1. #include<bits/stdc++.h>
2. **using** **namespace** std;
4. #define mem(t, v) memset ((t) , v, sizeof(t))
5. #define all(x) x.begin(),x.end()
6. #define un(x) x.erase(unique(all(x)), x.end())
7. #define sf(n) scanf("%d", &n)
8. #define sff(a,b) scanf("%d %d", &a, &b)
9. #define sfff(a,b,c) scanf("%d %d %d", &a, &b, &c)
10. #define sl(n) scanf("%lld", &n)
11. #define sll(a,b) scanf("%lld %lld", &a, &b)
12. #define slll(a,b,c) scanf("%lld %lld %lld", &a, &b, &c)
13. #define D(x) cerr << \_\_LINE\_\_ << ": " << #x << " = " << (x) << '\n'
14. #define DD(x,y) cerr << \_\_LINE\_\_ << ": " << #x << " = " << x << " " << #y << " = " << y << '\n'
15. #define DDD(x,y,z) cerr << \_\_LINE\_\_ << ": " << #x << " = " << x << " " << #y << " = " << y << " " << #z << " = " << z << '\n'
16. #define DBG cerr << \_\_LINE\_\_ << ": Hi" << '\n'
17. #define pb push\_back
18. #define PI acos(-1.00)
19. #define xx first
20. #define yy second
21. #define eps 1e-9
23. typedef unsigned long long int ULL;
24. typedef long long int LL;
25. typedef pair<int,int> pii;
26. typedef pair<LL,LL> pll;

29. **inline** int setBit(int N, int pos) { return N=N | (1<<pos); }
30. **inline** int resetBit(int N, int pos) { return N= N & ~(1<<pos); }
31. **inline** bool checkBit(int N, int pos) { return (bool)(N & (1<<pos)); }
33. *//int fx[] = {+0, +0, +1, -1, -1, +1, -1, +1};*
34. *//int fy[] = {-1, +1, +0, +0, +1, +1, -1, -1}; //Four & Eight Direction*
36. LL pk[1010], inv[1010];
37. const LL MOD = 1000000007;
39. LL expo(int a, int n)
40. {
41. if(n == 0)
42. return 1;
43. if(n&1) return ( a \* expo(a, n-1))%MOD;
44. LL ret = expo(a, n/2);
45. return (ret \* ret)%MOD;
46. }
48. int main()
49. {
50. *// freopen("maxon.txt","r",stdin);*
51. *//freopen("out.txt","w",stdout);*
52. int n, k, t;
53. inv[0] = 1;
54. for(int i = 1; i<=1000; i++)
55. inv[i] = expo(i, MOD-2);
56. sf(t);
57. for(int cs = 1; cs<=t; cs++)
58. {
59. sff(n,k);
60. pk[0] = 1;
61. for(int i = 1; i<=n; i++)
62. pk[i] = (pk[i-1] \* k)%MOD;
63. LL ans = 0;
64. for(int i = 1; i<=n; i++)
65. {
66. ans += pk[\_\_gcd(i,n)];
67. if(ans >= MOD)
68. ans -= MOD;
69. }
70. ans \*= inv[n];
71. ans %= MOD;
72. printf("Case %d: %lld**\n**",cs,ans);
73. }
74. return 0;
75. }