < 0 ) continue;

if( j >= MAX )break;

ady[ i ][ j ] += c;

if( ady[ i ][ j ] > maxi ){

maxi = ady[ i ][ j ];

}

}

}

}

for( int i = 0 ; i < MAX ; ++i )

for( int j = 0 ; j < MAX ; ++j )

if( ady[ i ][ j ] == maxi ){

printf("%d %d %d\n" , i , j , ady[ i ][ j ] );

i = MAX;

break;

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Tic Tac Toe

\*\*\*ID: 10363

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

using namespace std;

char map[4][4];

bool vict(char c){

//filas

for(int i = 0; i < 3; i++)

for(int j = 0; j < 3; j++){

if( map[i][j] != c ) break;

if( j == 2 ) return 1;

}

//columnas

for(int i = 0; i < 3; i++)

for(int j = 0; j < 3; j++){

if( map[j][i] != c ) break;

if( j == 2 ) return 1;

}

//diagonal derecha

for( int i = 0 ; i < 3 ; i++){

if( map[i][i] != c ) break;

if( i == 2 ) return 1;

}

//diagonal izquierda

for(int i = 0 ; i < 3 ; i++){

if( map[i][2-i] != c ) break;

if( i == 2 ) return 1;

}

return 0;

}

bool check(){

int no, nx, nf;

no = nx = nf = 0;

for(int i=0; i<3; i++)

for(int j=0; j<3; j++)

if(map[i][j] == 'O') no++;

else if(map[i][j] == 'X') nx++;

else nf++;

if( nx < no || nx > no + 1 ) return 0;

if( vict('O') && nx > no ) return 0;

if( vict('X') && no == nx ) return 0;

if( !nf || vict('O') || vict('X') ) return 1;

return 1;

}

int main(){

int t;

cin >> t;

while( t-- ){

for(int i=0; i<3; i++)

for(int j=0; j<3; j++)

cin >> map[i][j];

if( check() ) cout << "yes\n";

else cout << "no\n";

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Square

\*\*\*ID: 10364

\*\*\*Juez: UVA

\*\*\*Tipo: DP + Bitmask

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int a[ 25 ], n, suma;

int seen[ 10001 ];

bool dfs( int val , int last , int len){

if( val == suma ){

len++;

val = 0;

last = 0;

}

if( len == 4 ) return true;

for( int i = last ; i < n ; ++i ){

if( seen[ i ] )continue;

seen[ i ] = true;

if( ( a[ i ] + val ) <= suma ){

if( dfs( val + a[ i ] , i + 1 , len ) ) return true;;

}

else{

seen[ i ] = false;

break;

}

seen[ i ] = false;

}

return false;

}

int main(){

int t, sum;

scanf("%d" , &t );

while( t-- ){

sum = 0;

scanf("%d" , &n );

for( int i = 0 ; i < n ; ++i ){ scanf("%d" , &a[ i ] ); sum += a[ i ]; }

if( sum % 4 != 0 ){

printf("no\n");

continue;

}

suma = sum / 4;

memset( seen , 0 , sizeof( seen ) );

sort( a , a + n );

if( dfs( 0 , 0 , 0 ) )printf("yes\n");

else printf("no\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Blocks

\*\*\*ID: 10365

\*\*\*Juez: UVA

\*\*\*Tipo: Brute force

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 1000

int getArea( int x , int y , int z ){

return ( x \* y \* 2 + z \* y \* 2 + x \* z \* 2 );

}

int main(){

int t , n , ans, area;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

ans = 1<<30;

for( int i = 1 ; i <= MAX ; ++i ){

if( i > n )break;

for( int j = i ; j <= MAX ; ++j ){

if( i \* j > n )break;

for( int k = j ; k <= MAX ; ++k ){

if( i \* j \* k > n )break;

if( i \* j \* k == n ){

area = getArea( i , j , k );

if( ans > area ) ans = area;

}

}

}

}

printf("%d\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Election

\*\*\*ID: 10374

\*\*\*Juez: UVA

\*\*\*Tipo: Data Structure, Map

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t , n;

char s[ 85 ], s2[ 85 ];

scanf("%d" , &t );

map<string , int >::iterator it;

for( int q = 0 ; q < t ; ++q ){

if( q )putchar('\n');

scanf("%d" , &n );

gets( s );

map<string , string > name;

map<string , int > mp;

while( n-- ){

gets( s );

gets( s2 );

name[ s ] = s2;

mp[ s ] = 0;

}

scanf("%d" , &n );

gets( s );

while( n-- ){

gets( s );

mp[ s ]++;

}

int maxi = 0;

bool b = true;

string resp;

for( it = mp.begin() ; it != mp.end() ; ++it ){

if( (\*it).second > maxi ){

maxi = (\*it).second;

resp = (\*it).first;

}

}

for( it = mp.begin() ; it != mp.end() ; ++it ){

if( (\*it).second == maxi && (\*it).first != resp ){

b = false;

break;

}

}

if( b ) printf("%s\n" , name[ resp ].c\_str() );

else puts( "tie" );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Maze Traversal

\*\*\*ID: 10377

\*\*\*Juez: UVA

\*\*\*Tipo: Simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 65

char ady[ MAX ][ MAX ];

int f , c;

//N W S E

int dx[ 4 ] = { -1 , 0 , 1 , 0 };

int dy[ 4 ] = { 0 , -1 , 0 , 1 };

char commands[ 105 ];

void simulate( int x , int y , int dir ){

int nx , ny , i;

char command;

while( gets( commands ) ){

for( i = 0 ; commands[ i ] ; ++i ){

command = commands[ i ];

if( command == 'R' ) dir = ( dir - 1 + 4 ) % 4;

else if( command == 'L') dir = ( dir + 1 ) % 4;

else if( command == 'F' ){

nx = x + dx[ dir ]; ny = y + dy[ dir ];

if( nx >= 0 && nx < f && ny >= 0 && ny < c && ady[ nx ][ ny ] != '\*' ){

x = nx;

y = ny;

}

}

else if( command == 'Q' ) break;

}

if( command == 'Q') break;

}

printf("%d %d %c\n" , x + 1 , y + 1 , ( dir == 0 )?'N': ( dir == 1 )?'W': ( dir == 2 )?'S':'E' );

}

int main(){

int t , x , y;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &f , &c );

gets( ady[ 0 ] );

for( int i = 0 ; i < f ; ++i ){

gets( ady[ i ] );

}

scanf("%d %d" , &x , &y );

gets( commands );

simulate( x - 1 , y - 1 , 0 );

if( t > 0 ) printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Watering Grass

\*\*\*ID: 10382

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

#include <cmath>

#include <cstdio>

using namespace std;

#define MAX 10055

#define EPS 1e-6

struct Interval{

double ini , fin;

Interval( double i , double f ): ini( i ) , fin( f ){}

Interval(){}

bool operator < ( const Interval t1 ) const{

return ini < t1.ini;

}

}a[ MAX ];

double getDx( double R , double W ){

return sqrt( R \* R - ( W \* W / 4.0 ) );

}

int main(){

int n , w , h, x , r , ans , len;

double dx ,lastRight, newRight;

while( scanf("%d %d %d" , &n , &h , &w ) == 3 ){

for( int i = 0 ; i < n ; ++i ){

scanf("%d %d" , &x , &r );

if( r \* 2 < w ){

a[ i ].ini = a[ i ].fin = 0;

continue;

}

dx = getDx( (double)r , (double)w );

a[ i ].ini = x - dx;

a[ i ].fin = x + dx;

}

sort( a , a + n );

ans = 0;

lastRight = 0;

for( int i = 0 , j = 0 ; i < n ; i = j ){

newRight = 0;

while( j < n && a[ j ].ini <= lastRight ){

if( a[ j ].fin > newRight ){

newRight = a[ j ].fin;

}

++j;

}

lastRight = newRight;

ans++;

if( j == i || lastRight + EPS > h ) break;

}

if( lastRight + EPS > h ) printf("%d\n" , ans );

else printf("-1\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The One-Handed Typist

\*\*\*ID: 10393

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc Strings

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 55

#define MAXN 1005

int f , n , del[ 15 ] , valid[ MAXN ];

char s[ MAXN ][ MAX ];

char table[ 11 ][ 7 ] = { "qaz" , "wsx" , "edc" , "rfvtgb" , "space" ,

"space" , "yhnujm" , "ik," , "ol." , "p;/"};

bool possible( char c){

int i , j;

for( i = 0 ; i < 10 ; ++i ){

if( del[ i ] ) continue;

for( j = 0 ; table[ i ][ j ] ; ++j )

if( table[ i ][ j ] == c ) return true;

}

return false;

}

int main(){

int i , x , j , maxi , len ;

while( scanf("%d %d" , &f , &n ) != EOF ){

memset( del , 0 , sizeof( del ) );

for( i = 0 ; i < f && scanf("%d" , &x ) ; ++i ) del[ x - 1 ] = 1;

maxi = 0;

for( i = 0 ; i < n && scanf("%s" , &s[ i ] ) ; ++i ){

valid[ i ] = 0;

for( j = 0 ; s[ i ][ j ] ; ++j )

if( !possible( s[ i ][ j ] ) ){ j = -1; break;}

if( j != -1 ){

maxi = max( maxi , j );

valid[ i ] = 1;

}

}

set< string > ss;

for( i = 0 ; i < n ; ++i ){

if( strlen( s[ i ] ) == maxi && valid[ i ] ) ss.insert( s[ i ] );

}

printf("%d\n" , ss.size() );

for( set< string > :: iterator it = ss.begin() ; it != ss.end() ; ++it )

printf("%s\n" , (\*it).c\_str());

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Game Show Math

\*\*\*ID: 10400

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

int n, a[ MAX ] , dp[ 32005 ][ MAX ][ 2 ] , res;

bool end;

bool solve( int sum , int K , string s ){

if( end ) return 1;

if( K == n ){

if( sum == res ) {

printf("%d" , a[ 0 ] );

for( int i = 1 ; i < n ; ++i ){

printf("%c%d", s[ i - 1 ] , a[ i ] );

}

printf("=%d\n" , res );

end = 1;

return 1;

}

return 0;

}

if( sum < 0 ){

if( dp[ -sum ][ K ][ 0 ] != -1 ) return dp[ -sum ][ K ][ 0 ];

}

else if( dp[ sum ][ K ][ 1 ] != -1 ) return dp[ sum ][ K ][ 1 ];

bool ans = 0;

if( sum - a[ K ] >= -32000 && sum - a[ K ] <= 32000 ){

ans |= solve( sum - a[ K ] , K + 1 , s + "-" );

}

if( sum + a[ K ] >= -32000 && sum + a[ K ] <= 32000 ){

ans |= solve( sum + a[ K ] , K + 1 , s + "+" );

}

if( sum \* a[ K ] >= -32000 && sum \* a[ K ] <= 32000 ){

ans |= solve( sum \* a[ K ] , K + 1 , s + "\*" );

}

if( sum % a[ K ] == 0 && sum/a[ K ] >= -32000 && sum/a[ K ] <= 32000 ){

ans |= solve( sum / a[ K ] , K + 1 , s + "/" );

}

if( sum < 0 ) return dp[ -sum ][ K ][ 0 ] = ans;

return dp[ sum ][ K ][ 1 ] = ans;

}

int main(){

int t ;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

for( int i = 0 ; i < n; ++i ) scanf("%d" , &a[ i ] );

scanf("%d" , &res );

memset( dp , -1 , sizeof( dp ) );

end = false;

if( !solve( a[ 0 ] , 1 , "" ) ) puts("NO EXPRESSION");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Injured Queen Problem

\*\*\*ID: 10401

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <string.h>

#define MAX 20

int n;

char s[ MAX ];

long long dp[ MAX ][ MAX ];

int ady[ MAX ][ MAX ] , row[ MAX ] , col[ MAX ], seen[ MAX ][ MAX ] , aux[ MAX ][ MAX ];

int kx[ 8 ] = { 1 , -1 , 0 , 0 , 1 , 1 , -1 , -1 };

int ky[ 8 ] = { 0 , 0 , 1 , -1 , 1 , -1 , 1 , -1 };

int dx[ 2 ] = { -2 , 2 };

int dy[ 2 ] = { 1 };

void solve(){

memset( dp , 0 , sizeof( dp ) );

int i , j , nx , ny , x;

if( col[ 0 ] >= 0 ) dp[ col[ 0 ] ][ 0 ] = 1;

for( j = 0 ; j < n - 1 ; ++j ){

if( col[ j ] >= 0 ){

//si no hay pieza en la columna siguiente

if( col[ j + 1 ] == -1 ){

for( i = 0 ; i < n ; ++i ){

if( !seen[ i ][ j + 1 ] ) dp[ i ][ j + 1 ] += dp[ col[ j ] ][ j ];

}

}

//si hay cuento todo a esa pieza

else{

dp[ col[ j + 1 ] ][ j + 1 ] = dp[ col[ j ] ][ j ];

}

}

else{

if( j == 0 ){

//la segunda columna puede tener una pieza

for( i = 0 ; i < n ; ++i ){

if( !seen[ i ][ j ] ) dp[ i ][ j ] = 1;

}

}

//pruebo en cada fila de esta columna con una pieza y voy contando

for( x = 0 ; x < n ; ++x ){

if( seen[ x ][ j ] ) continue;

memcpy( aux , seen , sizeof( seen ) );

aux[ x ][ j ] = 1;

for( i = 0 ; i < 8 ; ++i ){

nx = kx[ i ] + x; ny = ky[ i ] + j;

if( nx >= 0 && ny >= 0 && nx < n && ny < n ){

aux[ nx ][ ny ] = 1;

}

}

//si no hay pieza en la columna siguiente

if( col[ j + 1 ] == -1 ){

for( i = 0 ; i < n ; ++i ){

if( !aux[ i ][ j + 1 ] ) dp[ i ][ j + 1 ] += dp[ x ][ j ];

}

}

//si hay cuento todo a esa pieza

else{

dp[ col[ j + 1 ] ][ j + 1 ] += dp[ x ][ j ];

}

}

}

}

long long ans = 0;

//cuento los posibles finales

if( col[ n - 1 ] == -1 ){

for( i = 0 ; i < n ; ++i ){

ans += dp[ i ][ n - 1 ];

}

}

//sino solo cuento finales en la pieza ingresada

else ans += dp[ col[ n - 1 ] ][ n - 1 ];

printf("%lld\n" , ans );

}

int correct(){

int i , nx , ny , j;

for( i = 0 ; i < n - 1 ; ++i ){

if( col[ i ] >= 0 && col[ i + 1 ] >= 0 ){

for( j = 0 ; j < 8 ; ++j ){

nx = kx[ j ] + col[ i ]; ny = ky[ j ] + i;

if( nx >= 0 && ny >= 0 && nx < n && ny < n ){

if( nx == col[ i + 1 ] && ny == i + 1 ) return 0;

}

}

}

}

return 1;

}

int main(){

int i , j , nx , ny;

while( scanf("%s" , &s ) != EOF ){

memset( seen , 0 , sizeof( seen ) );

memset( col , -1 , sizeof( col ) );

n = strlen( s );

if( n == 1 ){

printf("1\n" );

continue;

}

for( i = 0 ; s[ i ] ; ++i )

{

if( s[ i ] != '?' ){

if( s[ i ] >= '0' && s[ i ] <= '9'){

seen[ s[ i ] - '0' - 1 ][ i ] = 1;

for( j = 0 ; j < 8 ; ++j ){

nx = kx[ j ] + s[ i ] - '0' - 1; ny = ky[ j ]+ i;

if( nx >= 0 && ny >= 0 && nx < n && ny < n ){

seen[ nx ][ ny ] = 1;

}

}

col[ i ] = s[ i ] - '0' - 1;

}else{

seen[ s[ i ] - 'A' + 9 ][ i ] = 1;

col[ i ] = s[ i ] - 'A' + 9;

for( j = 0 ; j < 8 ; ++j ){

nx = kx[ j ] + s[ i ] - 'A' + 9; ny = ky[ j ]+ i;

if( nx >= 0 && ny >= 0 && nx < n && ny < n ){

seen[ nx ][ ny ] = 1;

}

}

}

}

}

if( correct() ) solve();

else printf("0\n");

}

return 0;

}

/\*

1?4??3

6???4???

???A???B????E??

???A???BC???E??

???A???B??4?E?F

??

?

1??

??2

??????????????A

??????????????F

??????????????1\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Simple Division

\*\*\*ID: 10407

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory, GCD

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1001

int main(){

int n,i,min;

int a[MAX]={0};

while(scanf("%d",&n) && n){

i=0;

min=n;

a[i++]=n;

while(scanf("%d",&n) && n){

a[i++]=n;

min=std::min(min,n);

}

for(int j=0;j<i;j++)a[j]-=min;

int q=a[0];

for(int j=1;j<i;j++)q=\_\_gcd(q,a[j]);

printf("%d\n",q);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Die Game

\*\*\*ID: 10409

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

char s[ 7 ];

int dice[ 3 ][ 4 ];

int main(){

int q , tmp;

while( scanf("%d" , &q ) , q ){

dice[ 0 ][ 1 ] = 2;

dice[ 1 ][ 0 ] = 3;

dice[ 1 ][ 1 ] = 1;

dice[ 1 ][ 2 ] = 4;

dice[ 2 ][ 1 ] = 5;

dice[ 3 ][ 1 ] = 6;

while( q-- ){

scanf("%s" , &s );

if( s[ 0 ] == 'n' ){

tmp = dice[ 0 ][ 1 ];

dice[ 0 ][ 1 ] = dice[ 1 ][ 1 ];

dice[ 1 ][ 1 ] = dice[ 2 ][ 1 ];

dice[ 2 ][ 1 ] = dice[ 3 ][ 1 ];

dice[ 3 ][ 1 ] = tmp;

}

else if( s[ 0 ] == 's' ){

tmp = dice[ 3 ][ 1 ];

dice[ 3 ][ 1 ] = dice[ 2 ][ 1 ];

dice[ 2 ][ 1 ] = dice[ 1 ][ 1 ];

dice[ 1 ][ 1 ] = dice[ 0 ][ 1 ];

dice[ 0 ][ 1 ] = tmp;

}

else if( s[ 0 ] == 'w' ){

tmp = dice[ 1 ][ 0 ];

dice[ 1 ][ 0 ] = dice[ 1 ][ 1 ];

dice[ 1 ][ 1 ] = dice[ 1 ][ 2 ];

dice[ 1 ][ 2 ] = dice[ 3 ][ 1 ];

dice[ 3 ][ 1 ] = tmp;

}

else if( s[ 0 ] == 'e' ){

tmp = dice[ 1 ][ 2 ];

dice[ 1 ][ 2 ] = dice[ 1 ][ 1 ];

dice[ 1 ][ 1 ] = dice[ 1 ][ 0 ];

dice[ 1 ][ 0 ] = dice[ 3 ][ 1 ];

dice[ 3 ][ 1 ] = tmp;

}

}

printf("%d\n" , dice[ 1 ][ 1 ] );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Knights in FEN

\*\*\*ID: 10422

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, BFS, MAP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int dx[ 8 ]={ -2 , -2 , -1 , -1 , 1 , 1 , 2 , 2 };

int dy[ 8 ]={ -1 , 1 , -2 , 2 , -2 , 2 , -1 , 1 };

struct Nodo{

char ady[ 5 ][ 5 ];

int d;

};

bool ok( char str[ ]){

if( strcmp( str , "111110111100 110000100000") == 0 ) return true;

return false;

}

char a[ 5 ][ 6 ] , ini[ 26 ];

int bfs( ){

queue< Nodo> Q;

Nodo nodo;

for( int i = 0 ; i < 5 ; ++i ){

for( int j = 0 ; j < 5 ; ++j ){

nodo.ady[ i ][ j ] = a[ i ][ j ];

}

}

nodo.d = 0;

Q.push( nodo );

map<string, bool > seen;

seen[ ini ] = true;

while( !Q.empty() ){

Nodo act = Q.front(); Q.pop();

for( int i = 0 ; i < 5 ; ++i ){

int j = 0;

for( ; j < 5 ; ++j ){

if( act.ady[ i ][ j ] == ' '){

for( int k = 0 ; k < 8 ; ++k ){

int nx = i + dx[ k ];

int ny = j + dy[ k ];

if( nx >= 0 && nx < 5 && ny >= 0 && ny < 5 ){

Nodo vec;

for( int q = 0; q < 5 ; ++q ){

for( int k = 0 ; k < 5 ; ++k ){

a[ q ][ k ] = act.ady[ q ][ k ];

}

}

a[ i ][ j ] = a[ nx ][ ny ];

a[ nx ][ ny ] = ' ';

int idx = 0;

for( int q = 0; q < 5 ; ++q ){

for( int k = 0 ; k < 5 ; ++k ){

ini[ idx++ ] = a[ q ][ k ];

vec.ady[ q ][ k ] = a[ q ][ k ];

}

}

ini[ idx ] = '\0';

if( !seen[ ini ] ){

seen[ ini ] = true;

vec.d = act.d + 1;

if( ok( ini ) ){

return vec.d;

}

if( vec.d > 10 )return -1;

Q.push( vec );

}

}

}

break;

}

}

if( j != 5 )break;

}

}

return -1;

}

int main(){

int t , idx;

scanf("%d" ,&t );

getchar();

while( t-- ){

idx = 0;

for( int i = 0 ; i < 5 ; ++i ){

gets( a[ i ] );

for( int j = 0 ; j < 5 ; ++j ){

ini[ idx++ ] = a[ i ][ j ];

}

}

ini[ idx ] = '\0';

if( ok( ini ) ) {printf("Solvable in 0 move(s).\n"); continue;}

int resp = bfs( );

if( resp == -1 )printf("Unsolvable in less than 11 move(s).\n");

else printf("Solvable in %d move(s).\n" , resp );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Naughty Sleepy Boys

\*\*\*ID: 10427

\*\*\*Juez: UVA

\*\*\*Tipo: Math

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#define MAX 20

int start[ MAX ], end[ MAX ];

void gen(){

start[ 0 ] = 1; end[ 0 ] = 9;

start[ 1 ] = 10; end[ 1 ] = 189;

for( int i = 1 ; i < 10 ; ++i ){

start[ i ] = start[ i - 1 ] \* 10;

end[ i ] = ( ( start[ i ] \* 10 ) - start[ i ] ) \* ( i + 1 ) + end[ i - 1 ];

}

}

int getDigit( int num , int idx ){

int cnt = 0;

while( num > 0 ){

if( cnt == idx ) return num % 10;

cnt++;

num /= 10;

}

return num % 10;

}

int main(){

gen();

int n , idx, num;

while( scanf("%d" , &n ) != EOF ){

//busco cuantos digitos tiene el indice

for( idx = 0 ; idx < 9 ; ++idx )

if( n <= end[ idx ] ) break;

num = start[ idx ]; //numero inicial

//veo la posicion comenzando desde 0 en el bloq

n = n - 1 - ( ( idx - 1 >= 0 )? end[ idx - 1 ]:0 );

num += n / ( idx + 1 );

printf("%d\n" , getDigit( num , idx - ( n % ( idx + 1 ) ) ) );

}

return 0;

}

/\*

189

190

191

192

193

1000

10000

100000

1000000

10000000

99999999

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Polygon Inside A Circle

\*\*\*ID: 10432

\*\*\*Juez: UVA

\*\*\*Tipo: Geometry

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cmath>

using namespace std;

const long double PI = 2 \* acos(.0);

//Convert angle from degrees to radians

long double Deg\_to\_Rad( double angle ){

return angle \* PI/180.0;

}

int main(){

long double radius, n;

while( scanf("%Lf %Lf", &radius , &n ) != EOF ){

long double angle = 360.0/n;

long double chord = sqrt( 2.0 \* radius \* radius \* ( 1.0 - cos( Deg\_to\_Rad( angle ) ) ));

long double p = (radius + radius + chord)/2.0;

long double areaTriangle = sqrt( p \* ( p - radius ) \* ( p - radius ) \* ( p - chord ) );

printf("%.3Lf\n" , n \* areaTriangle );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Ferry Loading

\*\*\*ID: 10440

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t , q , n ,m , k, total;

int viajes , x , aux , paso , j , l;

scanf("%d" , &q );

while( q-- ){

scanf("%d %d %d" , &n ,&t ,&m );

l = m%n; k = m/n;

total = 0; viajes = k;

if( l ){

viajes++;

for( int i = 0 ; i < l ; ++i ){

scanf("%d" , &x );

total = x + 2 \*t;

}

}

for( int i = 0 ; i < k ; ++i ){

for( int j = 0 ; j < n ; ++j )

scanf("%d" , &x );

if( x <= total ) total += 2\*t;

else total = x + 2 \* t;

}

printf("%d %d\n" , total - t , viajes );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Catenyms

\*\*\*ID: 10441

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Euler Tour, Connectivity

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 26

int aux[ MAX ][ MAX ];

vector<string> ady[ MAX ];

int in[ MAX ], out[ MAX ];

int ini;

bool caminoEulerianoDirigido(){

int cont1 = 0, cont2 = 0 , cont3 = 0;

for( int i = 0 ; i < MAX ; ++i ){

if( in[ i ] == out[ i ] + 1 ) { cont1++; }

if( out[ i ] == in[ i ] + 1 ) { cont2++; ini = i; }

if( in[ i ] == out[ i ] ) cont3++;

}

if( cont1 == 0 && cont2 == 0 ){

for( int i = 0 ; i < MAX ; ++i ) if( ady[ i ].size() != 0 ){ini = i;break;}

}

if( cont3 == MAX || ( cont1 <= 1 && cont2 <= 1 && cont1 + cont2 + cont3 == MAX) ) return true;

return false;

}

///vemos si es conexo x medio de transitive closure, solo para nodos <= 100

///si es mas usar union find

bool connected(){

for( int k = 0 ; k < MAX ; ++k ){

for( int i = 0 ; i < MAX ; ++i ){

for( int j = 0 ; j < MAX ; ++j ){

aux[ i ][ j ] |= aux[ i ][ k ] & aux[ k ][ j ];

}

}

}

for( int i = 0 ; i < MAX ; ++i ){

for( int j = 0 ; j < MAX ; ++j ){

if( i != j && in[ i ] + out[ i ] > 0 && in[ j ] + out[ j ] > 0 && !aux[ i ][ j ] ){

return false;

}

}

}

return true;

}

int cmp( const void \*a , const void \*b ){

char \*x = ( char \*) a, \*y = ( char \*) b;

return -strcmp( x , y );

}

vector<string> resp;

void tour(int x)

{

while ( ady[ x ].size() > 0) {

string y = ady[ x ].back();

ady[ x ].pop\_back();

tour( y[ y.size() - 1 ] - 'a' );

resp.push\_back( y );

}

}

int main(){

int t, n;

scanf("%d", &t );

char s[ MAX ];

while( t-- ){

for( int i = 0 ; i < MAX ; ++i )ady[ i ].clear();

scanf("%d", &n );

memset( in , 0 , sizeof( in ) );

memset( out , 0 , sizeof( out ) );

memset( aux , 0 ,sizeof( aux ) );

for( int i = 0 ; i < n ; ++i ){

scanf("%s", s);

int u = s[ 0 ] - 'a' , v = s[ strlen( s ) - 1 ] - 'a';

ady[ u ].push\_back( string( s ) );

aux[ u ][ v ] = aux[ v ][ u ] = 1;

out[ u ]++;

in[ v ]++;

}

///verificamos q sea conexo

if( !connected() ){

printf("\*\*\*\n");

continue;

}

///verificamos reglas de camino de euler en grafo dirigido

if( caminoEulerianoDirigido() ){

for( int i = 0 ; i < MAX ; ++i ) {

sort( ady[ i ].begin(), ady[ i ].end() );

reverse( ady[ i ].begin() , ady[ i ].end() );

}

resp.clear();

tour( ini );

printf("%s", resp[ resp.size() - 1].c\_str() );

for( int i = resp.size() - 2 ; i >= 0 ; --i ){

printf(".%s", resp[ i ].c\_str() );

}

printf("\n");

}

else printf("\*\*\*\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Rock, Scissors, Paper

\*\*\*ID: 10443

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 105

int ady[ MAX ][ MAX ];

int next[ MAX ][ MAX ];

int dx[ 4 ] = { 1 , -1 , 0 , 0 };

int dy[ 4 ] = { 0 , 0 , 1 , -1 };

int main(){

int t , h , w , it;

scanf("%d" , &t );

for( int q = 0 ; q < t ; ++q ){

if( q )printf("\n");

scanf("%d %d %d" , &h , &w , &it );

for( int i = 0 ; i < h ; ++i )

for( int j = 0 ; j < w ; ++j )

scanf(" %c" , &ady[ i ][ j ] );

while( it-- ){

memset( next , 0 , sizeof( next ) );

for( int i = 0 ; i < h ; ++i ){

for( int j = 0 ; j < w ; ++j ){

if( !next[ i ][ j ] )next[ i ][ j ] = ady[ i ][ j ];

for( int k = 0 ; k < 4 ; ++k ){

int nx = dx[ k ] + i , ny = dy[ k ] + j;

if( nx >= 0 && ny >= 0 && nx < h && ny < w ){

if( ady[ i ][ j ] == 'R' && ady[ nx ][ ny ] == 'S' )next[ nx ][ ny ] = 'R';

if( ady[ i ][ j ] == 'S' && ady[ nx ][ ny ] == 'P' )next[ nx ][ ny ] = 'S';

if( ady[ i ][ j ] == 'P' && ady[ nx ][ ny ] == 'R' )next[ nx ][ ny ] = 'P';

}

}

}

}

memcpy( ady , next , sizeof( next ) );

}

for( int i = 0 ; i < h ; ++i ){

for( int j = 0 ; j < w ; ++j ){

printf("%c" , ady[ i ][ j ] );

}

printf("\n");

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Is There A Second Way Left

\*\*\*ID: 10462

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory , MST

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

#include <cstring>

#include <iostream>

using namespace std;

#define MAX 105

int uf[ MAX ];

int Find( int x ){

return ( uf[ x ] == x ) ? x : uf[ x ] = Find( uf[ x ] );

}

void Union( int x , int y ){

uf[ Find( x ) ] = Find( y );

}

struct Arista{

int u , v , w;

bool mst;

Arista( int uu , int vv , int ww , bool mstt ): u( uu ) , v( vv ) , w( ww ) , mst( mstt ){}

Arista(){}

bool operator <( const Arista &a ) const{

return w < a.w;

}

}graph[ MAX ];

int main(){

int t, V , E , u , v , w , first , second , aux , size;

scanf("%d" , &t );

for( int q = 1 ; q <= t ; ++q ){

first = size = 0;

scanf("%d %d" , &V , &E );

for( int i = 0 ; i < E ; ++i ){

scanf("%d %d %d" , &u , &v , &w );

graph[ i ] = Arista( u , v , w , false );

}

sort( graph , graph + E );

for( int i = 1 ; i <= V ; ++i ) uf[ i ] = i;

for( int i = 0 ; i < E ; ++i ){

u = graph[ i ].u; v = graph[ i ].v; w = graph[ i ].w;

if( Find( u ) != Find( v ) ){

first += w;

size++;

graph[ i ].mst = true;

Union( u , v );

}

}

printf("Case #%d : " , q );

if( size != V - 1 ){

printf("No way\n");

continue;

}

///vemos aristas del MST y probamos nuevo MST eliminando una de dichas aristas

second = 1000000;

for( int i = 0 ; i < E ; ++i ){

if( !graph[ i ].mst ) continue;

aux = 0; size = 0;

for( int j = 1 ; j <= V ; ++j ) uf[ j ] = j;

for( int j = 0 ; j < E ; ++j ){

if( i == j )continue;

u = graph[ j ].u; v = graph[ j ].v; w = graph[ j ].w;

if( Find( u ) != Find( v ) ){

aux += w;

size++;

Union( u , v );

}

}

///si tengo todos los nodos, puede ser q la arista q halla eliminado sea la unica

///coneccion a ese nodo

if( size == V - 1 && second > aux ) second = aux;

}

if( second == 1000000 )printf("No second way\n");

else printf("%d\n" , second );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Homer Simpson

\*\*\*ID: 10465

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

int dp[ MAX ], seen[ MAX ];

int n , m , t;

int solve( int sum , int cont ){

if( sum < 0 ) return -1;

if( dp[ sum ] != -1 ) return dp[ sum ];

int maxi = 0;

if( sum - n < 0 ){

if( sum - m < 0 ){

maxi = cont;

}

else{

maxi = max( solve( sum - m , cont + 1 ) , cont ) ;

}

}

else {

if( sum - m < 0 ){

maxi = max( solve( sum - n , cont + 1 ) , cont );

}

else{

maxi = max( solve( sum - m , cont + 1 ) , solve( sum - n , cont + 1 ) );

}

}

return dp[ sum ] = maxi;

}

int main(){

while( scanf("%d %d %d" , &n , &m , &t ) != EOF ){

memset( dp , 0 , sizeof( dp ) );

memset( seen , 0 , sizeof( seen ) );

seen[ t ] = 1;

for( int i = t ; i >= 0 ; --i ){

if( seen[ i ] ){

if( i - n >= 0 ){

dp[ i - n ] = max( dp[ i - n ] , dp[ i ] + 1 );

seen[ i - n ] = 1;

}

if( i - m >= 0 ){

dp[ i - m ] = max( dp[ i - m ] , dp[ i ] + 1 );

seen[ i - m ] = 1;

}

}

}

int i = 0;

for( ; i <= t ; ++i ) if( seen[ i ] ) break;

if( i == 0 ) printf("%d\n" , dp[ i ] );

else printf("%d %d\n" , dp[ i ] , i );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# To Carry or not to Carry

\*\*\*ID: 10469

\*\*\*Juez: UVA

\*\*\*Tipo: Bitwise

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main(){

unsigned int a , b;

while( scanf("%u %u" , &a ,&b ) != EOF ){

printf("%u\n" , a^b );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Help the Leaders

\*\*\*ID: 10475

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

bool isLowerCase(char c){return c>='a' && c<='z';}

char toUpperCase(char c){return (isLowerCase(c))?(c-32):c;}

int n , sz;

string name[ 25 ];

bool cmp( string s1 , string s2 ){

int l1 = s1.length() , l2 = s2.length();

return l1 > l2 || ( l1 == l2 && s1 < s2 );

}

bool visto[ 25 ][ 25 ];

bool seen[ 25 ];

int ans[ 25 ];

void solve( int len , int last ){

if( len == sz ){

for( int i = 0 ; i < len ; ++i ){

for( int j = i + 1 ; j < len ; ++j ){

if( visto[ ans[ i ] ][ ans[ j ] ] || visto[ ans[ j ] ][ ans[ i ] ] )return;

}

}

for( int i = 0 ; i < sz ; ++i ){

if( i ) printf(" ");

printf("%s" , name[ ans[ i ] ].c\_str() );

}

printf("\n");

return;

}

for( int i = last ; i < n ; ++i ){

if( !seen[ i ] ){

seen[ i ] = 1;

ans[ len ] = i;

solve( len + 1 , i );

seen[ i ] = 0;

}

}

}

int main(){

int t , p;

char s1[ 25 ][ 25 ], s2[ 25 ][ 25 ] , s[ 25 ];

scanf("%d" , &t );

for( int q = 1 ; q <= t ; ++q ){

scanf("%d %d %d" , &n , &p , &sz );

for( int i = 0 ; i < n ; ++i ) {

scanf("%s" , &s );

name[ i ] = "";

for( int j = 0 ; s[ j ] ; ++j ) name[ i ]+= toUpperCase( s[ j ] );

}

for( int i = 0 ; i < p ; ++i ){

scanf("%s %s" , &s1[ i ] , &s2[ i ] );

for( int j = 0 ; j < s1[ i ][ j ] ; ++j ) s1[ i ][ j ] = toUpperCase( s1[ i ][ j ] );

for( int j = 0 ; j < s2[ i ][ j ] ; ++j ) s2[ i ][ j ] = toUpperCase( s2[ i ][ j ]);

}

sort( name , name + n , cmp );

memset( visto , 0 , sizeof( visto ) ) ;

memset( seen , 0 , sizeof( seen ) );

int idx1 = 0 , idx2 = 0;

for( int i = 0 ; i < p ; ++i ){

for( int j = 0 ; j < n ; ++j ){

if( name[ j ] == string( s1[ i ] )){

idx1 = j;

break;

}

}

for( int j = 0 ; j < n ; ++j ){

if( name[ j ] == string( s2[ i ] ) ){

idx2 = j;

break;

}

}

visto[ idx1 ][ idx2 ] = visto[ idx2 ][ idx1 ] = 1; //para prohibiciones

}

printf("Set %d:\n" , q );

solve( 0 , 0 );

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Sabotage

\*\*\*ID: 10480

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, MinCut - Maxflow

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

struct Edge{

int u, v ,c , r;

Edge( int uu , int vv , int cc , int rr ): u(uu), v(vv), c(cc), r(rr){ }

};

vector< Edge > E;

vector< vector<int> > ady;

int V;

int maxflow( int source, int sink ){

for( int flow = 0; ;){

vector<int> prev( ady.size() , -1);

queue<int> Q;

Q.push( source );

prev[ source ] = 0;

while( !Q.empty() ){

int act = Q.front(); Q.pop();

for( int i = 0 ; i < ady[ act ].size(); ++i ){

int e = ady[ act ][ i ];

int v = E[ e ].v;

if( prev[ v ] >= 0 || E[ e ].c == 0 )continue;

prev[ v ] = e;

Q.push( v );

}

}

if( prev[ sink ] < 0 ) return flow;

int c = 0x7ffffff;

for( int x = sink; x != source ; x = E[ prev[ x ] ].u ){

c <?= E[ prev[ x ] ].c;

}

for( int x = sink ; x != source ; x = E[ prev[ x ] ].u ){

E[ prev[ x ] ].c -= c;

E[ E[ prev[ x ] ].r ].c += c;

}

flow += c;

}

}

#define MAX 60

int visited[ MAX ];

void mincut( int source ){

visited[ source ] = 1;

for( int i = 0 ; i < ady[ source ].size(); ++i ){

if( !visited[ E[ ady[ source ][ i ] ].v ] && E[ ady[ source ][ i ] ].c ){

mincut( E[ ady[ source ] [ i ] ].v );

}

}

}

void add( int u, int v , int c ){

Edge e( u , v , c , E.size() + 1 );

ady[ u ].push\_back( E.size() );

E.push\_back( e );

Edge e2( v , u , c , E.size() - 1 );

ady[ v ].push\_back( E.size() );

E.push\_back( e2 );

}

int main(){

int V, A, u , v, c;

while( scanf("%d %d", &V , &A ) && V|A ){

E.clear();

ady = vector< vector<int> > ( V + 1 , vector<int>() );

for( int i = 0 ; i < A ;++i ){

scanf("%d %d %d", &u, &v , &c );

add( u , v , c );

}

int flow = maxflow( 1 , 2 );

memset( visited , 0 , sizeof( visited ) );

mincut( 1 );

for( int i = 1 ; i <= V ;++i ){

for( int j = 0 ; j < ady[ i ].size() ;++j ){

if( E[ ady[ i ][ j ] ].c == 0 && visited[ i ] != visited[ E[ ady[ i ][ j ] ].v ] ){

printf("%d %d\n", i , E[ ady[ i ][ j ] ].v );

}

}

}

printf("\n");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Divisibility of Factors

\*\*\*ID: 10484

\*\*\*Juez: UVA

\*\*\*Tipo: Prime Factorization

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

map< long long , int > getFactors( long long n , bool factorial ){

long long i;

map< long long , int > f;

if( !factorial ){

for( i = 2 ; i \* i <= n ; ++i ){

while( n % i == 0 ){

n /= i;

f[ i ]++;

}

}

if( n > 1 ) f[ n ]++;

}

else{

long long aux = n;

int j;

for( j = 2 ; j <= aux ; ++j ){

n = j;

for( i = 2 ; i \* i <= n ; ++i ){

while( n % i == 0 ){

n /= i;

f[ i ]++;

}

}

if( n > 1 ) f[ n ]++;

}

}

return f;

}

int main(){

int n;

long long d , ans;

map< long long , int > fN , fD;

map< long long , int > :: iterator it;

bool possible;

while( scanf("%d %lld" , &n , &d ) , n + d > 0 ){

fN = getFactors( n , true );

fD = getFactors( d , false );

possible = true;

for( it = fD.begin() ; it != fD.end() ; ++it ){

if( fN[ it -> first ] < it -> second ){ possible = false; break; }

fN[ it -> first ] -= it -> second;

}

if( !possible ) puts("0");

else{

ans = 1;

for( it = fN.begin() ; it != fN.end() ; ++it ) ans \*= ( it -> second + 1 );

printf("%lld\n" , ans );

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Boxes of Chocolates

\*\*\*ID: 10489

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, modulo

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main(){

int t, N , B, K, resp, a, p;

scanf( "%d" ,&t );

while( t-- ){

scanf( "%d %d" ,&N , &B );

resp = 0;

while( B-- ){

scanf( "%d" ,&K );

p = 1;

while( K-- ){

scanf( "%d" ,&a );

p = ( p % N \* a % N )% N;

}

resp = (resp % N + p % N ) % N;

}

printf("%d\n",resp);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Mr. Azad and His Son

\*\*\*ID: 10490

\*\*\*Juez: UVA

\*\*\*Tipo: Primes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

bool isPrime( long long n){

long long i;

if( n == 2 )

return 1;

if( n % 2 == 0)

return 0;

for( i = 3 ; i \* i <= n; i += 2 )

if( n % i == 0 )

return 0;

return 1;

}

int main(){

int n;

long long p;

while( scanf("%d" , &n ) , n ){

bool prime , perfect;

prime = perfect = false;

if( isPrime( ( 1LL<<n ) - 1 ) )

printf("Perfect: %lld!\n" , ( ( 1LL<<( n - 1 ) ) \* ( ( 1LL<<n ) - 1 ) ) );

else if( isPrime( n ) ) puts("Given number is prime. But, NO perfect number is available.");

else puts("Given number is NOT prime! NO perfect number is available.");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Cows and Cars

\*\*\*ID: 10491

\*\*\*Juez: UVA

\*\*\*Tipo: Probability

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main(){

double ncows , ncars , nshow , p1 , p2;

while( scanf("%lf %lf %lf" , &ncows , &ncars , &nshow ) != EOF ){

p1 = ncows/( ncows + ncars ) \* ncars/( ncows + ncars - nshow - 1 );

p2 = ncars/( ncows + ncars ) \* ( ncars - 1 )/( ncows + ncars - nshow - 1 );

printf("%.5lf\n" , p1 + p2 );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Collecting Beepers

\*\*\*ID: 10496

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking , DP + bitwise

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstdio>

#include <cstring>

#include <cstdlib>

#define MAX 30

int min( int a , int b ){return ( a < b )? a : b ;}

struct Point{

int x;

int y;

}V[ MAX ];

int dist[ MAX ][ MAX ], memo[ MAX ][ 1<<11 ] , k , h, w;

///ciudad representada x el bit actual posicion

///mask ciudades visitadas representada x bits 11111 5 ciudades

int solve( int city , int mask ){

if( memo[ city ][ mask ] >= 0 ) return memo[ city ][ mask ];

///si visite todas las ciudades excepto la inicial

if( mask == 1 ) return memo[ city ][ mask ] = dist[ city ][ 0 ];

memo[ city ][ mask ] = 1<<30;

for( int i = 1 ; i < k ; ++i ){

///if no visitado

if( mask & (1<<i) ){

///como mi maskara de no visitados es puro 1111 para marcarlo como visitado

///necesito convertirlo a 0 x ello uso xor

memo[ city ][ mask ] = min( memo[ city ][ mask ] ,

dist[ city ][ i ] + solve( i , mask ^( 1<< i ) ) );

}

}

return memo[ city ][ mask ];

}

int main(){

int t , x , y , p , q;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d %d %d %d" , &h ,&w , &V[ 0 ].x , &V[ 0 ].y , &k );

k++;

for( int i = 1 ; i < k ; ++i ){

scanf("%d %d" , &V[ i ].x , &V[ i ].y );

}

for( int i = 0 ; i < k ; ++i ){

for( int j = 0 ; j < k ; ++j ){

dist[ i ][ j ] = abs( V[ i ].x - V[ j ].x ) + abs( V[ i ].y - V[ j ].y );

}

}

memset( memo , 0xff , sizeof( memo ) );

printf("The shortest path has length %d\n" , solve( 0 , (1<<k) - 1 ) );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The dominoes solitaire

\*\*\*ID: 10503

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

int xini , yini , xfin , yfin, x[ 15 ], y[ 15 ] , m , sz;

bool seen[ 15 ] , finish;

void solve( int len , int yini ){

if( len == sz ){

if( xfin == yini )finish = true;

return;

}

for( int i = 0 ; i < m ; ++i ){

if( !seen[ i ] ){

if( x[ i ] == yini ){

seen[ i ] = 1;

solve( len + 1 , y[ i ] );

seen[ i ] = 0;

}

else if( y[ i ] == yini ) {

seen[ i ] = 1;

solve( len + 1 , x[ i ] );

seen[ i ] = 0;

}

}

}

}

int main(){

while( scanf("%d" , &sz ) , sz ){

scanf("%d%d%d%d%d" , &m , &xini , &yini , &xfin , &yfin );

for( int i = 0 ; i < m ; ++i ) scanf("%d %d" , &x[ i ] , &y[ i ] );

finish = false;

for( int i = 0 ; i < m ; ++i ){

memset( seen , 0 , sizeof( seen ) );

seen[ i ] = 1;

solve( 0 , yini );

if( finish ) break;

}

if( finish ) puts("YES");

else puts("NO");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Montesco vs Capuleto

\*\*\*ID: 10505

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Conectivity, union find

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 255

int V;

vector< int > ady[ MAX ];

int color[ MAX ] , contColor[ 5 ];

int bfs( int inicial ){

int act;

queue< int > Q;

Q.push( inicial );

color[ inicial ] = 1;

memset( contColor , 0 , sizeof( contColor ) );

bool finish = false;

while( !Q.empty() ){

act = Q.front(); Q.pop();

contColor[ color[ act ] ]++;

for( int i = 0 ; i < ady[ act ].size() ; ++i ){

if( !color[ ady[ act ][ i ] ] ){

color[ ady[ act ][ i ] ] = ( color[ act ] == 1 )?2:1;

Q.push( ady[ act ][ i ] );

}

else if( color[ ady[ act ][ i ] ] == color[ act ] ){

finish = true;

}

}

}

if( finish ) return 0;

return max( contColor[ 1 ] , contColor[ 2 ] );

}

int main(){

int t , k , v , ans;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &V );

for( int i = 0 ; i < V && scanf("%d" , &k ) == 1 ; ++i ){

while( k-- ){

scanf("%d" , &v );

if( v - 1 < V ){

ady[ i ].push\_back( v - 1 );

ady[ v - 1 ].push\_back( i );

}

}

}

ans = 0;

memset( color , 0 , sizeof( color ) );

for( int i = 0 ; i < V ; ++i ){

if( color[ i ] ) continue;

ans += bfs( i );

}

printf("%d\n" , ans );

for( int i = 0 ; i <= V ; ++i ){ ady[ i ].clear(); }

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Waking up brain

\*\*\*ID: 10507

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int num;

char mp[ 30 ];

int getId( char c ){

for( int i = 0 ; i < num ; ++i ){

if( c == mp[ i ] )return i;

}

mp[ num ] = c;

return num++;

}

vector<int> ady[ 30 ];

int main(){

char line[ 256 ];

int n , m , x , y, ans, cnt;

char c1 , c2;

bool seen[ 30 ];

vector<int> greedy[ 5 ];

while( scanf("%d" , &n ) == 1 ){

for( int i = 0 ; i < 30 ; ++i ) ady[ i ].clear();

scanf("%d %s" , &m , &line );

num = 0;

memset( seen , 0 , sizeof( seen ) );

for( int i = 0 ; line[ i ] ; ++i ){

seen[ getId( line[ i ] ) ] = 1;

}

if( m == 0 ){

if( num != n ){

puts("THIS BRAIN NEVER WAKES UP");

continue;

}

else{

puts("WAKE UP IN, 0, YEARS");

continue;

}

}

while( m-- ){

scanf("%s" , line );

x = getId( line[ 0 ] ); y = getId( line[ 1 ] );

ady[ x ].push\_back( y );

ady[ y ].push\_back( x );

}

for( int i = 0 ; i < num ; ++i ){

if( seen[ i ] )continue;

cnt = 0;

for( int j = 0 ; j < ady[ i ].size() ; ++j ){

if( seen[ ady[ i ][ j ] ] ) cnt++;

}

if( cnt > 3 ) cnt = 3;

greedy[ cnt ].push\_back( i );

}

ans = 0;

if( greedy[ 3 ].size() == 0 || num != n){

puts("THIS BRAIN NEVER WAKES UP");

continue;

}

for( int i = 3 ; ;){

if( greedy[ i ].size() == 0 )break;

for( int j = 0 ; j < greedy[ i ].size() ; ++j ){

seen[ greedy[ i ][ j ] ] = 1;

}

ans++;

for( int j = 0 ; j < 4 ; ++j ) greedy[ j ].clear();

for( int k = 0 ; k < num ; ++k ){

if( seen[ k ] )continue;

cnt = 0;

for( int j = 0 ; j < ady[ k ].size() ; ++j ){

if( seen[ ady[ k ][ j ] ] ) cnt++;

}

if( cnt > 3 ) cnt = 3;

greedy[ cnt ].push\_back( k );

}

}

int i = 0;

for( ; i < num ; ++i )if( !seen[ i ] )break;

if( i == num )printf("WAKE UP IN, %d, YEARS\n" , ans );

else puts("THIS BRAIN NEVER WAKES UP");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Wavio Sequence

\*\*\*ID: 10534

\*\*\*Juez: UVA

\*\*\*Tipo: DP, LIS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

int n;

int a[ MAX ], lis[ MAX ], incr[ MAX ] , decr[ MAX ] , k;

int solve(){

int len , q ,t , u , v , i;

incr[ 0 ] = k = 1;

lis[ k ] = a[ 0 ];

for( int p = 1 ; p < n ; ++p ){

if( lis[ k ] < a[ p ] ){

incr[ p ] = ++k ;

lis[ incr[ p ] ] = a[ p ];

continue;

}

for( u = 1 , v = k ; u < v ;){

if( lis[ t = ( u + v )>>1 ] < a[ p ] ) u = t + 1; else v = t;

}

incr[ p ] = u;

if( a[ p ] < lis[ u ] ) lis[ u ] = a[ p ];

}

//inverted LIS

decr[ n - 1 ] = k = 1;

lis[ k ] = a[ n - 1 ];

for( int p = n - 2 ; p >= 0 ; --p ){

if( lis[ k ] < a[ p ] ){

decr[ p ] = ++k;

lis[ k ] = a[ p ];

continue;

}

for( u = 1 , v = k ; u < v ; ){

if( lis[ t = ( u + v )>>1 ] < a[ p ] ) u = t + 1; else v = t;

}

decr[ p ] = u;

if( lis[ u ] > a[ p ] ) lis[ u ] = a[ p ];

}

for( len = 1 , i = 0 ; i < n ; ++i ){

u = 2 \* min( incr[ i ] , decr[ i ] ) - 1;

len = ( len < u )? u : len;

cout<<incr[ i ]<<" "<<decr[ i ]<<endl;

}

return len;

}

int main(){

while( scanf("%d" , &n ) != EOF ){

for( int i = 0 ; i < n ; ++i )scanf("%d" , &a[ i ] );

printf("%d\n" , solve() );

}

return 0;

}

/\*

TEST CASES

10

1 1 1 1 5 4 3 2 1 4

6

1 4 2 3 2 1

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Almost Prime Numbers

\*\*\*ID: 10539

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory, sieve, binary search

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

using namespace std;

#define MAX 1000005

int prime[ MAX ];

int primes[ MAX ];

int idx;

void sieve(){

prime[ 0 ] = 0;

for( int i = 1 ; i<= MAX ; ++i ) prime[ i ] = 1;

for( int i = 2 ; i\*i <= MAX ;++i ){

if( prime[ i ] ){

for( int j = i ; j \* i <= MAX ; ++j ){

prime[ j \* i ] = false;

}

}

}

}

long long dp[ MAX ];

void almost(){

idx = 0;

for( int i = 2 ; i <= MAX ; ++i ){

if( prime[ i ] ){

long long ini = i;

for( long long j = ini \* ini ; j <= (1000000000000LL); j \*= ini ){

dp[ idx++ ] = j;

}

}

}

}

int main(){

int t, index, cont;

long long low, high;

sieve();

almost();

sort( dp , dp + idx );

scanf("%d", &t );

while( t-- ){

cont = 0;

scanf("%lld %lld",&low, &high);

index = (int)( lower\_bound( dp , dp + idx , low ) - dp );

for( int i = index ; ; ++i ){

if( dp[ i ] <= high ){

if( dp[ i ] >= low )cont++;

}

else break;

}

printf("%d\n", cont );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Traveling Politician

\*\*\*ID: 10543

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 55

#define INF 1<<30

int V, k;

vector< int > ady[ MAX ];

int dp[ MAX ][ 25 ];

int solve( int city , int speech ){

if( speech > 20 ) return INF;

if( city == V - 1 ){

if( speech >= k ) return 0;

}

if( dp[ city ][ speech ] != -1 ) return dp[ city ][ speech ];

int ans = INF , i , y;

for( i = 0 ; i < ady[ city ].size() ; ++i ){

y = ady[ city ][ i ];

ans = min( ans , 1 + solve( y , speech + 1 ) );

}

return dp[ city ][ speech ] = ans;

}

int main(){

int E , u , v , i , ans;

while( scanf("%d %d %d" , &V , &E , &k ) , V | E | k ){

while( E-- ){

scanf("%d %d" , &u , &v );

if( u != v ) ady[ u ].push\_back( v );

}

memset( dp , -1 , sizeof( dp ) );

ans = solve( 0 , 1 ) + 1;

if( ans > 20 || ans < k ) puts("LOSER");

else printf("%d\n" , ans );

for( i = 0 ; i < V ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Combination Lock

\*\*\*ID: 10550

\*\*\*Juez: UVA

\*\*\*Tipo: Simulation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int main(){

int d , a , b , c , ans, aux;

while( scanf("%d %d %d %d" , &d , &a , &b , &c ) && a|b|c|d ){

ans = 1080;

if( d < a ) ans += ( d + 40 - a ) \* 9;

else ans += ( d - a ) \* 9;

if( b >= a ) ans += ( b - a ) \* 9;

else ans += ( 40 - a + b ) \* 9;

if( b < c ) ans += ( b + 40 - c ) \* 9;

else ans += ( b - c ) \* 9;

printf("%d\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# XYZZY

\*\*\*ID: 10557

\*\*\*Juez: UVA

\*\*\*Tipo: Bellman Ford

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

#define pii pair< int , int >

#define mp make\_pair

#define pb push\_back

int V, cost[ MAX ] , dist[ MAX ] , seen[ MAX ];

vector< int > ady[ MAX ];

//Para conectividad

void dfs( int x ){

seen[ x ] = 1;

for( int i = 0 ; i < ady[ x ].size() ; ++i ){

if( !seen[ ady[ x ][ i ] ] ){

dfs( ady[ x ][ i ] );

}

}

}

void bellmanFord(){

int i , k , j , v;

for( i = 0 ; i < V ; ++i ) dist[ i ] = -1<<30;

dist[ 0 ] = 100;

for( k = 0 ; k < V - 1 ; ++k ){

for( i = 0 ; i < V ; ++i ){

for( j = 0 ; j < ady[ i ].size() ; ++j ){

v = ady[ i ][ j ];

if( dist[ i ] > 0 && dist[ v ] < dist[ i ] + cost[ v ] ){

dist[ v ] = dist[ i ] + cost[ v ];

}

}

}

}

//Verifico ciclo positivo

int node = -1;

bool positive = false;

for( i = 0 ; i < V ; ++i ){

for( j = 0 ; j < ady[ i ].size() ; ++j ){

v = ady[ i ][ j ];

if( dist[ i ] > 0 && dist[ v ] < dist[ i ] + cost[ v ] ){

dist[ v ] = dist[ i ] + cost[ v ];

positive = true;

node = v;

}

}

seen[ i ] = 0;

}

//Si tengo ciclo positivo, veo si a partir de ese ciclo llego al destino

//si es posible entonces winnable

if( positive ){

dfs( node );

if( seen[ V - 1 ] ){

puts("winnable");

return;

}

}

//si llegue al destino con energias gano

if( dist[ V - 1 ] > 0 ) puts("winnable");

else puts("hopeless");

}

int main(){

int i , k , v;

while( scanf("%d" , &V ) , V != -1 ){

for( i = 0 ; i < V && scanf("%d %d" , &cost[ i ] , &k ); ++i ){

while( k-- > 0 && scanf("%d" , &v ) ){

ady[ i ].pb( v - 1 );

}

seen[ i ] = 0;

}

dfs( 0 );

if( !seen[ V - 1 ] ) puts("hopeless");

else bellmanFord();

for( i = 0 ; i < V ; ++i ) ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Geometry Paradox

\*\*\*ID: 10573

\*\*\*Juez: UVA

\*\*\*Tipo: Geometry

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <sstream>

#include <cmath>

using namespace std;

#define MAX 105

const double PI = 2 \* acos(.0);

char s[ MAX ];

int main(){

int q;

scanf("%d" , &q );

gets( s );

double r1 , r2 , t , area , r;

while( q-- && gets( s ) ){

stringstream ss( s );

ss>>r1;

if( !(ss>>r2) ){

t = r1;

area = PI \* ( t \* t / 8.0 );

}

else{

r = r1 + r2;

area = PI \* r \* r - ( PI \* r1 \* r1 + PI \* r2 \* r2 );

}

printf("%.4lf\n" , area );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Y2K Accounting Bug

\*\*\*ID: 10576

\*\*\*Juez: UVA

\*\*\*Tipo: Brute Force

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

long long s ,d , maxi;

int main(){

vector<int> v;

while( scanf("%lld %lld" , &s , &d) != EOF ){

maxi = -1;

for( int i = 0 ; i < 1<<12 ; ++i ){

for( int j = 0 ; j < 12 ; ++j ){

if( i & ( 1<<j) ){

v.push\_back( -d );

}else

v.push\_back( s );

}

int lose = 0;

for( int j = 0 ; j < 12 ; ++j ){

if( j + 4 < 12 ){

int sum = 0;

for( int k = 0 ; k < 5 ; ++k ){

sum += v[j + k];

}

if( sum < 0 )

lose++;

}

}

if( lose == 8 ){

int sum = 0;

for( int j = 0 ; j < 12 ; ++j ){

sum += v[ j ];

}

if( maxi < sum ){

maxi = sum;

}

}

v.clear();

}

if( maxi < 0 )

puts("Deficit");

else

printf("%lld\n" , maxi );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Center of symmetry

\*\*\*ID: 10585

\*\*\*Juez: UVA

\*\*\*Tipo: Sorting

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define EPS 1e-9

const long double PI = 2 \* acos(.0);

#define MAX 10005

struct Point{

double x , y;

Point( double X , double Y ):x(X) , y(Y){}

bool operator <( Point other ) const{

if( fabs( x - other.x ) < EPS ){

return y + EPS < other.y;

}

return x + EPS < other.x;

}

bool operator ==( Point other )const{

return ( fabs( x - other.x ) < EPS && fabs( y - other.y ) < EPS );

}

Point operator -(Point other)const{

return Point( x - other.x , y - other.y );

}

Point(){}

}p[ MAX ];

int main(){

int t , n;

scanf("%d" ,&t );

while( t-- > 0 && scanf("%d" , &n ) ){

for( int i = 0 ; i < n && scanf("%lf %lf" , &p[ i ].x , &p[ i ].y ); ++i );

sort( p , p + n );

bool b = false , error =false;

Point mid , mid2;

for( int i = 0 ; i < n/2 ; ++i ){

if( !b ){

mid = Point((p[ i ].x + p[ n - 1 - i ].x)/2.0,(p[ i ].y + p[ n - 1 - i ].y)/2.0);

b = true;

}

else{

mid2 = Point((p[ i ].x + p[ n - 1 - i ].x)/2.0,(p[ i ].y + p[ n - 1 - i ].y)/2.0);

if( mid == mid2 ) continue;

else{

error = true;

break;

}

}

}

if( !error ) puts("yes");

else puts("no");

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Area

\*\*\*ID: 10589

\*\*\*Juez: UVA

\*\*\*Tipo: Geometry, point inside circle

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cmath>

#define EPS 1e-15

const long double PI = 2 \* acos(.0);

struct Point{

double x , y;

Point( double X , double Y ):x(X) , y(Y){}

bool operator <( Point other ) const{

if( fabs( x - other.x ) < EPS ){

return y + EPS < other.y;

}

return x + EPS < other.x;

}

bool operator ==( Point other )const{

return ( fabs( x - other.x ) < EPS && fabs( y - other.y ) < EPS );

}

Point operator -(Point other)const{

return Point( x - other.x , y - other.y );

}

Point operator + (Point other)const{

return Point( x + other.x , y + other.y );

}

Point(){}

};

double dist( Point p1 , Point p2 ){

return hypot( p1.x - p2.x , p1.y - p2.y );

}

int insideCircle( Point p , Point c , double r ){

double d = dist( p , c );

if( d + EPS < r ) return 1;

if( fabs( d - r ) < EPS ) return 2;

return 0;

}

int main(){

int n , cnt , radius;

double y , x;

while( scanf("%d %d", &n , &radius ) , n ){

cnt = 0;

for( int i = 0 ; i < n && scanf("%lf %lf" , &x , &y ) ; ++i ){

if( insideCircle( Point( x , y ) , Point( 0 , 0 ) , radius ) > 0 &&

insideCircle( Point( x , y ) , Point( radius , 0 ) , radius ) > 0 &&

insideCircle( Point( x , y ) , Point( 0 , radius ) , radius ) > 0 &&

insideCircle( Point( x , y ) , Point( radius , radius ) , radius ) > 0 ){

cnt++;

}

}

printf("%.5lf\n" , (double)(cnt \* radius \* radius)/(double)n);

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Happy Number

\*\*\*ID: 10591

\*\*\*Juez: UVA

\*\*\*Tipo: Cycle Finding

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int f( int n ){

int sum = 0;

while( n > 0){

sum += ( n % 10 )\* ( n %10 );

n /= 10;

}

return sum;

}

void TortoiseHare(int x){

int s = 0, p = 1;

int i = f(x), j = f(f(x));

while (i != j) i = f(i), j = f(f(j));

j = i; i = x;

while (i != j) i = f(i), j = f(j), s++;

j = f(i);

while (i != j) j = f(j), p++;

if( p == 1 )printf("%d is a Happy number.\n",x);

else printf("%d is an Unhappy number.\n",x);

}

int main(){

int n;

int t;

scanf("%d", &t );

for( int i = 1; i <= t; ++i ){

scanf("%d", &n);

printf("Case #%d: ", i);

TortoiseHare( n );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Morning Walk

\*\*\*ID: 10596

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Euler Tour

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 205

int V;

int main(){

int E, u , v;

int degree[ MAX ];

while( scanf("%d %d", &V ,&E) == 2 ){

memset( degree , 0 , sizeof( degree ) );

while( E-- ){

scanf("%d %d", &u ,&v );

degree[ u ]++; degree[ v ]++;

}

bool possible = true;

for( int i = 0 ; i < V ;++i ){

if( degree[ i ] & 1 || degree[ 0 ] == 0){

possible = false;

break;

}

}

if( !possible )printf("Not Possible\n");

else{

printf("Possible\n");

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# ACM contest and Blackout

\*\*\*ID: 10600

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, MST, second best MST

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX\_SIZE 10001

struct Arista{

int origen;

int destino;

int peso;

int mst;

//cambiar a > para arbol de recubrimiento de maxima suma

bool operator<(const Arista &a) const{

return peso<a.peso;

}

}v[ MAX\_SIZE ];

///UNION-FIND

int parent[MAX\_SIZE],rank[MAX\_SIZE];

void Make\_Set(const int x){

parent[x]=x;

rank[x]=0;

}

int Find(const int x){

if(parent[x]!=x) parent[x]=Find(parent[x]);

return parent[x];

}

void Union(const int PX, const int PY){

if(rank[PX]>rank[PY]) parent[PY]=PX;

else{

parent[PX]=PY;

if(rank[PX]==rank[PY]) rank[PY]++;

}

}

void Merge(const int x,const int y){

Union(Find(x),Find(y));

}

///END UNION FIND

int main(){

int t , V , E, first , second, aux , size;

scanf("%d" ,&t );

while( t-- ){

scanf( "%d %d" , &V, &E );

for( int i = 0 ; i < E ; ++i ){

scanf("%d %d %d" , &v[ i ].origen , &v[ i ].destino , &v[ i ].peso );

}

sort( v , v + E );

for( int i = 1 ; i <= V ; ++i )Make\_Set( i );

first = 0;

for( int i = 0 ; i < E ; ++i ){

int u = v[ i ].origen , vv = v[ i ].destino , w = v[ i ].peso;

if( Find( u ) != Find( vv ) ){

v[ i ].mst = true;

first += w;

Merge( u , vv );

}

else v[ i ].mst = false;

}

second = INT\_MAX;

///vemos aristas del MST y probamos nuevo MST eliminando una de dichas aristas

for( int j = 0 ; j < E ; ++j ){

if( !v[ j ].mst ) continue;

aux = 0; size = 0;

for( int i = 1 ; i <= V ; ++i ) Make\_Set( i );

for( int i = 0 ; i < E ; ++i ){

if( i == j )continue;

int u = v[ i ].origen , vv = v[ i ].destino , w = v[ i ].peso;

if( Find( u ) != Find( vv ) ){

aux += w;

size++;

Merge( u , vv );

}

}

///si tengo todos los nodos, puede ser q la arista q halla eliminado sea la unica

///coneccion a ese nodo

if( size == V - 1 ) second = min( second , aux );

}

printf("%d %d\n" , first , second );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Friends

\*\*\*ID: 10608

\*\*\*Juez: UVA

\*\*\*Tipo: Union Find

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <cstring>

#define MAX 30005

int parent[ MAX ] , V;

void MakeSet(){

for( int i = 0 ; i < V ; ++i ) parent[ i ] = i;

}

int Find( int x ){

return ( parent[ x ] == x )? x : parent[ x ] = Find( parent[ x ] );

}

void Union( int x , int y ){

parent[ Find( x ) ] = Find( y );

}

int numElem[ MAX ];

int main(){

int t , E , x , y, ans;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &V , &E );

MakeSet();

while( E-- ){

scanf("%d %d" , &x , &y );

Union( x - 1 , y - 1 );

}

//Cuento numero de elementos por conjunto

memset( numElem , 0 , sizeof( numElem ) );

for( int i = 0 ; i < V ; ++i ){

numElem[ Find( i ) ]++;

}

//Hallo el maximo

ans = 0;

for( int i = 0 ; i < V ; ++i ) ans = ( ans < numElem[ i ] )?numElem[ i ]: ans;

printf("%d\n" , ans );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Gopher and Hawks

\*\*\*ID: 10610

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, BFS, Shortest Path

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define mp make\_pair

#define f first

#define s second

#define MAXN 10005

#define MAX 10005

vector<int> ady[ MAX ];

int Vs,T;

double xfin , yfin;

struct Estado{

int x, d;

Estado( int xx ,int dd):x(xx) , d(dd){}

};

int bfs(){

bool seen[ MAX ];

memset( seen , 0 , sizeof( seen ) );

queue<Estado> Q;

Q.push( Estado( 0 , 0 ) );

while( !Q.empty() ){

Estado act = Q.front(); Q.pop();

if( act.x == 1 ){

return act.d;

}

for( int i = 0 ; i < ady[ act.x ].size(); ++i ){

int w = ady[ act.x ][ i ];

if( !seen[ w ] ){

seen[ w ] = true;

Q.push( Estado( w , act.d + 1 ) );

}

}

}

return -1;

}

int main(){

char s[ MAXN ];

double x , y;

while( scanf("%d %d", &Vs ,&T ) && Vs|T ){

vector< pair<double , double> > v;

scanf("%lf %lf", &x ,&y );

v.push\_back( mp( x , y ) );

scanf("%lf %lf", &xfin ,&yfin );

v.push\_back( mp( xfin , yfin ) );

gets( s );

while( gets( s ) && strlen(s) > 0 ){

sscanf(s , "%lf %lf", &x , &y);

v.push\_back( mp( x , y ) );

}

for( int i = 0 ; i < v.size(); ++i ){

for( int j = i + 1; j < v.size(); ++j ){

double dist = ( v[ i ].f - v[ j ].f )\*( v[ i ].f - v[ j ].f ) + ( v[ i ].s - v[ j ].s )\*( v[ i ].s - v[ j ].s );

dist = sqrt( dist );

if( dist <= T\* 60.0 \* Vs ){

ady[ i ].push\_back( j );

ady[ j ].push\_back( i );

}

}

}

T = bfs();

if( T == -1 )printf("No.\n");

else printf("Yes, visiting %d other holes.\n" , T - 1);

for( int i = 0 ; i < MAX ; ++i )ady[ i ].clear();

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# The Playboy Chimp

\*\*\*ID: 10611

\*\*\*Juez: UVA

\*\*\*Tipo: Binary Search

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <algorithm>

using namespace std;

#define MAX 50005

int a[ MAX ];

int main(){

int n , q , h , l , u;

scanf("%d" , &n );

for( int i = 0 ; i < n ; ++i ) scanf("%d" , &a[ i ] );

scanf("%d" , &q );

while( q-- ){

scanf("%d" , &h );

l = lower\_bound( a , a + n , h ) - a - 1;

u = upper\_bound( a , a + n , h ) - a;

if( l < 0 ) putchar('X');

else printf("%d" , a[ l ] );

printf(" ");

if( u == n ) puts("X");

else printf("%d\n" , a[ u ] );

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Divisible Group Sums

\*\*\*ID: 10616

\*\*\*Juez: UVA

\*\*\*Tipo: DP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 205

long long a[ MAX ];

int d , m , n;

long long dp[ 25 ][ MAX ][ MAX ][ 2 ];

long long solve( int sum , int len , int last ){

if( len == m ){

if( sum % d == 0 ) return 1;

return 0;

}

if( sum < 0 ){

if( dp[ -sum ][ len ][ last ][ 1 ] != -1 ) return dp[ -sum ][ len ][ last ][ 1 ];

}

else if( dp[ sum ][ len ][ last ][ 0 ] != -1 ) return dp[ sum ][ len ][ last ][ 0 ];

long long ans = 0;

for( int i = last ; i < n ; ++i ){

ans += solve( ( sum % d + a[ i ] % d ) %d , len + 1 , i + 1 );

}

if( sum < 0 ) return dp[ -sum ][ len ][ last ][ 1 ] = ans;

else return dp[ sum ][ len ][ last ][ 0 ] = ans;

}

int main(){

int q , t = 1;

while( scanf("%d %d" , &n , &q ) , n | q ){

for( int i = 0 ; i < n ; ++i ) scanf("%lld" , &a[ i ] );

printf("SET %d:\n" , t++ );

for( int i = 1 ; i <= q ; ++i ){

scanf("%d %d" , &d , &m );

memset( dp , -1LL , sizeof( dp ) );

printf("QUERY %d: %lld\n" , i , solve( 0 , 0 , 0 ) );

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Prince and Princess

\*\*\*ID: 10635

\*\*\*Juez: UVA

\*\*\*Tipo: Hallar LCS x medio de LIS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 62505

int a[ MAX ], lis[ MAX ];

int n , p , q , k, j , i, len, t;

int LIS(){

lis[ len = 1 ] = a[ 0 ];

for( p = 1 ; p < k ; ++p ){

if( lis[ len ] < a[ p ] ){

lis[ ++len ] = a[ p ];

continue;

}

for( i = 1 , j = len ; i < j ; )

if( lis[ t = ( i + j ) >> 1 ] < a[ p ] ) i = t + 1; else j = t;

lis[ i ] = a[ p ];

}

for( i = 0 ; i < k ; ++i ){

cout<<a[ i ]<<" b "<<lis[ i ]<<endl;

}

return len;

}

int main(){

int t, seen[ MAX ] , index[ MAX ] , x , qq;

scanf("%d" , &t );

memset( seen , 0 , sizeof( seen ) );

for( qq = 1 ; qq <= t ; ++qq ){

scanf("%d %d %d", &n , &p , &q );

for( i = 0 ; i <= p ; ++i ){

scanf("%d" , &x );

seen[ x ] = qq;

index[ x ] = i;

}

for( j = 0, k = 0 ; j <= q ; ++j ){

scanf("%d" , &x );

if( seen[ x ] == qq ){

a[ k++ ] = index[ x ];

}

}

printf("Case %d: ", qq );

if( k == 0 )printf("0\n");

else printf("%d\n", LIS() );

}

return 0;

}