Tabla de contenido

[1. What is the Card? 2](#_Toc11690245)

[2. Determinate Prime 2](#_Toc11690246)

[3. Maximum Sum II 3](#_Toc11690247)

[4. The Wedding 3](#_Toc11690248)

[5. Luggage 4](#_Toc11690249)

[6. Largest Block 4](#_Toc11690250)

[7. Work Reduction 5](#_Toc11690251)

[8. Play with Floor and Ceil 5](#_Toc11690252)

[9. Base Equality 6](#_Toc11690253)

[10. The Grazing Cow 6](#_Toc11690254)

[11. LCM 6](#_Toc11690255)

[12. Teobaldo's Trip 7](#_Toc11690256)

[13. The jackpot 8](#_Toc11690257)

[14. Nature 8](#_Toc11690258)

[15. SQF Problems 9](#_Toc11690259)

[16. Yet another Number Sequence 9](#_Toc11690260)

[17. Pre in and post 10](#_Toc11690261)

[18. Travelling Salesman 10](#_Toc11690262)

[19. Free Spots 10](#_Toc11690263)

[20. Number Sequence 11](#_Toc11690264)

[21. Count the Numbers 11](#_Toc11690265)

[22. Graph Construction 12](#_Toc11690266)

[23. Bar Codes 12](#_Toc11690267)

[24. Road Construction 13](#_Toc11690268)

[25. Test 14](#_Toc11690269)

[26. Euler Circuit 15](#_Toc11690270)

[27. String to Palindrome 16](#_Toc11690271)

[28. New Rule in Euphomia 17](#_Toc11690272)

[29. Dominant Strings 17](#_Toc11690273)

[30. Crime Wave -The Sequel 18](#_Toc11690274)

[31. Maximum Subsequence 19](#_Toc11690275)

[32. Back to Intermediate Math 20](#_Toc11690276)

[33. Collector's Problem 21](#_Toc11690277)

[34. Again Prime! No Time 21](#_Toc11690278)

[35. Diagonal 22](#_Toc11690279)

[36. The Mad Numerologist 22](#_Toc11690280)

[37. Minimum Sum LCM 23](#_Toc11690281)

[38. The Orc Attack 23](#_Toc11690282)

[39. Not That Kind of Graph 24](#_Toc11690283)

[40. Thunder Mountain 25](#_Toc11690284)

[41. Dijkstra, Dijkstra. 25](#_Toc11690285)

[42. Traditional BINGO 26](#_Toc11690286)

[43. Andy's First Dictionary 27](#_Toc11690287)

[44. Trouble of 13-Dots 27](#_Toc11690288)

[45. Maximum sum on a torus 28](#_Toc11690289)

[46. Traffic Flow 28](#_Toc11690290)

[47. Move the bishop 29](#_Toc11690291)

[48. 2D Hieroglyphs decoder 30](#_Toc11690292)

[49. Less Prime 30](#_Toc11690293)

[50. Rotated Squares 30](#_Toc11690294)

[51. Unique Factorization 31](#_Toc11690295)

[52. Brownie Points I 32](#_Toc11690296)

[53. Recurrences 32](#_Toc11690297)

[54. Decode the tape 33](#_Toc11690298)

[55. Code Refactoring 33](#_Toc11690299)

[56. Colin and Ryan 33](#_Toc11690300)

[57. Matrix Transpose 34](#_Toc11690301)

[58. Ferry Loading III 34](#_Toc11690302)

[59. Marks Distribution 35](#_Toc11690303)

[60. Forming Quiz Teams 35](#_Toc11690304)

[61. Simple Minded Hashing 36](#_Toc11690305)

[62. Walking on a Grid 36](#_Toc11690306)

[63. Factstone Benchmark 37](#_Toc11690307)

[64. 2 the 9s 37](#_Toc11690308)

[65. My Dear Neighbours 38](#_Toc11690309)

[66. A-Sequence 38](#_Toc11690310)

[67. Throwing cards away I 38](#_Toc11690311)

[68. Flea Circus 39](#_Toc11690312)

[69. How do you add? 40](#_Toc11690313)

[70. Mother Bear 40](#_Toc11690314)

[71. Bad Code 41](#_Toc11690315)

[72. Add All 41](#_Toc11690316)

[73. The Party , Part 1 42](#_Toc11690317)

[74. Triangle Counting 42](#_Toc11690318)

[75. Enchanted Forest 43](#_Toc11690319)

[76. Let's Play Magic! 43](#_Toc11690320)

[77. Ignoring Digits 44](#_Toc11690321)

[78. Simple Addition 44](#_Toc11690322)

[79. Cheapest Base 45](#_Toc11690323)

[80. Ferry Loading IV 45](#_Toc11690324)

[81. Building designing 46](#_Toc11690325)

[82. Add bricks in the wall 46](#_Toc11690326)

[83. Searching for Nessy 46](#_Toc11690327)

[84. My T-shirt suits me 47](#_Toc11690328)

[85. The Scrooge Co Problem 47](#_Toc11690329)

[86. Automatic correction of misspellings 48](#_Toc11690330)

[87. Wine Trading in Gergovia 49](#_Toc11690331)

[88. Formula 1 49](#_Toc11690332)

[89. Exact Sum 50](#_Toc11690333)

[90. Beverages 50](#_Toc11690334)

[91. Andy's Second Dictionary 51](#_Toc11690335)

[92. B2-Sequence 51](#_Toc11690336)

[93. Number Theory 52](#_Toc11690337)

[94. Little Red Riding Hood 52](#_Toc11690338)

[95. A graph Problem 53](#_Toc11690339)

[96. Open Credit System 53](#_Toc11690340)

[97. Place the Guards 53](#_Toc11690341)

[98. Back to the 8-Queens 54](#_Toc11690342)

[99. Composite Prime 54](#_Toc11690343)

[100. Continents 55](#_Toc11690344)

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

# What is the Card?

\*\*\*ID: 10646

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc - Games

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t, aux, Y , X , top;

scanf("%d" , &t );

char s[ 3 ];

for( int q = 1 ; q <= t ; ++q ){

string deck[ 53 ];

for( int i = 1 ; i <= 52 ; ++i ){

scanf("%s" , &s );

deck[ i ] = string( s );

}

aux = 3;

top = 27;

Y = 0;

while( aux-- ){

X = 10;

if( '0'<= deck[ top ][ 0 ] && deck[ top ][ 0 ] <= '9' ) X = deck[ top ][ 0 ] - '0';

Y += X;

top--;

top -= ( 10 - X );

}

if( Y >= top ) Y = 27 + ( Y - top );

printf("Case %d: %s\n" , q , deck[ Y ].c\_str() );

}

return 0;

}

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

# Determinate Prime

\*\*\*ID: 10650

\*\*\*Juez: TJU

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

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

#include <stdio.h>

#define MAX 32100

int prime[MAX],p[MAX];

void sieve(){

for(int i = 0 ; i<MAX; i++)p[i]=1;

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

if(p[i]){

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

p[j\*i] = 0;

}

}

}

int id=0;

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

if(p[i])prime[id++]=i;

}

}

int main(){

sieve();

int L,U,idx, cont,index,ant,act,dist;

while(scanf("%d %d",&L,&U)){

if(L == 0 && U == 0)break;

int tmp = L;

if(U < L){

L = U;

U = tmp;

}

idx=0;

while(prime[idx]<L){

idx++;

}

//arreglo de progresiones

int a[MAX];

//anterior de limite para verificar si anterior de L esta en la secuencia

idx--;

ant = prime[idx++]; act = prime[idx++]; dist = act-ant;

a[0] = ant; a[1] = act;

ant = act;

cont = 2;

index = 2;

// U+50 para ver el limite superior si hay algun primo q esta en la secuencia y mayor a U

for(int i = prime[idx]; i <= U + 50 ; i = prime[++idx]){

if(i - ant == dist){

a[index++]=i;

cont++;

}

else {

if( cont > 2 ){

if( a[0] >= L && a[index-1] <= U){

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

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

printf(" %d" , a[j]);

}

printf("\n");

}

}

cont=2; a[0]=ant; a[1]=i; dist= i - ant; index = 2; //inicializo todo

}

ant=i;

}

}

return 0;

}

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

# Maximum Sum II

\*\*\*ID: 10656

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

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

#include <stdio.h>

int main(){

int a, n, k;

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

for( k = 0 ; n-- > 0 && scanf( "%d", &a ) == 1 ; )

if ( a > 0 )

printf( k++ ? " %d" : "%d" , a );

printf( ( !k ) ? "0\n" : "\n");

}

return 0;

}

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

# The Wedding

\*\*\*ID: 10662

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, cycle detection

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

#include <cstdio>

#include <cstring>

int cost[ 3 ][ 32 ];

int adj[ 3 ][ 32 ][ 32 ];

int N, M, K;

int main()

{

while ( scanf("%d %d %d" , &N , &M , &K ) == 3 ) {

memset( adj , 0 , sizeof( adj ) );

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

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

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

scanf("%d" , &adj[ 0 ][ i ][ j ] );

}

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

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

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

scanf("%d", &adj[ 1 ][ i ][ j ] );

}

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

scanf("%d" , &cost[ 2 ][ i ] );

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

scanf("%d" , &adj[ 2 ][ i ][ j ]);

}

int bestCost = 0x7fffffff;

int best[3]={ -1 , -1 , -1 };

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

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

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

int c = cost[ 0 ][ i ] + cost[ 1 ][ j ] + cost[ 2 ][ k ];

if ( c >= bestCost ) continue;

if ( adj[ 0 ][ i ][ j ] ) continue;

if ( adj[ 1 ][ j ][ k ] ) continue;

if ( adj[ 2 ][ k ][ i ] ) continue;

bestCost = c;

best[ 0 ] = i;

best[ 1 ] = j;

best[ 2 ] = k;

}

}

}

if ( best[ 0 ] < 0) {

printf("Don't get married!\n");

} else {

printf("%d %d %d:%d\n", best[ 0 ] , best[ 1 ] , best[ 2 ] , bestCost );

}

}

}

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

# Luggage

\*\*\*ID: 10664

\*\*\*Juez: UVA

\*\*\*Tipo: DP + bitmasks || just DP

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 25

#define MAXN 105

char s[ MAXN ];

int a[ MAX ], n, val;

int dp[ ( 1<<20 ) + 10 ];

bool solve( int sum , int mask , int len ){

if( len >= n || sum > val ) return false;

if( sum == val ){

return true;

}

if( dp[ mask ] != -1 ) return dp[ mask ];

bool b = false;

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

if( !( mask & 1<<i ) ){

b |= solve( sum + a[ i ] , mask | 1<<i , len + 1 );

}

}

return dp[ mask ] = b;

}

int main(){

int t , suma;

scanf("%d" , &t );

getchar();

while( t-- ){

gets( s );

stringstream ss( s );

n = 0;

suma = 0;

while( ss>>a[ n ] ){

suma += a[ n ];

n++;

}

val = suma/2;

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

if( !( suma & 1 ) && solve( a[ 0 ] , 1 , 1 ) ) puts("YES");

else puts("NO");

}

return 0;

}

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

# Largest Block

\*\*\*ID: 10667

\*\*\*Juez: UVA

\*\*\*Tipo: DP, Maximum 2D sum

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#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 < 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 , b , x1 , y1 , x2 , y2;

scanf("%d" , &t );

while( t-- ){

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

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

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

a[ i ][ j ] = 1;

}

}

while( b-- ){

scanf("%d %d %d %d" , &x1 , &y1 , &x2 , &y2 );

x1--; y1--; x2--; y2--;

for( int i = x1 ; i <= x2 ; ++i ){

for( int j = y1 ; j <= y2 ; ++j ){

a[ i ][ j ] = -10050;

}

}

}

printf("%d\n" , Kadane2D() );

}

return 0;

}

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

# Work Reduction

\*\*\*ID: 10670

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t ,N , M , k ,A, B , n , l , j;

char s[ 50 ];

long long resp;

string cad;

scanf("%d" , &t );

for( int q = 1 ; q <= t ; ++q ){

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

vector<pair<long long , string> > v;

gets( s );

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

gets( s );

l = strlen( s );

cad = "";

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

if( s[ j ] == ':' || s[ j ] ==',')cad += " ";

else cad += s[ j ];

}

cad += " ";

stringstream ss( cad );

ss>>cad>>A>>B;

resp = 0;

n = N;

while( n > M ){

int half = n/2;

if( half < M ){

resp += ( n - M )\* A;

n = M;

}

else{

if( B <= ( n - half ) \* A ){

resp += B;

}

else resp += ( n - half ) \* A;

n = half;

}

}

v.push\_back( make\_pair( resp , string( cad ) ) );

}

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

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

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

printf("%s %lld\n" , v[ i ].second.c\_str() , v[ i ].first );

}

}

return 0;

}

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

# Play with Floor and Ceil

\*\*\*ID: 10673

\*\*\*Juez: UVA

\*\*\*Tipo: Extended Euclid Algorithm

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

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; }

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

y -= a/b \*x;

return d;

}

int main(){

int t;

scanf("%d" , &t );

long long x , k ,a , b , y , p ,q;

while( t-- > 0 && scanf("%lld %lld" , &x , &k ) ){

a = ( long long )floor( x/(double)k );

b = ( long long )ceil( x/(double)k );

x /= extended\_euclid( a , b , p , q );

printf("%lld %lld\n" , p \* x , q \* x );

}

return 0;

}

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

# Base Equality

\*\*\*ID: 10677

\*\*\*Juez: UVA

\*\*\*Tipo: Math

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int pot( int a , int b ){

int r = 1;

for( int i = 1 ; i <= b ; ++i ) r \*= a;

return r;

}

///convierte de base "base" a base 10

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){string s;ostringstream buffer;buffer<<n;s=buffer.str();return s;}

int toInt(string str){int n;istringstream buffer(str);buffer>>n;return n;}

string toBaseN( int n , int base ){

string s = "" , dig;

while( n > 0 ){

dig = "";

if( n % base > 9 ) dig += (char)( ( n % base - 10 ) + 'A');

else dig +=( char ) ( ( n % base ) + '0' );

s = dig + s;

n /= base;

}

return s;

}

int main(){

int t , b1 , i , b2 , r1 , r2, n1 , n2;

string s;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d %d %d" , &b1 ,&b2 , &r1 , &r2 );

for( i = r2 - 1 ; i > r1 ; --i ){

n2 = toBase10( toBaseN( i , b1 ) , b2 );

if( n2 % i == 0 ){

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

break;

}

}

if( i == r1 )puts("Non-existent.");

}

return 0;

}

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

# The Grazing Cow

\*\*\*ID: 10678

\*\*\*Juez: UVA

\*\*\*Tipo: Area Ellipsis

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

#include <stdio.h>

#include <cmath>

const double PI = 2 \* acos( .0 );

int main(){

int t;

scanf("%d" , &t );

double D , L;

while( t-- > 0 && scanf("%lf %lf" , &D , &L ) ){

printf("%.3lf\n" , PI \* L/2.0 \* sqrt( (L \* L/4.0 ) - (D \* D /4.0) ) );

}

return 0;

}

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

# LCM

\*\*\*ID: 10680

\*\*\*Juez: UVA

\*\*\*Tipo: Prime Factorization

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1000005

int prime[ MAX ] , p[ MAX ] , len;

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 = 2 ; i < MAX ; ++i ) if( prime[ i ] ) p[ len++ ] = i;

}

int main(){

sieve();

int n , ceros , dos , cinco, aux , i;

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

dos = 0;

aux = n;

while( aux > 1 ){

dos++;

aux /= 2;

}

cinco = 0;

aux = n;

while( aux > 4 ){

cinco++;

aux /= 5;

}

//ceros al final de lcm

ceros = min( dos , cinco ) + 1;

long long val = 1 , primo;

for( i = 0 ; i < ceros ; ++i ) val \*= 10;

int ans = 1 , cnt;

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

if( p[ i ] > n ) break;

primo = 1;

//primos con potencia mas alta

while( primo \* p[ i ] <= n ){

primo \*= p[ i ];

}

ans = (ans % val \* primo % val ) % val;

}

while( ans > 0 && ans % 10 == 0 ) ans /= 10;

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

}

return 0;

}

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

# Teobaldo's Trip

\*\*\*ID: 10681

\*\*\*Juez: UVA

\*\*\*Tipo: Matrix Exponentiation

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 101

typedef int type;

struct Matrix{

type mat[ MAX ][ MAX ];

};

int n;

Matrix MatrixMult( Matrix A , Matrix B ){

Matrix ans; int i , j , k;

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

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

for( ans.mat[ i ][ j ] = k = 0 ; k < n ; ++k )

ans.mat[ i ][ j ] |= A.mat[ i ][ k ] & B.mat[ k ][ j ];

return ans;

}

void print( Matrix m ){

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

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

cout<<m.mat[ i ][ j ]<<" ";

}

cout<<endl;

}

}

Matrix MatrixPow( Matrix A , int p ){

int i , j;

Matrix ans;

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

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

ans.mat[ i ][ j ] = ( i == j );

while( p ){

if( p & 1 ) ans = MatrixMult( ans , A );

A = MatrixMult( A , A );

p >>= 1;

}

return ans;

}

int main(){

int E , u , v , l;

Matrix ady;

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

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

while( E-- > 0 && scanf("%d %d" , &u , &v ) ){

ady.mat[ u - 1 ][ v - 1 ] = ady.mat[ v - 1 ][ u - 1 ] = 1;

}

scanf("%d %d %d" , &u , &v , &l );

ady = MatrixPow( ady , l );

if( ady.mat[ u - 1 ][ v - 1 ] > 0 ) puts("Yes, Teobaldo can travel.");

else puts("No, Teobaldo can not travel.");

}

return 0;

}

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

# The jackpot

\*\*\*ID: 10684

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <stdio.h>

int main(){

int n, max, x , sum;

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

sum = 0;

max = 0;

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

scanf("%d",&x);

sum+=x;

if( sum > max) max = sum;

if( sum < 0) sum = 0;

}

if( max > 0)printf("The maximum winning streak is %d.\n",max);

else printf("Losing streak.\n");

}

return 0;

}

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

# Nature

\*\*\*ID: 10685

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Union Find

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX\_SIZE 5005

///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 V, E, len;

int resp[ MAX\_SIZE ];

char s[ 40 ], s1[ 40 ];

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

map<string , int > mp;

len = 0;

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

scanf("%s" , s );

if( mp.find( s ) == mp.end() ){

mp[ s ] = len++;

}

}

for( int i = 0 ; i < len ; ++i )Make\_Set( i );

while( E-- ){

scanf("%s %s" , s , s1 );

int a = mp[ s ] , b = mp[ s1 ];

if( Find( a ) != Find( b ) ){

Merge( a , b );

}

}

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

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

resp[ Find( i ) ]++;

}

int ans = 0;

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

if( resp[ i ] > ans )ans = resp[ i ];

}

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

}

return 0;

}

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

# SQF Problems

\*\*\*ID: 10686

\*\*\*Juez: UVA

\*\*\*Tipo: Data Structure, Map

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 21

int main(){

int t , n , m , c, cont[ MAX ], need[ MAX ];

scanf("%d" , &t );

char s[ 205 ];

map<int, string> parse

map<string , bool > seen; //descripcion -> 0|1

map<string , set<int> > into;

while( t-- ){

seen.clear();

parse.clear();

into.clear();

memset( cont , 0 , sizeof( cont ) );

scanf("%d" , &n );

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

scanf("%s %d %d" , &s , &m ,&c );

need[ i ] = c;

parse[ i ] = s;

while( m-- ){

scanf("%s" , &s );

into[ s ].insert( i );

}

}

gets( s );

while( gets( s ) ){

int l = strlen( s );

if( l == 0 )break;

s[ l++ ] = ' ';

s[ l ] = '\0';

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

if( !isalpha( s[ i ] ) ){

s[ i ] = ' ';

}

}

char \*p = strtok( s , " ");

while( p != NULL ){

if( !seen[ p ] ){

vector<int> pos( into[ p ].begin() , into[ p ].end() );

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

cont[ pos[ i ] ]++;

}

seen[ p ] = 1;

}

p = strtok( NULL , " " );

}

}

bool first = true;

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

if( cont[ i ] >= need[ i ] ){

if( !first )printf(",");

printf("%s" , parse[ i ].c\_str() );

first = false;

}

}

if( first )printf("SQF Problem.");

printf("\n");

}

return 0;

}

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

# Yet another Number Sequence

\*\*\*ID: 10689

\*\*\*Juez: UVA

\*\*\*Tipo: Fibonacci

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int mod[] = { 1 , 10 , 100 , 1000 , 10000 };

int val[] = { 1 , 60 , 300 , 1500 , 15000 };

int main(){

int t , i , a , b , n , m , c , tmp;

scanf("%d" , &t );

while( t-- > 0 && scanf("%d %d %d %d" , &a , &b , &n , &m ) ){

a %= mod[ m ]; b %= mod[ m ];

n %= val[ m ];

if( n == 0 ){

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

continue;

}

else if( n == 1 ){

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

continue;

}

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

c = ( a + b ) % mod[ m ];

a = b;

b = c;

}

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

}

return 0;

}

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

# Pre in and post

\*\*\*ID: 10701

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Tree Traversal, recursion

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

#include <stdio.h>

#define MAX 55

void Solve( char \*pre , char \*in, int l){

if( l < 1) return;

if( l == 1 ){

putchar( pre[ 0 ] );

return;

}

int i;

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

if( pre[ 0 ] == in[ i ] )break;

}

Solve( pre + 1 , in , i );

Solve( pre + i + 1 , in + i + 1 , l - i - 1);

putchar( pre[ 0 ] );

}

int main(){

int t , l;

scanf("%d", &t );

char pre[ MAX ], in [ MAX ];

while( t-- ){

scanf("%d %s %s", &l , pre, in);

Solve( pre, in , l );

putchar('\n');

}

return 0;

}

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

# Travelling Salesman

\*\*\*ID: 10702

\*\*\*Juez: UVA

\*\*\*Tipo: DPa

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

#include <stdio.h>

#include <string.h>

#define MAX 105

#define INF 1<<30

int max( int a , int b ){ return ( a < b )?b:a; }

int V , S , K, end[ MAX ] , ady[ MAX ][ MAX ] , dp[ MAX ][ 1005 ];

int solve( int cur , int k ){

if( k > K ) return -INF;

if( k == K && !end[ cur ] ) return -INF;

if( end[ cur ] && k == K ) return 0;

if( dp[ cur ][ k ] != -1 ) return dp[ cur ][ k ];

int ans = 0 , i;

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

if( i == cur ) continue;

ans = max( ans , ady[ cur ][ i ] + solve( i , k + 1 ) );

}

return dp[ cur ][ k ] = ans;

}

int main(){

int T , i , j, x;

while( scanf("%d %d %d %d" , &V , &S , &T , &K ) , V|S|T|K){

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

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

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

}

}

memset( end , 0 , sizeof( end ) );

while( T-- ){

scanf("%d" , &x );

end[ x - 1 ] = 1;

}

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

printf("%d\n" , solve( S - 1 , 0 ) );

}

return 0;

}

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

# Free Spots

\*\*\*ID: 10703

\*\*\*Juez: UVA

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

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int w , h , n , cont , x1 , y1 , x2 , y2 , tmpx , tmpy;

while( scanf( "%d %d %d" , &w , &h , &n ) && (w+h+n)){

int ady [ w + 1 ][ h + 1 ];

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

while( n-- ){

scanf( "%d %d %d %d" , &x1 , &y1 , &x2 , &y2 );

if( x1 > x2 && y1<=y2){

tmpx = x1;

x1 = x2;

x2 = tmpx;

}

else if( x1 >= x2 && y1 > y2 ){

tmpx = x1;

x1 = x2;

x2 = tmpx;

tmpy = y1;

y1 = y2;

y2 = tmpy;

}

if( x1 <= x2 && y1 > y2){

tmpy = y1;

y1 = y2;

y2 = tmpy;

}

for( int i = x1 ; i <= x2 ; ++i ){

if( i > w)break;

for ( int j = y1 ; j <= y2 ; ++j ){

if( j > h)break;

ady [ i ][ j ] = 1;

}

}

}

cont = 0;

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

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

if( ady [ i ][ j ] == 0 )cont++;

}

}

if ( cont == 0 )printf("There is no empty spots.\n");

else if( cont == 1 )printf("There is one empty spot.\n");

else printf("There are %d empty spots.\n", cont );

}

return 0;

}

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

# Number Sequence

\*\*\*ID: 10706

\*\*\*Juez: UVA

\*\*\*Tipo: Binary Search

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 65550

long long sum[ MAX ];

int main(){

for( long long i = 1 ; i < MAX ; ++i ){

sum[ i ] = ( i \* ( i + 1 ) )/2LL;

}

int t, idx;

scanf("%d" , &t );

long long val;

while( t-- ){

scanf("%lld" , &val );

idx = lower\_bound( sum , sum + MAX , val ) - sum;

printf("%lld\n" , val - sum[ idx - 1 ] );

}

return 0;

}

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

# Count the Numbers

\*\*\*ID: 10712

\*\*\*Juez: UVA

\*\*\*Tipo: DP

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 11

int digit[ MAX ] , sz , sz2;

void getDigits( long long n ){

sz = 0;

int digitInv[ MAX ] , i , j;

while( n > 0 ){

digitInv[ sz++ ] = n % 10;

n /= 10;

}

for( i = sz - 1 , j = 0 ; i >= 0 ; --i , ++j ){

digit[ j ] = digitInv[ i ];

}

}

char D[ 4 ];

long long dp[ MAX ][ 2 ][ 3 ][ 2 ][ 2 ];

map< long long , bool > seen;

long long solve( int len , int flag , int len2 , bool correct , string s , bool cero , long long num ){

if( len == sz ){

if( correct && !seen[ num ] ){

seen[ num ] = 1;

return 1;

}

return 0;

}

if( dp[ len ][ flag ][ len2 ][ correct ][ cero ] != -1 )

return dp[ len ][ flag ][ len2 ][ correct ][ cero ];

int maxDigit = digit[ len ] , i , aux;

if( flag ) maxDigit = 9;

long long ans = 0;

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

bool new\_cero = cero;

if( cero ){

if( i > 0 )

new\_cero = false;

}

bool aux = correct;

if( len2 == sz2 ){

ans += solve( len + 1 , flag | ( ( i < digit[ len ] )?1:0 ) , len2 , true,

s + " " + (char)(i + '0' ) , new\_cero , num \* 10 + i ) ;

}

else if( len2 < sz2 && D[ len2 ] - '0' == i && !new\_cero){

if( len2 + 1 == sz2 )

aux = true;

else{

ans += solve( len + 1 , flag | ( ( i < digit[ len ] )?1:0 ) , 0 , false,

s + " " + (char)( i + '0' ) , new\_cero , num \* 10 + i ) ;

}

ans += solve( len + 1 , flag | ( ( i < digit[ len ] )?1:0 ) , len2 + 1 , aux,

s + " " + D[ len2 ] , new\_cero , num \* 10 + i) ;

}

else ans += solve( len + 1 , flag | ( ( i < digit[ len ] )?1:0 ) , 0 , false,

s + " " + (char)(i + '0' ) , new\_cero , num \* 10 + i ) ;

}

return dp[ len ][ flag ][ len2 ][ correct ][ cero ] = ans;

}

int main(){

int A , B;

long long ans;

while( scanf("%d %d %s" , &A , &B , D ) , A > -1 ){

sz2 = strlen( D );

getDigits( B );

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

seen.clear();

ans = solve( 0 , 0 , 0 , 0 , " " , 1 , 0 );

if( A == 0 && D[ 0 ] == '0' ) ans++;

getDigits( A - 1 );

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

seen.clear();

ans -= solve( 0 , 0 , 0 , 0 , " " , 1 , 0 );

printf("%lld\n" , ans );

}

return 0;

}

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

# Graph Construction

\*\*\*ID: 10720

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

bool cmp( int n1 , int n2 ){

return n2 < n1;

}

int main(){

int n, d[ 10005 ] , sum, sum2;

bool p;

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

sum = 0;

p = true;

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

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

sum += d[ i ];

if( d[ i ] < 0 ) p = false;

}

if( ( sum & 1 ) || !p ){

puts("Not possible");

continue;

}

sort( d + 1 , d + n + 1 , cmp );

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

sum = 0;

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

sum += d[ i ];

sum2 = k \* ( k - 1 );

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

sum2 += min( k , d[ j ] );

}

if( sum > sum2 ){

p = false;

break;

}

}

if( !p ) break;

}

if( !p )puts("Not possible");

else puts("Possible");

}

return 0;

}

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

# Bar Codes

\*\*\*ID: 10721

\*\*\*Juez: UVA

\*\*\*Tipo: DP

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 55

long long dp[ MAX ][ MAX ][ 2 ];

int n , k , m;

long long solve( int sum , int prev , int cantidad ){

if( cantidad > k || sum > n ) return 0;

if( sum == n ){

if( cantidad == k ) return 1;

return 0;

}

if( dp[ sum ][ cantidad] [ prev ] != -1 ) return dp[ sum ][ cantidad] [ prev ];

long long ans = 0;

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

ans += solve( sum + i , !prev , cantidad + 1 );

}

return dp[ sum ][ cantidad][ prev ] = ans;

}

int main(){

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

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

printf("%lld\n" , solve( 0 , 0 , 0 ) );

}

return 0;

}

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

# Road Construction

\*\*\*ID: 10724

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Floyd Warshall

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 100

#define INF 1LL<<60

#define EPS 1e-8

double ady[MAX][MAX];

int a[ MAX ][ MAX ];

double x[ MAX ], y[ MAX ];

double getDist(int i , int j ){

return sqrt( (x[i] - x[j]) \* (x[i] - x[j]) + (y[i] - y[j]) \* (y[i] - y[j] ) );

}

int V, E;

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 ){

double t = ady[ i ][ k ] + ady[ k ][ j ];

if( t + EPS < ady[ i ][ j ] ){

ady[ i ][ j ] = t;

}

}

}

}

}

int main(){

int E, u , v;

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

for( int i = 0 ; i < V && scanf("%lf %lf" , &x[i] , &y[i]); ++i );

Init();

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

for( int i = 0 ; i < E && scanf("%d %d" , &u , &v ); ++i ){

u--; v--;

a[u][v] = a[v][u] = 1;

ady[u][v] = ady[v][u] = getDist(u , v);

}

floyd();

double ans = 0.0;

double maxi = -1.0, mini = 1LL<<60;

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

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

if( i == j || a[i][j] ) continue;

double d = getDist(i,j);

double cost = 0.0;

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

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

if( ii == jj ) continue;

if( (ady[ii][i] + d + ady[j][jj]) + EPS <= ady[ii][jj] ){

cost += (ady[ii][jj] - (ady[ii][i] + d + ady[j][jj]));

}

}

}

//cout<<i<<" "<<j<<" -- "<<cost<<" "<<d<<endl;

if( cost > 0 ){

if( cost > maxi + EPS ){

maxi = cost;

u = i; v = j;

mini = d;

}else if( cost == maxi ){

if( d + EPS < mini ){

mini = d;

u = i; v = j;

}

}

}

}

}

if( maxi < 0) puts("No road required");

else{

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

}

}

return 0;

}

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

# Test

\*\*\*ID: 10731

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, SCC

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1000

#define MAXN 26

vector<int> ady[ MAX ];

bool used[ MAXN ] , in\_stack[ MAXN ];

int lowlink[ MAXN ] , dfsNum[ MAXN ] , num\_scc, scc[ MAXN ] , idx;

stack<int> S;

void tarjan( int v ){

dfsNum[ v ] = lowlink[ v ] = idx++;

S.push( v ); in\_stack[ v ] = true;

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

int w = ady[ v ][ i ];

if( dfsNum[ w ] == -1 ){

tarjan( w );

lowlink[ v ] = min( lowlink[ v ] , lowlink[ w ] );

}

else if( in\_stack[ w ])lowlink[ v ] = min( lowlink[ v ] , lowlink[ w ] );

}

if( dfsNum[ v ] == lowlink[ v ] ){

int aux;

do{

aux = S.top(); S.pop();

scc[ aux ] = num\_scc;

in\_stack[ aux ] = false;

}while( aux != v );

num\_scc++;

}

}

void build\_scc(){

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

memset( in\_stack , false , sizeof( in\_stack ) );

idx = num\_scc = 0;

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

if( used[ i ] && dfsNum[ i ] == -1 )tarjan( i );

}

}

bool seen[ MAXN ];

int main(){

int n;

char c[ 6 ];

bool b = false;

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

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

b = true;

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

memset( used, 0 , sizeof( used ) );

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

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

scanf("\n%c %c %c %c %c %c", &c[ 1 ], &c[ 2 ], &c[ 3 ] , &c[ 4 ], &c[ 5 ], &c[ 0 ] );

used[ c[ 0 ]-'A' ] = true;

for( int j = 1 ; j < 6 ; ++j ){

used[ c[ j ] - 'A' ] = true;

if( c[ 0 ] != c[ j ] ){

ady[ c[ 0 ] - 'A' ].push\_back( c[ j ] - 'A' );

}

}

}

build\_scc();

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

if( used[ i ] && !seen[ i ] ){

bool first = true;

for( int j = i ; j < MAXN ; ++j ){

if( !seen[ j ] && used[ j ] && scc[ j ] == scc[ i ] ){

( first )? printf("%c", ( j +'A') ):printf(" %c", (j+'A') );

seen[ j ] = true;

first = false;

}

}

printf("\n");

}

}

}

return 0;

}

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

# Euler Circuit

\*\*\*ID: 10735

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, max flow, euler circuit

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 505

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;

int maxflow( int source , int sink ){

for( int f = 0; ;){

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 ){

int e = ady[ u ][ i ];

int v = E[ e ].v;

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

Q.push( v );

prev[ v ] = e;

}

}

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

int c = 99999999;

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

c = std::min( 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;

}

f += c;

}

}

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 );

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

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

E.push\_back( e2 );

}

vector<int> graph[MAX ];

int V;

int m;

void tour(int x)

{

while ( graph[ x ].size() > 0) {

int y = graph[ x ].back();

graph[ x ].pop\_back();

tour( y );

}

printf(m++ ? " %d" : "%d", x);

}

int main(){

int t, A , u , v ;

char c;

scanf("%d", &t );

int degree[ MAX ];

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

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

scanf("%d %d", &V , &A);

E.clear();

ady = vector<vector<int> >( V + A + 3, vector<int>() );

memset( degree, 0 , sizeof( degree ) );

int undirected[ A + 1 ];

memset( undirected, 0 , sizeof( undirected ) );

for( int i = 0 ; i <= V ;++i) graph[ i ].clear();

///auxiliar contiene aristas de grafo normal inicial

vector<Edge> aux;

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

scanf("%d %d %c",&u,&v,&c);

degree[ u ]++;

degree[ v ]++;

add( 0 , i , 1 );

add( i , A + v , 1 );

aux.push\_back( Edge( u , v , 0 , 0 ) );

if( c == 'U' ){

add( i , A + u , 1 );

undirected[ i ] = 1;

}else graph[ v ].push\_back( u );

}

bool possible = true;

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

if( degree[ i ] & 1 ){

possible = false;

break;

}

}

if( !possible ){

printf("No euler circuit exist\n");

continue;

}

///super source y super sink

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

add( A + i , A + V + 1 , degree[ i ]>>1 );

}

///MAX flow sobre grafo bipartito para hallar la cantidad de matchings

if( maxflow( 0 , V + A + 1 ) == A ){

///armo circuito para imprimir

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

if( undirected[ i ] ){

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

if( E[ ady[ i ][ j ] ].c == 0 ){

if( aux[ i - 1 ].u != E[ ady[ i ][ j ] ].v - A ){

graph[ E[ ady[ i ][ j ] ].v - A ].push\_back( aux[ i - 1].u);

}

else graph[ E[ ady[ i ][ j ] ].v - A ].push\_back( aux[ i - 1].v );

}

}

}

}

m = 0;

///para q funcione revertir las aristas

tour( 1 );

printf("\n");

}

else printf("No euler circuit exist\n");

}

return 0;

}

/\*\*

Grafo bipartito, Lado izquierdo aristas, Lado derecho vertices

No dirido, arista uno ambos, Dirigo solo al segundo

ejemplo 1 conecciones

1 -> 9 , 11

2 -> 9 , 12

3 -> 10, 12

4 -> 13

5 -> 12

6 -> 12, 13

7 -> 14

8 -> 13 , 14

Ahora creamos super source y super sink, super source con capacidad 1 x rekerir recorrer una arista

super sink con capacidad grado de vertice entre 2

\*\*/

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

# String to Palindrome

\*\*\*ID: 10739

\*\*\*Juez: UVA

\*\*\*Tipo: DP

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1005

char s[ MAX ];

int dp[ MAX ][ MAX ];

int solve( int left , int right ){

if( left + 1 == right ){

if( s[ left ] == s[ right ] ) return 0;

return 1;

}

if( left == right ){

return 0;

}

if( dp[ left ][ right ] != -1 ) return dp[ left ][ right ];

int mini = 1<<30;

if( s[ left ] == s[ right ] ) mini = min( mini , solve( left + 1 , right - 1 ) );

else{

mini = min( mini , min( 1 + solve( left + 1 , right ) , min( 1 + solve( left , right - 1 ) , 1 + solve( left + 1 , right - 1 ) ) ) );

}

return dp[ left ][ right ] = mini;

}

int main(){

int t , q;

scanf("%d" , &t );

for( q = 1 ; q <= t && scanf("%s" , &s ) ; ++q ){

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

printf("Case %d: %d\n" , q , solve( 0 , strlen( s ) - 1 ) );

}

return 0;

}

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

# New Rule in Euphomia

\*\*\*ID: 10742

\*\*\*Juez: UVA

\*\*\*Tipo: Sieve, Binary Search

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

#include <stdio.h>

#include <algorithm>

#include <cstring>

#include <iostream>

using namespace std;

#define MAX 1000005

int prime[ MAX ] , p[ MAX ], len;

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 = 2 ; i <= MAX ; ++i ){

if( prime[ i ] ) p[ len++ ] = i;

}

}

int main(){

sieve();

int n , idx , q = 1;

long long ans;

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

ans = 0;

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

if( p[ i ] > n || n <= p[ i ] \* 2 ) break;

idx = lower\_bound( p + i + 1 , p + len , n - p[ i ] ) - p;

if( !prime[ n - p[ i ] ] ) idx--;

ans += idx - i;

}

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

}

return 0;

}

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

# Dominant Strings

\*\*\*ID: 10745

\*\*\*Juez: UVA

\*\*\*Tipo: Trie + Sorting

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 27

struct Node{

int words;

Node\* hijo[ MAX ];

Node(){

words = 0;

for( int i = 0 ; i < MAX ; ++i ) hijo[ i ] = NULL;

}

};

class Trie{

public:

Trie(){

root = new Node();

}

void insert( string s );

bool contains( Node\* t , string s , int len );

Node\* getRoot();

private:

Node\* root;

};

Node\* Trie:: getRoot(){

return root;

}

void Trie::insert( string s ){

int i;

Node \*t = root;

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

if( t -> hijo[ s[ i ] - 'a' ] == NULL ) t -> hijo[ s[ i ] - 'a' ] = new Node();

t = t -> hijo[ s[ i ] - 'a' ];

}

t -> words++;

}

bool Trie::contains( Node \*t , string s , int len ){

if( s[ len ] == '\0' ) return 1;

if( t == NULL ) return 0;

int i ;

bool b = false;

for( i = 0 ; i <= s[ len ] - 'a' ; ++i ){

if( t -> hijo[ i ] != NULL ){

b |= contains( t -> hijo[ i ] , s , ( s[ len ] - 'a' == i ? len + 1 : len ) );

if( b ) return 1;

}

}

return b;

}

#define MAXN 15005

char words[ MAXN ][ MAX ], aux[ MAX ];

int len;

struct Data{

int first;

string second;

int len;

Data( int f , string s , int l ): first( f ) , second( s ) , len( l ){}

};

bool cmp( Data s1 , Data s2 ){

return s1.len > s2.len;

}

int main(){

Trie\* trie = new Trie();

int i , j;

len = 0;

vector< Data > v;

vector< string > ans;

while( scanf("%s" , &words[ len ] ) != EOF ){

strcpy( aux , words[ len++ ] );

sort( aux , aux + strlen( aux ) );

v.push\_back( Data( len - 1 , aux , strlen( aux ) ) );

}

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

string ss;

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

if( trie -> contains( trie -> getRoot() , v[ i ].second , 0 ) )continue;

trie -> insert( v[ i ].second );

ans.push\_back( words[ v[ i ].first ] );

}

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

for( i = 0 ; i < ans.size() ; ++i ) printf("%s\n" , ans[ i ].c\_str() );

delete trie;

return 0;

}

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

# Crime Wave -The Sequel

\*\*\*ID: 10746

\*\*\*Juez: UVA

\*\*\*Tipo: Min-Cost Max-Flow

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 50

#define INF 1<<30

#define EPS 1e-8

struct Edge{

int u , v, c;

double w;

Edge( int uu , int vv , int cc , double ww ): u( uu ) , v( vv ) , c( cc ) , w( ww ){}

};

vector< Edge > E;

vector< int > ady[ MAX ];

double dist[ MAX ];

int seen[ MAX ] , prev[ MAX ];

bool augmentPath( int source , int sink ){

int i , u , v , c;

double w;

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

dist[ i ] = INF;

seen[ i ] = 0;

prev[ i ] = -1;

}

queue< int > Q;

Q.push( source );

dist[ source ] = 0;

while( !Q.empty() ){

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

seen[ u ] = 0;

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

v = E[ ady[ u ][ i ] ].v;

c = E[ ady[ u ][ i ] ].c;

w = E[ ady[ u ][ i ] ].w;

if( dist[ v ] > dist[ u ] + w + EPS && c > 0 ){

dist[ v ] = dist[ u ] + w;

prev[ v ] = ady[ u ][ i ];

if( !seen[ v ] ){

seen[ v ] = 1;

Q.push( v );

}

}

}

}

return dist[ sink ] != INF;

}

double mcmf( int source , int sink ){

double mc = 0;

int mini , x;

while( augmentPath( source , sink ) ){

mini = INF;

for( x = sink ; prev[ x ] != -1 ; x = E[ prev[ x ] ].u ){

mini = min( mini , E[ prev[ x ] ].c );

}

for( x = sink ; prev[ x ] != -1 ; x = E[ prev[ x ] ].u ){

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

E[ prev[ x ] ^ 1 ].c += mini;

}

mc += mini \* dist[ sink ];

}

return mc;

}

void add( int u , int v , int c , double w ){

Edge e= Edge( u , v , c , w );

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

E.push\_back( e );

e = Edge( v , u , 0 , -w );

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

E.push\_back( e );

}

int main(){

int n , m , source = 0 , sink , i, j;

double w;

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

sink = n + m + 5;

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

for( j = 1 ; j <= m && scanf("%lf" , &w ) == 1 ; ++j ){

add( i , n + j , 1 , w );

}

add( source , i , 1 , 0.0 );

}

for( j = 1 ; j <= m ; ++j ) add( j + n , sink , 1 , 0.0 );

printf("%.2lf\n" , mcmf( source , sink )/(double)n + EPS );

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

E.clear();

}

return 0;

}

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

# Maximum Subsequence

\*\*\*ID: 10747

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

int a[ MAX ];

int n ,negative , positive;

int main(){

int k , ini, ans;

long long sum , prod;

int zeros;

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

positive = negative = 0;

zeros = 0;

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

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

if( a[ i ] < 0 ){

negative++;

}else if( a[ i ] > 0 ) {

positive++;

}

}

zeros = n - positive - negative;

sort( a , a + n );

vector<int> neg , pos;

for( int i = negative - 1 ; i >=0 ; --i ){

neg.push\_back( a[ i ] );

}

sum = 0;

for( int i = n - positive ; i < n ; ++i ){

pos.push\_back( a[ i ] );

sum += a[ i ];

}

int N = ( negative & 1 )? negative - 1: negative;

if( k > positive + negative || ( zeros > 0 && k > positive + N )

|| ( zeros > 0 && !positive && k & 1 ) ){

k -= positive;

k -= zeros;

for( int i = negative - 1 ; k > 0 && i >= 0 ; --i ){

sum += a[ i ];

k--;

}

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

}

else{

sum = 0;

int prodPos, prodNeg ,a , b , c ,d;

int ini = 0;

bool iniPos = true;

if( k & 1 ){

if( positive > 0 ){

sum += pos[ positive - 1 ];

positive--;

}

else if( negative > 0 ){

sum += neg[ 0 ];

ini = 1;

iniPos = false;

}

k--;

}

bool enterPos , enterNeg;

while( k > 0 && k % 2 == 0 ){

enterPos = enterNeg = false;

if( positive > 1 ){

enterPos = true;

prodPos = pos[ positive - 1 ] \* pos[ positive - 2 ];

}

if( negative > 1 + ini ){

enterNeg = true;

prodNeg = neg[ negative - 1 ] \* neg[ negative - 2 ];

}

if( enterPos ){

if( enterNeg ){

if( prodPos >= prodNeg ){

sum += pos[ positive - 1 ] + pos[ positive - 2 ];

positive -= 2;

k -= 2;

}

else{

sum += neg[ negative - 1 ] + neg[ negative - 2 ];

negative -= 2;

k -= 2;

}

}

else{

sum += pos[ positive - 1 ] + pos[ positive - 2 ];

positive -= 2;

k-=2;

}

}

else if( enterNeg ){

if( !iniPos ){

sum += neg[ ini ] + neg[ ini + 1 ];

ini += 2;

k -= 2;

}

else{

sum += neg[ negative - 1 ] + neg[ negative - 2 ];

negative -= 2;

k -= 2;

}

}

else break;

}

if( k > 0 ){

for( int i = positive - 1 ; k > 0 && i >= 0 ; --i ){

sum += pos[ i ];

k--;

}

if( k > 0 ){

for( int i = 0 ; k > 0 && i < negative ; ++i ){

sum += neg[ i ];

k--;

}

}

}

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

}

}

return 0;

}

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

# Back to Intermediate Math

\*\*\*ID: 10773

\*\*\*Juez: UVA

\*\*\*Tipo: Math

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t , q , i;

double ans , u , v , d;

scanf("%d" , &t );

for( q = 1 ; q <= t && scanf("%lf %lf %lf" , &d , &v , &u ) ; ++q ){

printf("Case %d: " , q );

if( u == 0 || u <= v || v == 0 ) puts("can't determine");

else printf("%.3lf\n" , fabs( ( d/u ) - ( d/ sqrt( u \* u - v \* v ) ) ) );

}

return 0;

}

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

# Collector's Problem

\*\*\*ID: 10779

\*\*\*Juez: UVA

\*\*\*Tipo: Max Flow

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 50

int cnt[ 26 ] , c[ MAX ][ MAX ] , prev[ MAX ] , seen[ MAX ];

bool augmentPath( int source , int sink ){

int u , v;

queue< int > Q;

Q.push( source );

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

prev[ source ] = -1;

seen[ source ] = 1;

while( !Q.empty() ){

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

if( u == sink ) return true;

for( v = 0 ; v <= sink ; ++v ){

if( c[ u ][ v ] > 0 && !seen[ v ] ){

prev[ v ] = u;

seen[ v ] = 1;

Q.push( v );

}

}

}

return false;

}

int maxflow( int source , int sink ){

int mf = 0 , mini , i;

while( augmentPath( source , sink ) ){

mini = 1<<30;

for( i = sink ; prev[ i ] != -1 ; i = prev[ i ] ){

mini = min( mini , c[ prev[ i ] ][ i ] );

}

for( i = sink ; prev[ i ] != -1 ; i = prev[ i ] ){

c[ prev[ i ] ][ i ] -= mini;

c[ i ][ prev[ i ] ] += mini;

}

mf += mini;

}

return mf;

}

int main(){

int t , q , n , m , k , i , x , source = 0, sink , j ;

scanf("%d" , &t);

for( q = 1 ; q <= t && scanf("%d %d" , &n , &m ); ++q ){

sink = n + m + 1;

memset( c , 0 , sizeof( c ) );

for( i = 0 ; i < n && scanf("%d" , &k ) ; ++i ){

memset( cnt , 0 , sizeof( cnt ) );

while( k-- > 0 && scanf("%d" , &x ) ){

cnt[ x ]++;

}

if( !i ){

for( j = 1 ; j <= m ; ++j ){

c[ source ][ j ] = cnt[ j ];

c[ j ][ sink ] = 1;

}

}

else{

for( j = 1 ; j <= m ; ++j ){

if( cnt[ j ] == 0 ){

c[ j ][ m + i ] = 1;

}

else if( cnt[ j ] > 1 ){

//siempre se debe quedar con 1 sticker

c[ m + i ][ j ] = cnt[ j ] - 1;

}

}

}

}

printf("Case #%d: %d\n" , q , maxflow( source , sink ) );

}

return 0;

}

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

# Again Prime! No Time

\*\*\*ID: 10780

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory, Prime Factorization

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

#include <stdio.h>

#include <iostream>

using namespace std;

#define MAX 10005

int factors[ MAX ];

int main(){

int t, m , n, idx, cont, min, num;

scanf( "%d" , &t );

for( int q = 1 ; q <= t; ++q ){

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

idx = 0;

int time[ MAX ] = { 0 };

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

if( m%i == 0 )factors[ idx++ ] = i;

while( m%i == 0){

time[ idx - 1 ]++;

m /= i;

}

}

if( m > 1 ) { factors[ idx++ ] = m; time[ idx - 1 ]++;}

min = 1<<30;

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

cont = 0;

int cont2 = 0;

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

num = i;

if( num % factors[ j ] == 0)cont2++;

while( num % factors[ j ] == 0 ){

cont++;

num /= factors[ j ];

}

}

min = std::min( min , cont/time[ j ] );

}

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

if( min == 1<<30 || min == 0 )printf("Impossible to divide\n");

else printf( "%d\n", min );

}

return 0;

}

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

# Diagonal

\*\*\*ID: 10784

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, math

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

///numero de diagonales es n(n-3)/2

///n\*(n-3)/2 >= N

///n^2 - 3\*n - 2\*N >= 0 usamos formula general

#include <stdio.h>

#include <cmath>

int main(){

long long N, n, a = 1 , b = -3 , c;

int t = 1;

while( scanf("%lld", &N ) ){

if( N == 0LL )break;

c = -2\*N;

n = (long long) ceil((-b + sqrt(b\*b - 4\*a\*c)) / 2\*a);

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

}

return 0;

}

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

# The Mad Numerologist

\*\*\*ID: 10785

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

char vowel[ 6 ] = "AUEOI";

char consonant[ 26 ] = "JSBKTCLDMVNWFXGPYHQZR";

int main(){

int t , n , idxVowel , idxConsonant , countConsonant , countVowel , lenVowel ,lenConsonant;

scanf("%d" , &t );

for( int q = 1 ; q <= t ; ++q ){

scanf("%d" , &n );

printf("Case %d: " , q );

countVowel = countConsonant = 0;

idxVowel = idxConsonant = 0;

string V = "" , C = "";

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

if( i & 1 ){

V += vowel[ idxVowel ];

countVowel++;

if( countVowel >= 21 ){

countVowel = 0;

idxVowel = ( idxVowel + 1 ) % 5;

}

}

else{

C += consonant[ idxConsonant ];

countConsonant++;

if( countConsonant >= 5 ){

countConsonant = 0;

idxConsonant = ( idxConsonant + 1 ) % 21;

}

}

}

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

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

lenVowel = lenConsonant = 0;

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

if( i & 1 ){

printf("%c" , V[ lenVowel++ ] );

}

else printf("%c" , C[ lenConsonant++ ] );

}

puts("");

}

return 0;

}

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

# Minimum Sum LCM

\*\*\*ID: 10791

\*\*\*Juez: UVA

\*\*\*Tipo: Prime Factorization

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

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);

}

long long solve( long long n ){

if( n == 1 ) return 2;

map< long long , int > f;

map< long long , int > :: iterator it;

long long i;

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

while( n % i == 0 ){

f[ i ]++;

n /= i;

}

}

if( n > 1 ) f[ n ]++;

long long ans = 0;

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

ans += pot( it -> first , it -> second );

}

if( f.size() == 1 ) ans++;

return ans;

}

int main(){

int q = 1;

long long n;

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

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

}

return 0;

}

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

# The Orc Attack

\*\*\*ID: 10793

\*\*\*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 ];

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 ] ) continue;

ady[ i ][ j ] = ady[ i ][ k ] + ady[ k ][ j ];

}

}

}

}

int main(){

int t , u , v, w , E , q , i , j , ans , maxi;

scanf("%d" , &t );

for( q = 1 ; q <= t ; ++q ){

scanf("%d %d" , &V , &E );

init();

while( E-- ){

scanf("%d %d %d" , &u , &v , &w );

ady[ u - 1 ][ v - 1 ] = ady[ v - 1 ][ u - 1 ] = min( ady[ u - 1 ][ v - 1 ] , w );

}

floyd();

ans = 1<<30;

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

for( j = 1 ; j < 5 ; ++j ){

if( ady[ j ][ i ] != ady[ j - 1 ][ i ] ) break;

}

if( j == 5 ){

maxi = 0;

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

maxi = ( ady[ i ][ j ] > maxi )? maxi = ady[ i ][ j ] :maxi;

}

ans = ( ans > maxi )? maxi:ans;

}

}

printf("Map %d: %d\n" , q , ( ans == 1<<30 )?-1: ans );

}

return 0;

}

/\*

3

7 11

1 7 2

2 7 2

3 7 2

5 7 2

6 7 1

1 6 1

2 6 1

3 6 1

4 6 1

5 6 1

7 6 1

6 1

1 2 3

10 17

7 1 12

7 2 12

7 9 35

7 8 2

7 5 12

1 9 20

2 3 7

2 8 10

3 8 10

9 5 20

9 5 31

9 10 40

9 6 16

9 8 10

8 4 10

4 6 4

6 10 35

\*/

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

# Not That Kind of Graph

\*\*\*ID: 10800

\*\*\*Juez: UVA

\*\*\*Tipo: String Output

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 200

#define R 0

#define F 1

#define C 2

char c[] = { '/' , '\\' , '\_' };

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

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

char s[ MAX ] , ady[ MAX ][ MAX ];

int main(){

int t , q , x , y , maxX , maxY , i , idx , j , k , minX;

scanf("%d" , &t );

for( q = 1 ; q <= t && scanf("%s" , &s ) ; ++q ){

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

for( j = 0 ; j < MAX ; ++j )

ady[ i ][ j ] = ' ';

x = MAX/2 - 2; y = 2; minX = x; maxY = 1; maxX = x;

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

if( s[ i ] == 'R' ) idx = R;

else if( s[ i ] == 'F' ) idx = F;

else idx = C;

if( idx == F ) ady[ x + 1 ][ y ] = c[ idx ];

else ady[ x ][ y ] = c[ idx ];

x += dx[ idx ]; y += dy[ idx ];

maxX = max( maxX , x );

minX = min( minX , x );

maxY = max( maxY , y );

}

maxX++;

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

string str;

for( i = minX ; i <= maxX ; ++i ){

str = "";

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

if( j == 0 ){

if( i < maxX ) str += "|";

else str += "+";

}

else{

if( i == maxX ) str +="-";

else str += ady[ i ][ j ];

}

}

if( i == maxX ) str += "-";

//Remove last espaces

for( j = str.size() - 1 ; j >= 0 ; --j )

if( str[ j ] != ' ') break;

if( str[ j ] == '|' ) continue;

for( k = 0 ; k <= j ; ++k ) printf("%c" , str[ k ] );

printf("\n");

}

printf("\n");

}

return 0;

}

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

# Thunder Mountain

\*\*\*ID: 10803

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Floyd Warshall

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

#include <stdio.h>

#include <cmath>

#define MAX 128

double ady[ MAX ][ MAX ];

int V , x[ MAX ] , y[ MAX];

int seen[ MAX ][ MAX ];

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

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

}

void Floyd(){

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

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

if( i == k || seen[ i ][ k ] == 0 )continue;

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

if( seen[ k ][ j ] == 0 )continue;

if( seen[ i ][ j ] == 1 && ady[ i ][ j ] < ady[ i ][ k ] + ady[ k ][ j ] )continue;

ady[ i ][ j ] = ady[ j ][ i ] = ady[ i ][ k ] + ady[ k ][ j ];

seen[ i ][ j ] = seen[ j ][ i ] = 1;

}

}

}

}

int main(){

int t;

scanf("%d" ,&t );

for( int q = 1 ; q <= t; ++q ){

scanf("%d" ,&V );

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

scanf("%d %d" , &x[ i ] , &y[ i ] );

}

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

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

ady[ i ][ j ] = ady[ j ][ i ] = dist( x[ i ] , y[ i ] , x[ j ] , y[ j ] );

if( ady[ i ][ j ] < ( 10.0 + 10e-9 ) ){

seen[ i ][ j ] = seen[ j ][ i ] = 1;

}

else seen[ i ][ j ] = seen[ j ][ i ] = 0;

}

ady[ i ][ i ] = 0;

seen[ i ][ i ] = 1;

}

Floyd();

int i , j;

double r;

for( i = 0 , r = 0 ; i < V ; ++i ){

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

if( seen[ i ][ j ] == 0 )break;

if( ady[ i ][ j ] > r ) r = ady[ i ][ j ];

}

if( j < i )break;

}

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

if( i < V )

printf("Send Kurdy\n\n");

else

printf("%.4f\n\n", r);

}

return 0;

}

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

# Dijkstra, Dijkstra.

\*\*\*ID: 10806

\*\*\*Juez: UVA

\*\*\*Tipo: Min-Cost Max-Flow

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

#define INF 1<<30

struct Edge{

int u , v, c;

long long w;

Edge( int uu , int vv , int cc , long long ww ): u( uu ) , v( vv ) , c( cc ) , w( ww ){}

};

vector< Edge > E;

vector< int > ady[ MAX ];

long long dist[ MAX ];

int seen[ MAX ] , prev[ MAX ];

bool augmentPath( int source , int sink ){

int i , u , v , c;

long long w;

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

dist[ i ] = INF;

seen[ i ] = 0;

prev[ i ] = -1;

}

queue< int > Q;

Q.push( source );

dist[ source ] = 0;

while( !Q.empty() ){

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

seen[ u ] = 0;

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

v = E[ ady[ u ][ i ] ].v;

c = E[ ady[ u ][ i ] ].c;

w = E[ ady[ u ][ i ] ].w;

if( dist[ v ] > dist[ u ] + w&& c > 0 ){

dist[ v ] = dist[ u ] + w;

prev[ v ] = ady[ u ][ i ];

if( !seen[ v ] ){

seen[ v ] = 1;

Q.push( v );

}

}

}

}

return dist[ sink ] != INF;

}

long long mc , mf;

void mcmf( int source , int sink ){

mc = mf = 0;

int mini , x;

while( augmentPath( source , sink ) ){

mini = INF;

for( x = sink ; prev[ x ] != -1 ; x = E[ prev[ x ] ].u ){

mini = min( mini , E[ prev[ x ] ].c );

}

for( x = sink ; prev[ x ] != -1 ; x = E[ prev[ x ] ].u ){

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

E[ prev[ x ] ^ 1 ].c += mini;

}

mf += mini;

mc += mini \* dist[ sink ];

}

}

void add( int u , int v , int c , long long w ){

Edge e= Edge( u , v , c , w );

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

E.push\_back( e );

e = Edge( v , u , 0 , -w );

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

E.push\_back( e );

}

int main(){

int V ,EE , u , v , i;

long long w;

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

scanf("%d" , &EE );

while( EE-- > 0 && scanf("%d %d %lld" , &u , &v , &w ) == 3 ){

add( u , v , 1 , w );

add( v , u , 1 , w );

}

add( 0 , 1 , 2 , 0 );

mcmf( 0 , V );

if( mf != 2 ) puts("Back to jail");

else printf("%lld\n" , mc );

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

E.clear();

}

return 0;

}

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

# Traditional BINGO

\*\*\*ID: 10813

\*\*\*Juez: UVA

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

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

#include <stdio.h>

int card[ 5 ][ 5 ];

int bingo[ 75 ];

bool Bingo(){

int r , c , di = 0, dd = 0;

for( int i = 0 ; i < 5 ; ++i ){

r = c = 0;

for( int j = 0 ; j < 5 ; ++j ){

r += card[ i ][ j ];

c += card[ j ][ i ];

}

if( !r || !c )return true;

di += card[ i ][ i ];

dd += card[ 4 - i ][ i ];

}

if( !di || !dd )return true;

return false;

}

int main(){

int t;

scanf("%d" , &t );

while( t-- ){

for( int i = 0 ; i < 5 ; ++i ){

for( int j = 0 ; j < 5 ; ++j ){

if( i == 2 && j == 2 ){

card[ i ][ j ] = 0; continue;

}

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

}

}

for( int i = 0 ; i < 75 ; ++i ) scanf("%d" , &bingo[ i ] );

int i = 0;

for( ; i < 75 ; ++i ){

for( int j = 0 ; j < 5 ; ++j ){

for( int k = 0 ; k < 5 ; ++k ){

if( card[ j ][ k ] == bingo[ i ] ) card[ j ][ k ] = 0;

}

}

if( Bingo() )break;

}

printf("BINGO after %d numbers announced\n" , i + 1 );

}

return 0;

}

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

# Andy's First Dictionary

\*\*\*ID: 10815

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, String

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

string s, str;

vector<string> v;

map<string , bool> mp;

while( getline( cin , s ) ){

str="";

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

if( isalpha( s [ i ]) ) str+=tolower( s[ i ] );

else str+=" ";

}

str+=" ";

stringstream ss(str);

while( ss>>s ){

if( !mp[ s ]) v.push\_back(s);

mp[ s ] = true;

}

}

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

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

cout<<v[i]<<endl;

}

return 0;

}

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

# Trouble of 13-Dots

\*\*\*ID: 10819

\*\*\*Juez: UVA

\*\*\*Tipo: DP , knapsack

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

#define MAXN 10500

int n , m , dp[ MAX ][ MAXN ] , a[ MAX ] , val[ MAX ];

int main(){

int ans;

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

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

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

}

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

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

for( int j = 1 ; j <= MAXN - 100 ; ++j ){

if( j - a[ i ] >= 0 ){

if( j == a[ i ] )

dp[ i ][ j ] = max( dp[ i - 1 ][ j - a[ i ] ] + val[ i ] , dp[ i - 1 ][ j ] );

else if( dp[ i - 1 ][ j - a[ i ] ] )

dp[ i ][ j ] = max( dp[ i - 1 ][ j - a[ i ] ] + val[ i ] , dp[ i - 1 ][ j ] );

else

dp[ i ][ j ] = dp[ i - 1 ][ j ];

}

else

dp[ i ][ j ] = dp[ i - 1 ][ j ];

}

}

ans = 0;

for( int i = 0 ; i <= MAXN - 100 ; ++i ){

if( i > 2000 && i - 200 <= m ) ans = max( ans , dp[ n ][ i ] );

else if( i <= m )ans = max( ans , dp[ n ][ i ] );

}

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

}

return 0;

}

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

# Maximum sum on a torus

\*\*\*ID: 10827

\*\*\*Juez: UVA

\*\*\*Tipo: DP

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 80

int h , w;

int a[ MAX ][ MAX ], sum , s , maxi, n;

int pr[ MAX ];

int Kadane2D(){

maxi = 0;

int resp = 0;

int k, l , j , tot, minSum, mini;

for( int z = 0 ; z < h; ++z ){

for( int i = 0 ; i < w ; ++i ) pr[ i ] = 0;

//si empiezo en la segunda fila, evaluo filas superiores

for( int x = z ,k = 0 ; k < h ; x = ( x + 1 ) % h , ++k ){

s = 1<<31;

sum = 0;

tot = 0;

minSum = 0;

mini = 1<<30;

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;

}

tot += pr[ i ];

minSum += pr[ i ];

if( minSum > 0 ) minSum = 0;

if( mini > minSum ) mini = minSum;

}

if( s > maxi ){

maxi = s;

}

if( tot - mini > maxi ) maxi = tot - mini;

}

}

return maxi;

}

int main(){

int t;

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 ] );

}

}

h = w = n;

printf("%d\n" , Kadane2D() );

}

return 0;

}

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

# Traffic Flow

\*\*\*ID: 10842

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, MST

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

#include <bits/stdc++.h>

#include <cstdlib>>

using namespace std;

#define MAX\_SIZE 16001

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;

}

}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));

}

int ady[ MAX\_SIZE ][ MAX\_SIZE ];

///END UNION FIND

int f(const void \*p, const void \*q)

{

return ((Arista \*)q )->peso - ((Arista \*)p)->peso;

}

int main(){

int t , V, E , u , vv , c;

scanf("%d" ,&t );

for( int q = 1 ; q <= t ; ++q ){

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 );

}

qsort( v, E , sizeof(v[0]), &f);

for( int i = 0 ; i < V ; ++i ) Make\_Set( i );

int min = v[ 0 ].peso;

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 ) ){

if( min > w )min = w;

Merge( u , vv );

}

}

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

}

return 0;

}

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

# Move the bishop

\*\*\*ID: 10849

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <stdio.h>

int abs( int x ){

if( x < 0 ) x = -x;

return x;

}

bool par( int x ){

if( x & 1 ) return 0;

return 1;

}

bool colorWhite( int x1 , int y1 , int x2 , int y2 ){

if( ( par( x1 ) && !par( y1 ) ) || ( !par( x1 ) && par( y1 ) ) ){

if(( par( x2 ) && !par( y2 ) ) || ( !par( x2 ) && par( y2) ))return true;

}

return false;

}

bool colorBlack( int x1 , int y1 , int x2 , int y2 ){

if( ( par( x1 ) && par( y1 ) ) || ( !par( x1 ) && !par( y1 ) ) ){

if( ( par( x2 ) && par( y2 ) ) || ( !par( x2 ) && !par( y2) ))return true;

}

return false;

}

int main(){

int t , n, xini , yini , xfin , yfin, q ,dif1 , dif2;

scanf("%d" , &t );

while( t-- ){

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

while( q-- ){

scanf("%d %d %d %d" , &xini , &yini , &xfin ,&yfin );

if( xini <= 0 || xini > n || yini <= 0 || yini > n || xfin <= 0 || xfin >n || yfin <= 0 || yfin > n ){

puts("no move");

continue;

}

if( xini == xfin && yini == yfin )puts("0");

else if( abs( xini - xfin ) == abs( yini - yfin ) )puts("1");

else if( ( xini + yini ) == ( xfin + yfin ) )puts("1");

else if( colorWhite( xini , yini , xfin , yfin ) || colorBlack( xini , yini , xfin , yfin ) )puts("2");

else puts("no move");

}

}

return 0;

}

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

# 2D Hieroglyphs decoder

\*\*\*ID: 10851

\*\*\*Juez: UVA

\*\*\*Tipo: Strings

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 100

#define ASCII 122

char ady[ MAX ][ MAX ];

int main(){

int t , i , M , k , j , c;

scanf("%d" , &t );

while( t-- > 0 ){

for( i = 0 ; i < 10 && scanf("%s" , ady[ i ] ) ; ++i );

M = strlen( ady[ 0 ]) ;

for( j = 1 ; j < M - 1 ; ++j ){

for( i = 8 , c = 0 ; i > 0 ; --i ){

if( ady[ i ][ j ] == '/') c <<=1;

else c = ( c<<1 ) | 1;

}

putchar( c );

}

printf("\n");

}

return 0;

}

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

# Less Prime

\*\*\*ID: 10852

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <stdio.h>

#include <math.h>

#define MAX 10005

int primes[MAX];

int idx;

void Sieve(){

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 main(){

int t , n, p, i;

scanf("%d", &t );

Sieve();

while( t-- ){

scanf("%d", &n);

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

if( 2\* primes[ i ] > n) break;

}

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

}

return 0;

}

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

# Rotated Squares

\*\*\*ID: 10855

\*\*\*Juez: UVA

\*\*\*Tipo: Simulation

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

int n , h;

char ady[ MAX ][ MAX ];

char table[ MAX ][ MAX ];

char aux[ MAX ][ MAX ];

void rotate(){

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

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

aux[ j ][ h - i - 1 ] = table[ i ][ j ];

}

}

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

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

table[ i ][ j ] = aux[ i ][ j ];

}

}

}

int check(){

int cnt = 0;

bool finish;

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

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

finish = true;

int ii = i, jj = j;

for( int k = 0 ; k < h ; ++k , ii++ ){

jj = j;

for( int q = 0 ; q < h ; ++q , jj++ ){

if( table[ k ][ q ] != ady[ ii ][ jj ] ){

finish = false;

k = h;

break;

}

}

}

if( finish ) cnt++;

}

}

return cnt;

}

int main(){

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

for( int i = 0 ; i < n ; ++i )scanf("%s" , &ady[ i ] );

for( int i = 0 ; i < h ; ++i )scanf("%s" , &table[ i ] );

printf("%d" , check());

rotate();

printf(" %d" , check());

rotate();

printf(" %d" , check());

rotate();

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

}

return 0;

}

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

# Unique Factorization

\*\*\*ID: 10858

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory, Recursion

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1500

int fact[ MAX ];

int ans[ 7000 ][ 500 ], lresp, resp[ MAX ];

int cont, N;

void solve( int n , int len , int act){

int l = ( int ) sqrt( n );

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

if( n % i == 0 ){

resp[ len ] = i;

solve( n / i , len + 1 , i );

}

}

resp[ len ] = n;

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

ans[ cont ][ i ] = resp[ i ];

}

cont++;

}

int main(){

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

int l = ( int ) sqrt( N );

int len = 0;

for( int i = 2 ; i <= l ; ++i ){

if( N % i == 0 ){

fact[ len++ ] = i;

}

}

if( len == 0 ){

printf("0\n");

continue;

}

cont = 0;

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

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

resp[ 0 ] = fact[ i ];

solve( N/fact[ i ] , 1 , fact[ i ] );

}

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

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

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

for( int j = 1 ; j < 500 ; ++j ){

if( ans[ i ][ j ] == -1 )break;

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

}

printf("\n");

}

}

return 0;

}

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

# Brownie Points I

\*\*\*ID: 10865

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, Geometry

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

#include <stdio.h>

#define EPS 1e-9

struct Point{

Point(int x1,int y1){

x=x1;

y=y1;

}

Point(){

}

int x;

int y;

}v[ 200005 ];

int main(){

int n, Stan, Ollie;

int x,y , len;

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

len = 0;

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

Point p;

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

v[ len++ ] = p;

}

x = v[ n/2 ].x;

y = v[ n/2 ].y;

Stan = 0; Ollie = 0;

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

if( ( v[ i ].x > x && v[ i ].y > y ) || ( v[ i ].x < x && v[ i ].y < y ) ){

Stan++;

}

else if( ( v[ i ].x > x && v[ i ].y < y ) || ( v[ i ].x < x && v[ i ].y >y ) ){

Ollie++;

}

}

printf("%d %d\n", Stan, Ollie);

}

return 0;

}

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

# Recurrences

\*\*\*ID: 10870

\*\*\*Juez: UVA

\*\*\*Tipo: Matrix Exponentiation

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 20

int n , MOD;

typedef long long type;

struct Matrix{

int r , c;

type mat[ MAX ][ MAX ];

};

Matrix MatrixMult( Matrix A , Matrix B ){

Matrix ans; int i , j , k;

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

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

for( ans.mat[ i ][ j ] = k = 0 ; k < n ; ++k )

ans.mat[ i ][ j ] += A.mat[ i ][ k ] \* B.mat[ k ][ j ] , ans.mat[ i ][ j ] %= MOD;

return ans;

}

Matrix MatrixPow( Matrix A , long long p ){

int i , j;

Matrix ans;

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

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

ans.mat[ i ][ j ] = ( i == j );

while( p ){

if( p & 1 ) ans = MatrixMult( ans , A );

A = MatrixMult( A , A );

p >>= 1;

}

return ans;

}

Matrix MatrixMultDif( Matrix A , Matrix B ){

Matrix ans; int i , j , k;

ans.r = A.r;

ans.c = B.c;

for( i = 0 ; i < A.r ; ++i )

for( j = 0 ; j < B.c ; ++j )

for( ans.mat[ i ][ j ] = k = 0 ; k < A.c ; ++k )

ans.mat[ i ][ j ] += A.mat[ i ][ k ] \* B.mat[ k ][ j ] , ans.mat[ i ][ j ] %= MOD;

return ans;

}

void print( Matrix m ){

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

for( int j = 0 ; j < m.c ; ++j ){

cout<<m.mat[ i ][ j ]<<" ";

}

cout<<endl;

}

}

int main(){

int idx , i;

Matrix ans , fib;

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

memset( ans.mat , 0 , sizeof( ans.mat ) );

for( i = 0 ; i < n && scanf("%lld" , &ans.mat[ 0 ][ i ] ) ; ++i );

for( i = 0 ; i < n && scanf("%lld" , &fib.mat[ n - i - 1 ][ 0 ] ) ; ++i );

for( i = 0 ; i < n ; ++i ) ans.mat[ i + 1 ][ i ] = 1;

fib.r = n; fib.c = 1;

ans = MatrixPow( ans , idx - 1 );

ans.r = ans.c = n;

ans = MatrixMultDif( ans , fib );

ans.r = n; ans.c = 1;

printf("%lld\n" , ans.mat[ n - 1 ][ 0 ] );

}

return 0;

}

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

# Decode the tape

\*\*\*ID: 10878

\*\*\*Juez: UVA

\*\*\*Tipo: Strings

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 30

char s[ MAX ];

int main(){

int len , i , c;

while( gets( s ) ){

if( s[ 0 ] == '\_' ) continue;

len = strlen( s );

c = 0;

for( i = len - 2 ; i > 0 ; --i ){

if( s[ i ] == 'o' ) c |= 1<<( len - i - 2 );

else if( s[ i ] == '.' ) len--;

}

putchar( c );

}

return 0;

}

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

# Code Refactoring

\*\*\*ID: 10879

\*\*\*Juez: UVA

\*\*\*Tipo: Factors

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 3200

int fact[ MAX ];

int main(){

int n, len , l , t , q , i;

scanf("%d" , &t );

for( q = 1 ; q <= t && scanf("%d" , &n ) ; ++q ){

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

len = (int)( sqrt( n ) );

l = 0;

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

if( n % i == 0 ){

fact[ l++ ] = i;

if( i != n/i )

fact[ l++ ] = n/i;

}

}

sort( fact , fact + l );

len = 0;

for( i = 0 ; i < l && len < 2 ; ++i ){

if( fact[ i ] == 1 || n == fact[ i ] ) continue;

printf(" = %d \* %d" , fact[ i ] , n/fact[ i ] );

len++;

}

printf("\n");

}

return 0;

}

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

# Colin and Ryan

\*\*\*ID: 10880

\*\*\*Juez: UVA

\*\*\*Tipo: Factores de un numero

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int t, C, R , n, len, fact[ 45000 ] , l ;

scanf("%d" , &t );

for( int q = 1 ; q<= t ; ++q ){

scanf("%d %d" , &C , &R );

n = C - R;

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

if( n == 0 ){

printf(" 0\n");

continue;

}

len = (int)( sqrt( n ) );

l = 0;

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

if( n % i == 0 ){

fact[ l++ ] = i;

if( i != n/i )

fact[ l++ ] = n/i;

}

}

sort( fact , fact + l );

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

if( fact[ i ] > R )printf(" %d" , fact[ i ] );

}

printf("\n");

}

return 0;

}

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

# Matrix Transpose

\*\*\*ID: 10895

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int h , w, k , col[ 10005 ], num;

while( scanf("%d %d" , &h , &w ) == 2 ){

vector<pair<int , int > > v[ 10005 ] , T[ 10005 ];

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

scanf("%d" , &k );

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

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

}

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

scanf("%d" , &num );

v[ i ].push\_back( make\_pair( col[ j ] , num ) );

}

}

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

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

T[ v[ i ][ j ].first - 1 ].push\_back( make\_pair( i + 1 , v[ i ][ j ].second ) );

}

}

printf("%d %d\n" , w , h );

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

printf("%d" , T[ i ].size() );

for( int j = 0 ; j < T[ i ].size() ; ++j ){

printf(" %d" , T[ i ][ j ].first );

}

putchar('\n');

for( int j = 0 ; j < T[ i ].size() ; ++j ){

if( j != 0 )printf(" ");

printf("%d" , T[ i ][ j ].second );

}

putchar('\n');

}

}

return 0;

}

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

# Ferry Loading III

\*\*\*ID: 10901

\*\*\*Juez: UVA

\*\*\*Tipo: Simulation, queue

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 10005

#define LEFT 0

#define RIGHT 1

struct Data{

int idx , val, dir;

Data( int i , int v , int d ): idx( i ) , val( v ), dir( d ){}

Data(){}

};

int main(){

int C, x,time, a[ MAX ] , pos, n , t, m;

char s[ 8 ];

scanf("%d" , &C );

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

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

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

deque<Data> Q;

scanf("%d %d %d" , &n , &t , &m );

pos = LEFT;

time = 0;

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

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

if( !strcmp( s , "right" ) ){

Q.push\_back( Data( i , x , RIGHT ) );

}

else Q.push\_back( Data( i , x , LEFT ) );

}

int acu = 0;

Data last;

while( !Q.empty() ){

deque<Data> copy = Q;

Data act = Q.front();

if( act.val > time ) time += ( act.val - time );

Data next;

vector<int> idx;

vector<int> borrar;

int value = 1;

for( int i = 0 , j = 0 ; j < n && !copy.empty() ; ++i ){

next = copy.front(); copy.pop\_front();

if( next.dir == pos && next.val <= time ){

idx.push\_back( next.idx );

borrar.push\_back( i );

j++;

}

if( next.dir == pos && next.val > time )break;

}

int del = 0;

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

Q.erase( Q.begin() + borrar[ i ] - del );

del++;

}

//regreso

time += t;

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

a[ idx[ i ] ] = time;

}

pos = !pos;

}

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

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

}

}

return 0;

}

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

# Marks Distribution

\*\*\*ID: 10910

\*\*\*Juez: UVA

\*\*\*Tipo: DP

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

#include <stdio.h>

#include <cstring>

#define MAX 75

int n , p, dp[ MAX ][ MAX ] , t;

int solve( int sum , int k ){

if( sum < 0 ) return 0;

if( k == n ){

if( sum == 0 ) return 1;

return 0;

}

if( dp[ sum ][ k ] != -1 ) return dp[ sum ][ k ];

int ans = 0;

for( int i = p ; i <= t ; ++i ){

ans += solve( sum - i , k + 1 );

}

return dp[ sum ][ k ] = ans;

}

int main(){

int k;

scanf("%d" , &k );

while( k-- ){

scanf("%d %d %d" , &n , &t , &p );

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

printf("%d\n" , solve( t , 0 ) );

}

return 0;

}

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

# Forming Quiz Teams

\*\*\*ID: 10911

\*\*\*Juez: UVA

\*\*\*Tipo: DP + bitmask

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

#include <stdio.h>

#include <cmath>

#include <iostream>

using namespace std;

#define MAX 9

#define MAXN 20

int n, y[ MAXN ], x[ MAXN ];

char s[ 25 ];

double d[ MAXN ][ MAXN ];

double dp[ MAXN ][ ( 1<<16 ) + 5 ];

double dist( int x1 , int y1 , int x2 , int y2 ){

return sqrt( ( x1 - x2 ) \* ( x1 - x2 ) + ( y1 - y2 ) \* ( y1 - y2 ) );

}

int bitcount( int n ){

int cnt = 0;

while( n ){

n = n & ( n - 1 );

cnt++;

}

return cnt;

}

double solve( int last , int mask ){

int bc = bitcount( mask );

if( bc == n ) return 0;

if( dp[ last ][ mask ] != -1 ) return dp[ last ][ mask ];

double mini = 1<<30;

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

if( !( mask & 1<<i ) ){

if( bc & 1 ) mini = min( mini , d[ last ][ i ] + solve( i , mask | 1<<i ) );

else mini = min( mini , solve( i , mask | 1<<i ) );

}

}

return dp[ last ][ mask ] = mini;

}

int main(){

int q = 1;

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

n \*= 2;

for( int i = 0 ; i < n ; ++i ) scanf("%s %d %d" , &s , &x[ i ] , &y[ i ] );

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

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

d[ j ][ i ] = d[ i ][ j ] = dist( x[ i ] , y[ i ] , x[ j ] , y[ j ] );

}

}

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

for( int j = 0 ; j < 1<<n ; ++j )

dp[ i ][ j ] = -1;

printf("Case %d: %.2lf\n", q++ , solve( 0 , 1 ) );

}

return 0;

}

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

# Simple Minded Hashing

\*\*\*ID: 10912

\*\*\*Juez: UVA

\*\*\*Tipo: DP

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

#include <stdio.h>

#include <string.h>

#define MAX 10004

int n, s, dp[ MAX ][ 29 ][ 29 ];

int solve( int sum , int k , int last ){

if( sum > 351 || k > 26 || sum > s ) return 0;

if( n == k ){

if( sum == s ) return 1;

return 0;

}

if( dp[ sum ][ k ][ last ] != -1 ) return dp[ sum ][ k ][ last ];

int ans = 0, i;

for( i = last + 1 ; i <= 26 ; ++i ){

ans += solve( sum + i , k + 1 , i );

}

return dp[ sum ][ k ][ last ] = ans;

}

int main(){

int q = 1 , i , j , k;

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

if( n > 26 || s > 351 ){

printf("Case %d: 0\n" , q++ );

continue;

}

for( i = 0 ; i < s + 5 ; ++i )

for( j = 0 ; j < 27 ; ++j )

for( k = 0 ; k < 27 ; ++k )

dp[ i ][ j ][ k ] = -1;

printf("Case %d: %d\n" , q++ , solve( 0 , 0 , 0 ) );

}

return 0;

}

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

# Walking on a Grid

\*\*\*ID: 10913

\*\*\*Juez: UVA

\*\*\*Tipo: DP

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

#include <stdio.h>

#include <string.h>

#define MAX 76

#define INF 1LL<<50

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

long long max( long long a , long long b ){ return ( a < b )? b : a;}

long long dp[ MAX ][ MAX ][ 5 ][ 6 ];

int dx[ 3 ] = { 0 , 0 , 1};

int dy[ 3 ] = { -1 , 1 , 0};

long long solve( int x , int y , int K , int ant , int px , int py ){

if( K > k ) return -INF;

if( x == n - 1 && y == n - 1 ){

if( K <= k )return 0;

return -INF;

}

if( dp[ x ][ y ][ ant ][ K ] != -1LL<<60 ) return dp[ x ][ y ][ ant ][ K ];

long long ans = -INF , aux;

int i , nx , ny;

for( i = 0 ; i < 3 ; ++i ){

nx = dx[ i ] + x;

ny = dy[ i ] + y;

if( nx >= 0 && ny >= 0 && nx < n && ny < n && ( nx != px || ny != py ) ){

aux = solve( nx , ny , K +( ( ady[ nx ][ ny ] < 0 )?1:0 ) , i , x , y );

if( aux != -INF )

ans = max( ans , ady[ nx ][ ny ] + aux );

}

}

return dp[ x ][ y ][ ant ][ K ] = ans;

}

int main(){

int i , j , q = 1 , kk , jj;

long long ans;

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

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

for( j = 0 ; j < n && scanf("%d" , &ady[ i ][ j ] ) == 1 ; ++j );

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

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

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

dp[ i ][ j ][ 0 ][ kk ] = -1LL<<60;

dp[ i ][ j ][ 1 ][ kk ] = -1LL<<60;

dp[ i ][ j ][ 2 ][ kk ] = -1LL<<60;

}

}

}

ans = solve( 0 , 0 , ( ady[ 0 ][ 0 ] < 0 )?1:0 , 0 , 0 , 0 );

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

if( ans == -INF ) puts("impossible");

else printf("%lld\n" , ans + ady[ 0 ][ 0 ] );

}

return 0;

}

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

# Factstone Benchmark

\*\*\*ID: 10916

\*\*\*Juez: UVA

\*\*\*Tipo: Logaritmos

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1000000

double f[ MAX ];

//2^( 2 ^( y + 1 ) ) > n!

int main(){

int year , y , n , i;

f[ 0 ] = log( 1 );

for( i = 1 ; i < MAX ; ++i )

f[ i ] = log( i ) + f[ i - 1 ];

int left , right , mid;

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

y = ( year - 1960 )/10;

left = 0; right = MAX;

while( left < right ){

mid = ( left + right )>>1;

if( f[ mid ] > (double)( 1<<( y + 2 ) ) \* log( 2 ) ) right = mid;

else left = mid + 1;

}

printf("%d\n" , right - 1 );

}

return 0;

}

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

# 2 the 9s

\*\*\*ID: 10922

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <stdio.h>

#include <cstring>

int main(){

char s[ 1005 ];

int degree;

long long sum, number;

while( scanf("%s" , &s ) , strcmp( s , "0" )){

sum = 0;

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

sum += s[ i ] - '0';

}

degree = 0;

number = sum;

while( number % 9 == 0 ){

degree++;

if( number == 9 )break;

sum = 0;

while( number > 0 ){

sum += number % 10;

number /= 10;

}

number = sum;

}

if( degree == 0 )printf("%s is not a multiple of 9.\n", s );

else printf("%s is a multiple of 9 and has 9-degree %d.\n", s, degree );

}

return 0;

}

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

# My Dear Neighbours

\*\*\*ID: 10928

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, Graph(grado de nodo)

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define INF 1<<20

int Int(string s){

int n;

stringstream ss( s );

ss>>n;

return n;

}

int main(){

int t , p, min, index;

cin>>t;

string line;

while( t-- ){

cin>>p;

int ady[ p + 1 ];

for( int i = 1 ; i <= p ; ++i )ady[i] = 0;

getline( cin , line );

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

getline( cin , line );

line += " ";

stringstream ss( line );

while( ss>>line ){

ady[ i ]++;

}

}

min = INF;

index = 1;

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

if( min > ady[ i ] ){

min = ady[i];

index = i;

}

}

cout<<index;

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

if( i != index && ady [ i ] == min )cout<<" "<<i;

}

cout<<endl;

}

return 0;

}

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

# A-Sequence

\*\*\*ID: 10930

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 35

int a[ MAX ] , seen[ 1005 ];

bool b;

int dp[ MAX ][ 1005 ];

int possible( int len , int s ){

if( s < 0 ) return 1;

if( len == -1 ){

if( s == 0 ) return 0;

return 1;

}

if( dp[ len ][ s ] != -1 ) return dp[ len ][ s ];

bool b = possible( len - 1 , s - a[ len ] ) & possible( len - 1 , s );

return dp[ len ][ s] =b;

}

int main(){

int q = 1 , i , n ;

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

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

for( i = 0 ; i < n && scanf("%d" , &a[ i ] ) ; ++i ) printf(" %d" , a[ i ] );

printf("\n");

b = 1;

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

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

if( a[ i - 1 ] > a[ i ] || !possible( i - 1 , a[ i ] ) ){

b = 0; break;

}

}

if( b ) puts("This is an A-sequence.");

else puts("This is not an A-sequence.");

}

return 0;

}

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

# Throwing cards away I

\*\*\*ID: 10935

\*\*\*Juez: UVA

\*\*\*Tipo: Queue

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int n, ini;

bool first;

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

printf("Discarded cards:");

queue<int> Q;

for( int i = 1 ; i <= n ; ++i ) Q.push( i );

ini = 1;

first = true;

while( !Q.empty() && n > 1 ){

if( first ) printf(" 1");

else printf(", %d" , Q.front() );

Q.pop();

n--;

Q.push( Q.front() );

ini = Q.front();

Q.pop();

first = false;

}

printf("\nRemaining card: %d\n" , ini );

}

return 0;

}

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

# Flea Circus

\*\*\*ID: 10938

\*\*\*Juez: UVA

\*\*\*Tipo: LCA

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

#include <stdio.h>

#include <cstring>

#include <vector>

#include <iostream>

using namespace std;

#define MAX 10010

#define pb push\_back

#define mp make\_pair

#define x first

#define y second

#define pii pair<int , int >

#define MIN( a , b ) ( a < b ) ? a : b

int n, parent[ MAX ] , kparent[ MAX ][ 15 ] , depth[ MAX ], seen[ MAX ], dist[ MAX ];

vector< pii > ady[ MAX ];

void dfs( int x , int lvl ){

seen[ x ] = 1;

depth[ x ] = lvl;

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

if( !seen[ ady[ x ][ i ].first ] ){

parent[ ady[ x ][ i ].first ] = x;

dist[ ady[ x ][ i ].first ] = dist[ x ] + ady[ x ][ i ].second;

dfs( ady[ x ][ i ].first , lvl + 1 );

}

}

}

void lca\_init(){

int i , j;

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

for( i = 0 ; i < n ; ++i ) kparent[ i ][ 0 ] = parent[ i ];

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

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

kparent[ i ][ j ] = kparent[ kparent[ i ][ j - 1 ] ][ j - 1 ];

}

int LCA( int p , int q ){

int i, j;

if( depth[ p ] < depth[ q ] ) swap( p , q );

for( j = 1 ; 1<<j <= depth[ p ] ; ++j ); --j;

for( i = j ; i >= 0 ; --i )

if( depth[ p ] - ( 1<<i ) >= depth[ q ] ) p = kparent[ p ][ i ];

if( p == q ) return q;

for( i = j ; i >= 0 ; --i )

if( kparent[ p ][ i ] != -1 && kparent[ p ][ i ] != kparent[ q ][ i ] )

p = kparent[ p ][ i ] , q = kparent[ q ][ i ];

return parent[ p ];

}

//devuelve el kth elemento del lca desde u a v

//k es la profundidad al que tengo q igualar

int find\_kth( int p , int k ){

int j , i ;

for( j = 1 ; 1<<j <= depth[ p ] ; ++j ); --j;

for( i = j ; i >= 0 ; --i )

if( depth[ p ] - ( 1<<i ) >= k ) p = kparent[ p ][ i ];

return p;

}

int main(){

int d, Q , u , v, w, ini, lca , k , uu, vv;

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

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

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

u--;

v--;

ady[ u ].pb( pii( v , 1 ) );

ady[ v ].pb( pii( u , 1 ) );

ini = u;

}

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

memset( dist , 0 , sizeof( dist ) );

dfs( ini , 0 );

lca\_init();

scanf("%d" , &Q );

while( Q-- ){

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

u--; v--;

lca = LCA( u , v );

d = dist[ u ] + dist[ v ] - 2 \* dist[ lca ] + 1;

int l1 = depth[ u ] - depth[ lca ] + 1;

if( d & 1 ){

k = d/2 + 1;

if( k <= l1 ){

printf("The fleas meet at %d.\n" , find\_kth( u , depth[ lca ] + ( l1 - k )) + 1 );

}

else{

printf("The fleas meet at %d.\n" , find\_kth( v , depth[ lca ] + (k - l1 ) ) + 1 );

}

}

else{

k = d/2;

if( k <= l1 ){

uu = find\_kth( u , depth[ lca ] + ( l1 - k )) + 1 ;

}

else{

uu = find\_kth( v , depth[ lca ] + (k - l1 ) ) + 1 ;

}

k++;

if( k <= l1 ){

vv = find\_kth( u , depth[ lca ] + ( l1 - k )) + 1 ;

}

else{

vv = find\_kth( v , depth[ lca ] + (k - l1 ) ) + 1 ;

}

printf("The fleas jump forever between %d and %d.\n" , min( uu , vv ) , max( uu , vv ) );

}

}

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

}

return 0;

}

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

# How do you add?

\*\*\*ID: 10943

\*\*\*Juez: UVA

\*\*\*Tipo: DP

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

#include <bits/stdc++.h>

#include <cstdlib>>

using namespace std;

#define MOD 1000000

#define MAX 105

int dp[ MAX ][ MAX ];

int n , K;

int solve( int sum , int k ){

if( sum < 0 ) return 0;

if( k == K ){

if( sum == 0 ) return 1;

return 0;

}

if( dp[ sum ][ k ] != -1 ) return dp[ sum ][ k ] % MOD;

int cnt = 0;

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

cnt = ( cnt % MOD + solve( sum - i , k + 1 ) % MOD ) % MOD;

}

return dp[ sum ][ k ] = cnt % MOD;

}

int main(){

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

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

printf("%d\n" , solve( n % MOD , 0 ) );

}

return 0;

}

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

# Mother Bear

\*\*\*ID: 10945

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <stdio.h>

#include <cstring>

#include <ctype.h>

int main(){

char line[ 1005 ];

char s[ 1005 ];

int l , len;

while( gets( line ) && strcmp( line , "DONE") ){

l = strlen( line );

len = 0;

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

if( ( 'a' <=line[ i ] && line[ i ] <= 'z' ) || ( 'A' <=line[ i ] && line[ i ] <= 'Z' ))

s[ len++ ] = tolower( line[ i ] );

}

s[ len ] = '\0';

int i;

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

if( s[ i ] != s[ len - i - 1 ] )break;

}

if( i == len )puts("You won't be eaten!");

else puts("Uh oh..");

}

return 0;

}

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

# Bad Code

\*\*\*ID: 10950

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAXN 10005

#define MAX 30

int n , sz;

struct Data{

char c;

int val;

Data( char cc , int vv ) : c( cc ) , val( vv ){}

Data(){}

bool operator < ( Data d2 ) const{

return c < d2.c;

}

}a[ MAX ];

char s[ MAXN ];

int ans;

void solve( int len , string str ){

if( ans >= 100 ) return;

if( len == sz ){

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

ans++;

return;

}

int i , j;

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

int d1 = a[ i ].val , d2 = -1;

if( d1 > 9 ){

d1 = a[ i ].val / 10;

d2 = a[ i ].val % 10;

}

if( s[ len ] == d1 + '0' ){

if( d2 != -1 ){

if( len + 1 < sz && s[ len + 1 ] == d2 + '0' )

solve( len + 2 , str + a[ i ].c );

}

else solve( len + 1 , str + a[ i ].c );

}

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

if( len + 1 < sz && s[ len + 1 ] == d1 + '0' ){

if( d2 != -1 ){

if( len + 2 < sz && s[ len + 2 ] == d2 + '0' )

solve( len + 3 , str + a[ i ].c );

}

else solve( len + 2 , str + a[ i ].c );

}

}

}

}

int main(){

int i , q = 1;

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

for( i = 0 ; i < n && scanf(" %c %d" , &a[ i ].c , &a[ i ].val ) ; ++i );

scanf("%s" , s );

sz = strlen( s );

sort( a , a + n );

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

ans = 0;

solve( 0 , "" );

printf("\n");

}

return 0;

}

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

# Add All

\*\*\*ID: 10954

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

struct cmp {

bool operator() (const int &a, const int &b) {

return a > b;

}

};

int main(){

int n , x, ans ;

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

priority\_queue<int , vector<int> , cmp > Q;

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

scanf("%d" , &x );

Q.push( x );

}

ans = 0;

while( !Q.empty() ){

x = Q.top();Q.pop();

x += Q.top();Q.pop();

ans += x;

Q.push( x );

if( Q.size() == 1 )break;

}

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

}

return 0;

}

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

# The Party , Part 1

\*\*\*ID: 10959

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, BFS

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1005

vector<int> ady[ MAX ];

int dist[ MAX ];

void bfs( int x ){

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

queue<int> Q;

Q.push( x );

dist[ x ] = 0;

while( !Q.empty() ){

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

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

int w = ady[ u ][ v ];

if( dist[ w ] == -1 ){

dist[ w ] = dist[ u ] + 1;

Q.push( w );

}

}

}

}

int main(){

int t ,V ,E , u , v;

scanf("%d", &t);

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

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

scanf("%d %d", &V , &E);

while( E-- ){

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

ady[ u ].push\_back( v );

ady[ v ].push\_back( u );

}

bfs( 0 );

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

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

}

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

ady[ i ].clear();

}

}

return 0;

}

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

# Triangle Counting

\*\*\*ID: 10973

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 3005

vector<int> ady[ MAX ];

int graph[ MAX ][ MAX ];

int V;

int main(){

int t , E , u , v, cont;

scanf( "%d" , &t );

while( t-- ){

scanf("%d %d" , &V , &E );

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

cont = 0;

while( E-- ){

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

if( v > u )ady[ u ].push\_back( v ) , graph[ u ][ v ] = 1;

if( u > v )ady[ v ].push\_back( u ) , graph[ v ][ u ] = 1;

}

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

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

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

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

cont++;

}

}

}

}

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

for( int i = 1 ; i <= V ; ++i )ady[ i ].clear();

}

return 0;

}

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

# Enchanted Forest

\*\*\*ID: 10977

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory , BFS

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 205

int R , C , JJ;

int ady[ MAX ][ MAX ];

struct Jigly{

int x , y , L;

Jigly( int xx , int yy , int LL ): x( xx ) , y( yy ) , L( LL ){}

Jigly(){}

}J[ MAX ];

struct Estado{

int x , y , d;

Estado( int xx , int yy , int dd ): x( xx ) , y( yy ) , d( dd ){}

};

bool seen[ MAX ][ MAX ];

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

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

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

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

}

bool possible( int x , int y ){

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

if( dist( x , y , J[ i ].x , J[ i ].y ) <= J[ i ].L ) return false;

}

return true;

}

int bfs(){

queue<Estado> Q;

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

Q.push( Estado( 0 , 0 , 0 ) );

seen[ 0 ][ 0 ] = 1;

while( !Q.empty() ){

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

if( act.x == R - 1 && act.y == C - 1 ){

return act.d;

}

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

int nx = act.x + dx[ i ], ny = act.y + dy[ i ];

if( nx >= 0 & ny >= 0 && nx < R && ny < C && ady[ nx ][ ny ] == 0 && !seen[ nx ][ ny ] && possible( nx , ny )){

seen[ nx ][ ny ] = 1;

Q.push( Estado( nx , ny , act.d + 1 ) );

}

}

}

return -1;

}

int main(){

int x , y , d , l;

while( scanf("%d %d" , &R , &C ) , R|C ){

scanf("%d" , &d );

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

while( d-- ){

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

ady[ x - 1 ][ y - 1 ] = 1;

}

scanf("%d" , &JJ );

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

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

J[ i ] = Jigly( x - 1 , y - 1 , l );

}

d = bfs();

if( d == - 1 )printf("Impossible.\n");

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

}

return 0;

}

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

# Let's Play Magic!

\*\*\*ID: 10978

\*\*\*Juez: UVA

\*\*\*Tipo: Simulation

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

int n, idx;

char resp[ 53 ][ 21 ];

char card[ 5 ] , word[ 25 ];

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

vector<int> v;

for( int i = 0 ; i < n ; ++i ) v.push\_back( i );

idx = 0;

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

scanf("%s %s" , &card , &word );

idx = ( idx + strlen( word ) - 1 ) % v.size();

strcpy( resp[ v[ idx ] ] , card );

v.erase( v.begin() + idx );

}

printf("%s" , resp[ 0 ] );

for( int i = 1 ; i < n ; ++i )printf(" %s" , resp[ i ] );

putchar('\n');

}

return 0;

}

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

# Ignoring Digits

\*\*\*ID: 10993

\*\*\*Juez: UVA

\*\*\*Tipo: BFS

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define D 10

int n , digits[ D ] , len;

struct Estado{

string num;

int residuo;

Estado( string n , int res ): num( n ) , residuo( res ){}

};

#define MAX 100005

bool seen[ MAX ];

void bfs(){

queue< Estado > Q;

int i;

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

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

if( digits[ i ] ){

string s = "";

Q.push( Estado( s + (char)( digits[ i ] + '0') , digits[ i ] % n ) );

seen[ digits[ i ] % n ] = 1;

}

}

while( !Q.empty() ){

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

if( act.residuo == 0 ){

printf("%s\n" , act.num.c\_str() );

return;

}

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

int new\_residuo = ( act.residuo \* 10 + digits[ i ] ) % n;

if( !seen[ new\_residuo ] ){

seen[ new\_residuo ] = 1;

Q.push( Estado( act.num + (char)( digits[ i ] + '0' ), new\_residuo ) );

}

}

}

puts("impossible");

}

int main(){

long long f;

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

len = 0;

while( f > 0 ){

digits[ len++ ] = f % 10;

f /= 10;

}

bfs();

}

return 0;

}

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

# Simple Addition

\*\*\*ID: 10994

\*\*\*Juez: UVA

\*\*\*Tipo: Math , Summation

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

long long S( long long n ){

return ( ( n + 1 ) \* n )/2;

}

long long F( long long n ){

long long ans =0;

while( n > 9 ){

ans += S( n % 10 ) + (n/10) \* 45;

n /= 10;

}

return ans + S( n );

}

int main(){

long long a , b , rem , div , ans;

while( scanf("%lld %lld" , &a , &b ) , a > -1 && b > -1 ){

printf("%lld\n" , F( b ) - F( a - 1 ) );

}

return 0;

}

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

# Cheapest Base

\*\*\*ID: 11005

\*\*\*Juez: UVA

\*\*\*Tipo: Number Theory

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

#include <stdio.h>

int min( int a , int b ){ return ( a < b ) ? a : b; }

int cost[ 400 ];

int toBaseN( int n , int base ){

int sum = 0;

while( n > 0 ){

if( n % base > 9 ) sum += cost[ ( n % base - 10 ) + 'A' ];

else sum += cost[ n % base + '0' ];

n /= base;

}

return sum;

}

int main(){

int t , Q , n , mini, a[ 40 ];

scanf("%d" , &t );

for( int q = 1 ; q <= t ; ++q ){

if( q - 1 )putchar('\n');

for( int i = 0 ; i < 10 ; ++i ) scanf("%d" , &cost[ i + '0'] );

for( int i = 0 ; i < 26 ; ++i ) scanf("%d" , &cost[ i + 'A' ] );

scanf("%d" , &Q );

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

while( Q-- ){

scanf("%d" , &n );

mini = 1<<30;

for( int i = 2 ; i < 37 ; ++i ){

a[ i ] = toBaseN( n , i );

mini = min( mini , a[ i ] );

}

printf("Cheapest base(s) for number %d:" , n );

for( int i = 2 ; i < 37 ; ++i ){

if( a[ i ] == mini )printf(" %d" , i );

}

printf("\n");

}

}

return 0;

}

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

# Ferry Loading IV

\*\*\*ID: 11034

\*\*\*Juez: UVA

\*\*\*Tipo: Queue

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define LEFT 0

#define RIGHT 1

int main(){

int n , l , m , w, pos , viajes;

char s[ 6 ];

scanf("%d" , &n);

while( n-- ){

scanf("%d %d" , &l , &m );

l \*= 100;

queue<int> left, right;

while( m-- ){

scanf("%d %s" , &w , &s );

if( s[ 0 ] == 'l') left.push( w );

else right.push( w );

}

viajes = 0;

pos = LEFT;

while( !left.empty() || !right.empty() ){

w = 0;

if( pos == LEFT ){

while( !left.empty() ){

if( w + left.front() > l )break;

w += left.front();

left.pop();

}

}

else{

while( !right.empty() ){

if( w + right.front() > l )break;

w += right.front();

right.pop();

}

}

viajes++;

pos = !pos;

}

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

}

return 0;

}

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

# Building designing

\*\*\*ID: 11039

\*\*\*Juez: UVA

\*\*\*Tipo: Sorting

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

#include <stdio.h>

#include <algorithm>

#define MAX 500005

struct Building{

int value , color;

Building(){}

Building( int v , int c ): value( v ) , color( c ){}

}a[ MAX ];

bool cmp( Building b1 , Building b2 ){

return b1.value < b2.value;

}

int main(){

int t, x , n , cnt;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

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

scanf("%d" , &x);

if( x < 0 ) a[ i ] = Building( -x , 'R');

else a[ i ] = Building( x , 'A');

}

std::sort( a , a + n , cmp );

int color = a[ 0 ].color;

cnt = 1;

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

if( a[ i ].color != color ){

cnt++;

}

color = a[ i ].color;

}

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

}

return 0;

}

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

# Add bricks in the wall

\*\*\*ID: 11040

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <stdio.h>

#include <cstring>

#define MAX 10

int ady[ MAX ][ MAX ];

int a[ MAX ][ MAX ];

int main(){

int t;

scanf("%d" , &t );

while( t-- ){

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

for( int i = 0 ; i < 9 ; i += 2 ){

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

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

}

}

//impares

for( int i = 2 ; i < 9 ; i += 2 ){

for( int j = 1 ; j <=i ; j += 2 ){

ady[ i ][ j ] = ( ady[ i - 2 ][ j - 1 ] - ady[ i ][ j - 1 ] - ady[ i ][ j + 1 ])/2;

}

}

//pares

for( int i = 1 ; i < 9 ; i += 2 ){

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

ady[ i ][ j ] = ady[ i + 1 ][ j ] + ady[ i + 1 ][ j + 1 ];

}

}

for( int i = 0 ; i < 9 ; ++i ){

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

if( j )printf(" ");

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

}

printf("\n");

}

}

return 0;

}

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

# Searching for Nessy

\*\*\*ID: 11044

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc, Simple Math

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

#include <stdio.h>

#include <math.h>

int main(){

int t , n , m ;

scanf( "%d" , &t );

while( t-- ){

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

printf("%d\n", (int)ceil((n - 2)/3.0) \* (int)ceil((m - 2)/3.0));

}

return 0;

}

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

# My T-shirt suits me

\*\*\*ID: 11045

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Maximum Bipartite Matching

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

#include <stdio.h>

#include <cstring>

#define M 35

#define N 8

bool graph[M][N];

bool seen[N];

int matchL[M], matchR[N];

int shirt[ 6 ] , MM;

bool bpm( int u )

{

for( int v = 0; v < 6; v++ ) if( graph[u][v] )

{

if( seen[v] ) continue;

seen[v] = true;

if( shirt[ v ] > 0 ){

matchL[ u ] = v;

shirt[ v ]--;

return true;

}

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

if( matchL[ k ] == v && bpm( k ) )

{

matchL[ u ] = v;

return true;

}

}

}

return false;

}

int getId( char \*s ){

if( strcmp( s , "L" ) == 0 )return 0;

if( strcmp( s , "S" ) == 0 )return 1;

if( strcmp( s , "XL" ) == 0 )return 2;

if( strcmp( s , "XXL" ) == 0 )return 3;

if( strcmp( s , "XS" ) == 0 )return 4;

if( strcmp( s , "M" ) == 0 )return 5;

}

int main(){

int t , NN ;

scanf("%d" ,&t );

char s1[ 4 ] , s2[ 4 ];

while( t-- ){

scanf("%d %d" , &NN , &MM );

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

scanf("%s %s" , &s1 , &s2 );

graph[ i ][ getId( s1 ) ] = graph[ i ][ getId( s2 ) ] = 1;

}

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

shirt[ i ] = NN/6;

}

memset( matchL, 0xff , sizeof( matchL ) );

///HALLAMOS MATCHING

int i;

for( i = 0; i < MM; i++ )

{

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

if( !bpm( i ) )break;

}

if( i < MM )puts("NO");

else puts("YES");

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

}

return 0;

}

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

# The Scrooge Co Problem

\*\*\*ID: 11047

\*\*\*Juez: UVA

\*\*\*Tipo: Floyd Warshall

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

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAX 105

#define INF 1<<30

int ady[ MAX ][ MAX ], path[ MAX ][ MAX ], V;

void initPath(){

int i , j;

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

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

if( ady[ i ][ j ] == INF || ady[ i ][ j ] == 0 ) path[ i ][ j ] = 0;

else path[ i ][ j ] = i;

}

}

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 ][ k ] + ady[ k ][ j ] < ady[ i ][ j ] ){

ady[ i ][ j ] = ady[ i ][ k ] + ady[ k ][ j ];

path[ i ][ j ] = path[ k ][ j ];

}

}

}

}

}

char cities[ MAX ][ 35 ];

int numcities;

int get\_Id( char \*s ){

int i;

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

if( strcmp( s , cities[ i ] ) == 0 ) return i;

}

strcpy( cities[ numcities ] ,s );

return numcities++;

}

char \*getString( int id ){

int i;

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

if( id == i ) return cities[ i ];

}

}

char s[ 35 ] , s1[ 35 ] , s2[ 35 ];

void print( int x , int y ){

if( x == INF || y == INF ) return;

if( y == x ) { printf("%s" , getString( x ) ); }

else{

print( x , path[ x ][ y ] );

printf(" %s" , getString( y ) );

}

return;

}

int main(){

int t , i , x , j , u , v;

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &V );

numcities = 0;

for( i = 0 ; i < V && scanf("%s" , &s ) == 1 ; ++i ){ get\_Id( s ); }

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

for( j = 0 ; j < V && scanf("%d" , &ady[ i ][ j ] ) == 1 ; ++j ){

if( ady[ i ][ j ] == -1 ) ady[ i ][ j ] = INF;

}

}

initPath();

floyd();

scanf("%d" , &x );

while( x-- ){

scanf("%s %s %s" , s , s1 , s2 );

u = get\_Id( s1 ); v = get\_Id( s2 );

if( ady[ u ][ v ] == INF ) printf("Sorry Mr %s you can not go from %s to %s\n" , s , s1 , s2 );

else{

printf("Mr %s to go from %s to %s, you will receive %d euros\n" , s , s1 , s2 , ady[ u ][ v ] );

printf("Path:");

if( u == v ) printf("%s %s" , s1 , s2 );

else print( u , v );

printf("\n");

}

}

}

return 0;

}

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

# Automatic correction of misspellings

\*\*\*ID: 11048

\*\*\*Juez: UVA

\*\*\*Tipo: Strings

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

#include <stdio.h>

#include <cstring>

int main(){

int n , q;

scanf("%d" , &n );

char dict[10000][26], word[256];

for( int i = 0 ; i < n ; ++i ) scanf("%s" , &dict[ i ] );

scanf("%d" , &q );

while( q-- ){

scanf("%s" , &word );

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

if( strcmp( dict[ i ] , word ) == 0 ){

printf("%s is correct\n" , word );

goto done;

}

}

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

char \*p = dict[ i ], \*q = word;

while( \*p == \*q )p++, q++;

if( \*p && strcmp( p + 1 , q ) == 0 )goto found;

if( \*q && strcmp( p , q + 1 ) == 0 )goto found;

if( \*p && \*q && strcmp( p + 1 , q + 1 ) == 0 )goto found;

if( p[ 0 ] && p[ 1 ] && p[ 0 ] == q[ 1 ] && p[ 1 ] == q[ 0 ] && strcmp( p + 2 , q + 2 ) == 0 ) goto found;

continue;

found: printf("%s is a misspelling of %s\n", word, dict[i] );

goto done;

}

printf("%s is unknown\n", word);

done:;

}

return 0;

}

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

# Wine Trading in Gergovia

\*\*\*ID: 11054

\*\*\*Juez: UVA

\*\*\*Tipo: Greedy

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

long long absll( long long n ){ return (n < 0 )?-n : n ;}

int main(){

int n;

long long x , aux , ant, resp;

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

aux = 0;ant = 0;resp = 0;

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

scanf("%lld", &x );

if( i > 0 ){

resp += absll( aux );

}

aux = aux + x;

}

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

}

return 0;

}

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

# Formula 1

\*\*\*ID: 11056

\*\*\*Juez: UVA

\*\*\*Tipo: Sorting + String Comparison

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

struct Data{

string name , low;

double time;

int id;

Data(){}

bool operator < ( Data d2 ) const {

return ( time < d2.time ) || ( time == d2.time && low < d2.low );

}

}a[ MAX ];

int n;

char s[ MAX ];

bool isUpperCase(char c){return c>='A' && c<='Z';}

char toLowerCase(char c){return (isUpperCase(c))?(c+32):c;}

string toLow( string s ){

int i;

string ans = "";

for( i = 0 ; s[ i ] ; ++i ) ans += toLowerCase( s[ i ] );

return ans;

}

int main(){

int i , j , k;

double min , sec , ms;

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

gets( s );

for( i = 0 ; i < n && gets( s ) ; ++i ){

stringstream ss( s );

ss>>a[ i ].name>>s>>min>>s>>sec>>s>>ms>>s;

a[ i ].low = toLow( a[ i ].name );

a[ i ].time = ( min \* 60 + sec ) + ms \* 0.001;

}

sort( a , a + n );

for( i = 0 , j = 0 , k = 1 ; i < n ; ++i , j = ( j + 1 ) %2 ){

if( !j ) printf("Row %d\n" , k++ );

printf("%s\n" , a[ i ].name.c\_str() );

}

printf("\n");

}

return 0;

}

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

# Exact Sum

\*\*\*ID: 11057

\*\*\*Juez: UVA

\*\*\*Tipo: Search

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

#include <stdio.h>

#include <cstring>

#include <iostream>

using namespace std;

#define MAX 10005

#define MAXN 2000005

int a[ MAX ], seen[ MAXN ];

int main(){

int n , s , l , u , diff, L , U;

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

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

for( int i = 0 ; i < n ; ++i ) { scanf("%d" , &a[ i ] ); seen[ a[ i ] ]++; }

scanf("%d" , &s );

diff = 1<<30;

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

if( a[ i ] <= s && seen[ s - a[ i ] ] ){

l = a[ i ];

u = s - a[ i ];

if( l == u && seen[ u ] == 1 ) continue;

if( l > u ) swap( l , u );

if( diff > u - l ){

diff = u - l;

L = l;

U = u;

}

}

}

printf("Peter should buy books whose prices are %d and %d.\n\n" , L , U );

}

return 0;

}

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

# Beverages

\*\*\*ID: 11060

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Topological Sort

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

#include <stdio.h>

#include <vector>

#include <queue>

using namespace std;

#define MAX 105

char cities[ MAX ][ 40 ];

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 let[ MAX ];

struct node{

int x;

};

class comp{

public:

bool operator() ( const node &a, const node &b)

{

return a.x > b.x;

}

};

int main(){

int n , m , q = 1;

char s1[ 52 ], s2[ 52 ];

int inDegree [ MAX ], idx;

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

memset (inDegree, 0, sizeof( inDegree ) );

idx = 0;

numcities = 0;

vector <int> ady[ n + 1 ];

for( int i = 0 ; i < n ; ++i ){ scanf("%s" , cities[ i ] ); let[ idx++ ] = get\_Id( cities[ i ] ); }

scanf("%d" , &m );

while( m-- ){

scanf("%s %s" , s1 , s2 );

int b = get\_Id( s2 );

ady[ get\_Id( s1 ) ].push\_back( b );

inDegree [ b ]++;

}

printf("Case #%d: Dilbert should drink beverages in this order:" , q++ );

priority\_queue< node , vector<node > , comp > Q;

node t;

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

if ( inDegree [ i ] == 0 ){

t.x = i;

Q.push( t );

}

}

vector <int> sortedList;

while ( !Q.empty () ) {

t = Q.top();

Q.pop ();

int pop = t.x;

sortedList.push\_back ( pop );

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

inDegree [ ady[ pop ] [ i ] ]--;

if ( inDegree [ ady[ pop ] [ i ] ] == 0 )

{

node aux;

aux.x = ady[ pop ][ i ];

Q.push ( aux);

}

}

}

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

printf(" %s" , cities[ sortedList[ i ] ] );

}

printf(".\n\n");

}

return 0;

}

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

# Andy's Second Dictionary

\*\*\*ID: 11062

\*\*\*Juez: UVA

\*\*\*Tipo: Map

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int main(){

char s[ 1005 ];

int l;

string con = "";

while( gets( s ) ){

l = strlen( s );

if( l > 0 && s[ l - 1 ] == '-' ){

s[ l - 1 ] = '\0';

con += s;

}

else{

con += s;

con += " ";

}

}

l = con.length();

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

con[ i ] = tolower( con[ i ] );

if( !isalpha( con[ i ] ) && con[ i ] != '-' ) con[ i ] = ' ';

}

stringstream ss( con + " " );

map<string , int > mp;

map<string , int>::iterator it;

while( ss>>con ) mp[ con ] = 1;

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

printf("%s\n" , (\*it).first.c\_str() );

}

return 0;

}

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

# B2-Sequence

\*\*\*ID: 11063

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 105

int a[ MAX ], seen[ 20005 ],n;

bool b2(){

int i , j;

if( a[ 0 ] < 1 ) return false;

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

if( i && a[ i ] < a[ i - 1 ] ) return false;

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

if( seen[ a[ i ] + a[ j ] ] ){

return 0;

}

seen[ a[ i ] + a[ j ] ] = 1 ;

}

}

return true;

}

int main(){

int i , j , q = 1;

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

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

for( i = 0 ; i < n && scanf("%d" , &a[ i ] ) ; ++i );

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

if( b2() ) puts("It is a B2-Sequence.");

else puts("It is not a B2-Sequence.");

printf("\n");

}

return 0;

}

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

# Number Theory

\*\*\*ID: 11064

\*\*\*Juez: UVA

\*\*\*Tipo: Euler Totient - Num Divisors

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

long long countRelativelyPrime( long long n ) {

long long 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;

}

long long numDivisors( long long n ){

long long i , ans = 1 , cnt;

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

cnt = 0;

while( n % i == 0 ){

n /= i;

cnt++;

}

ans \*= ( cnt + 1 );

}

if( n > 1 ) ans \*= 2;

return ans - 1;

}

int main(){

long long n;

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

printf("%lld\n" , n - countRelativelyPrime( n ) - numDivisors( n ) );

}

return 0;

}

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

# Little Red Riding Hood

\*\*\*ID: 11067

\*\*\*Juez: UVA

\*\*\*Tipo: Counting Paths DP

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

#include <stdio.h>

#include <string.h>

#define MAX 105

int w, h , prohibited[ MAX ][ MAX ];

long long dp[ MAX ][ MAX ];

int dx[ 2 ] = { 0 , 1 };

int dy[ 2 ] = { 1 , 0 };

void solve(){

int i , j , k , nx , ny;

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

dp[ 0 ][ 0 ] = 1;

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

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

if( prohibited[ 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 && !prohibited[ nx ][ ny ] ){

dp[ nx ][ ny ] += dp[ i ][ j ];

}

}

}

}

if( dp[ h - 1 ][ w - 1 ] == 1 )

puts("There is one path from Little Red Riding Hood's house to her grandmother's house.");

else if( dp[ h - 1 ][ w - 1 ] > 1 )

printf("There are %lld paths from Little Red Riding Hood's house to her grandmother's house.\n" , dp[ h - 1 ][ w - 1 ] );

else puts("There is no path.");

}

int main(){

int n , u , v;

while( scanf("%d %d" , &h , &w ) , w | h ){

scanf("%d" , &n );

w++; h++;

memset( prohibited , 0 , sizeof( prohibited ) );

while( n-- ){

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

prohibited[ u ][ v ] = 1;

}

solve();

}

return 0;

}

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

# A graph Problem

\*\*\*ID: 11069

\*\*\*Juez: UVA

\*\*\*Tipo: Simple DP, recurrences

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

#include <stdio.h>

#define MAX 80

long long dp[ MAX ];

void Solve(){

dp[ 1 ] = 1;

dp[ 2 ] = 2;

dp[ 3 ] = 2;

dp[ 4 ] = 3;

dp[ 5 ] = 4;

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

dp[ i ] = dp[ i - 1 ] + dp [ i - 5 ];

}

}

int main(){

Solve();

int n;

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

printf("%lld\n", dp[ n ]);

}

return 0;

}

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

# Open Credit System

\*\*\*ID: 11078

\*\*\*Juez: UVA

\*\*\*Tipo: Ad hoc

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

#include <stdio.h>

int MIN( int a , int b ){ return ( a < b ) ? a : b; }

int MAX( int a , int b ){ return ( a < b ) ? b : a; }

int main(){

int t , n , x , maxi, ant, aux , m, a[ 100005 ],b[100005];

scanf("%d" , &t );

while( t-- ){

scanf("%d" , &n );

maxi = -10000000;

ant = 0;

aux = maxi;

m = maxi;

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

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

}

b[ n - 1 ] = a[ n - 1 ];

for( int i = n - 2 ; i >= 0 ; --i ){

b[ i ] = MIN( a[ i ] , b[ i + 1 ] );

}

for( int i = 0 ; i < n - 1 ; ++i ) maxi = MAX( maxi , a[ i ] - b[ i + 1 ] );

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

}

return 0;

}

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

# Place the Guards

\*\*\*ID: 11080

\*\*\*Juez: UVA

\*\*\*Tipo: Graph Theory, Bipartite Graph Check

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 205

vector< int > ady[ MAX ];

int V, numColor[ MAX ], seen[ MAX ] , color[ MAX ];

int MIN( int a , int b ){

if( a == 0 || b == 0 ) return 1;

return ( a < b )? a : b; }

int bfs( int origen ){

memset( numColor , 0 , sizeof( numColor ) );

queue< int > Q;

Q.push( origen );

int act;

color[ origen ] = 1;

while( !Q.empty() ){

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

numColor[ color[ act ] ]++;

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

if( !color[ ady[ act ][ i ] ] ){

color[ ady[ act ][ i ] ] = ( color[ act ] + 1 ) % 2;

if( color[ ady[ act ][ i ] ] == 0 ) color[ ady[ act ][ i ] ] = 2;

Q.push( ady[ act ][ i ] );

}

else if( color[ ady[ act ][ i ] ] == color[ act ] ){

return -1;

}

}

}

return 1;

}

int main(){

int t , E , u , v, ans;

scanf("%d" , &t );

while( t-- ){

scanf("%d %d" , &V , &E );

while( E-- ){

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

ady[ u ].push\_back( v );

ady[ v ].push\_back( u );

}

ans = 0;

memset( color , 0 , sizeof( color ) );

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

if( color[ i ] ) continue;

if( bfs( i ) == -1 ){

ans = -1;

break;

}

ans += MIN( numColor[ 1 ] , numColor[ 2 ] );

}

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

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

ady[ i ].clear();

}

}

return 0;

}

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

# Back to the 8-Queens

\*\*\*ID: 11085

\*\*\*Juez: UVA

\*\*\*Tipo: Backtracking

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

#include <stdio.h>

#include <cstring>

#define MAX 25

int row[ MAX ], leftDiag[ MAX ] , rightDiag[ MAX ], resp[ 100 ][ 10 ] , ans[ MAX ] , len;

void gen( int col ){

if( col == 8 ){

for( int i = 0 ; i < 8 ; ++i ) resp[ len ][ i ] = ans[ i ];

len++;

return;

}

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

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

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

ans[ col ] = i + 1;

gen( col + 1 );

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

}

}

}

int main(){

len = 0;

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

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

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

gen( 0 );

int a[ 9 ], sum , mini , q = 1;

while( scanf("%d%d%d%d%d%d%d%d" , &a[0] ,&a[1] , &a[2] , &a[3] , &a[4],&a[5],&a[6],&a[7] ) != EOF ){

mini = 1<<30;

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

sum = 0;

for( int j = 0 ; j < 8 ; ++j )

sum += ( a[ j ] == resp[ i ][ j ] )?0:1;

if( sum < mini ) mini = sum;

}

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

}

return 0;

}

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

# Composite Prime

\*\*\*ID: 11086

\*\*\*Juez: UVA

\*\*\*Tipo: Prime Factorization

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

#define MAX 1048580

//int a[ MAX ];

//#define MAX 3450

int prime[ MAX ], len;

long long 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 = 2 ; i < MAX ; ++i ) if( prime[ i ] ) p[ len++ ] = i;

}

bool composite[ MAX ];

void gen(){

int i , j;

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

for( j = i ; j < len ; ++j ){

if( p[ i ] \* p[ j ] >= MAX ) break;

composite[ p[ i ] \* p[ j ] ] = 1;

}

}

}

int main(){

sieve();

gen();

int n , i , x , ans;

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

ans = 0;

for( i = 0 ; i < n && scanf("%d" , &x ) ; ++i ) ans += composite[ x ];

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

}

return 0;

}

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

# Continents

\*\*\*ID: 11094

\*\*\*Juez: UVA

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

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

#include <bits/stdc++.h>

#include <cstdlib>

using namespace std;

int h , w;

#define MAX 22

char ady[ MAX ][ MAX ];

bool seen[ MAX ][ MAX ];

int cnt;

void dfs( int x , int y , char c ){

if( x > h || x < 0 || ady[ x ][ y ] != c || seen[ x ][ y ] ) return;

cnt++;

seen[ x ][ y ] = 1;

dfs( x + 1 , y , c );

dfs( x , ( y + 1 ) % w , c );

dfs( x - 1 , y , c );

dfs( x , ( w + ( y - 1 ) ) % w , c );

}

int main(){

int xini , yini;

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

for( int i = 0 ; i < h ; ++i ) scanf("%s" , ady[ i ] );

scanf("%d %d" , &xini , &yini );

int maxi = 0;

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

dfs( xini , yini , ady[ xini ][ yini ] );

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

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

if( !seen[ i ][ j ] && ady[ i ][ j ] == ady[ xini ][ yini ] ){

cnt = 0;

dfs( i , j , ady[ xini ][ yini ] );

maxi = ( maxi < cnt )? cnt : maxi;

}

}

}

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

}

return 0;

}