1. #define \_CRT\_SECURE\_NO\_WARNINGS
2. #include<bits/stdc++.h>
3. #include <stdio.h>
4. #include <iostream>
5. #include <fstream>
6. #include <algorithm>
7. #include <iomanip>
8. #include <cmath>
9. #include <string>
10. #include <vector>
11. #include <deque>
12. #include <stack>
13. #include <set>
14. #include <map>
15. #include <queue>
16. #include <unordered\_map>
17. #include <bitset>
18. #include <numeric>
19. #include <cassert>
20. using namespace std;
21. #define ll long long
22. #define endl "\n"
23. #define cin(v) for(auto& i : v) cin >> i
24. #define cout(v) for(auto& i : v) cout << i
25. #define digits(n) fixed<<setprecision(n)
26. #define Go\_to\_the\_hell FAST();
27. #define sz(x) int(x.size())
28. #define pi acos(-1)
29. #define all(x) x.begin(), x.end()
30. #define rall(x) x.rbegin(), x.rend()
31. #define sin(x) (x\* pi / 180)
32. const ll INF = 1e18;
33. const int mod = 1e9 + 7;
34. const int N = 1e5+5;
35. int dx[] = { 0,0,1,-1 };
36. int dy[] = { 1,-1,0,0 };
37. int rr[] = { 1,1,1,-1,-1,-1,0,0 };
38. int cc[] = { 1,-1,0,1,-1,0,1,-1 };
39. ll fib(ll n) { return n < 2 ? 1 : fib(n - 1) + fib(n - 2); }
40. ll fact(ll n) { return (n == 0) ? 1 : n \* fact(n - 1); }
41. ll gcd(ll a, ll b) { return (a) ? gcd(b % a, a) : b; } // greatest common divisor
42. ll lcm(ll a, ll b) { return a / gcd(a, b) \* b; } //least common multiple
43. unsigned ll nCr(int n, int r) { if (r > n)return 0; r = max(r, n - r);
44. unsigned ll ans = 1, div = 1, i = (ll)r + 1; while (i <= n) { ans \*= i; i++; ans /= div; div++; }return ans; }
45. unsigned ll npr(int n, int r) { if (r > n)return 0;
46. unsigned ll a = 1, i = (ll)n - r + 1; while (i <= n) { a \*= i; i++; }return a; }
47. ll fastPower(ll base, ll power) { if (power == 0) { return 1; } if (power == 1) { return base % mod; }
48. return (((fastPower(base, power >> 1) \* fastPower(base, power >> 1)) % mod) \* fastPower(base, power & 1)) % mod; }
49. bool isprime[1000];
50. void sieve() {
51. isprime[1] = 1;
52. isprime[0] = 1;
53. for (ll i = 2; i \* i <= 1000; i++){
54. If (!isprime[i]) {
55. for (ll j = i + i; j <= 1000; j += i){
56. isprime[j] = 1;
57. }
58. }
59. }
60. }
61. void FAST() {
62. std::ios\_base::sync\_with\_stdio(0); cin.tie(NULL); cout.tie(NULL);
63. #ifndef ONLINE\_JUDGE
64. // freopen("input.txt", "r", stdin), freopen("output.txt", "w", stdout);
65. #endif
66. }
67. /\*\*
68. Think on paper not on PC Please -\_-
69. Pleaaaaase read the problem well and slowlllllllllllllllly..
70. Choose the smallest problem to solve first
71. Test each point in the problem
72. If it is math, think of gcd, lcm, primes, factorial, or power..
73. Dont use built in math fuctions directly in cout!!!
74. Try the samples on the idea first, then type the code
75. Don't take too much time through one problem if you didn't read all problems
76. Analyze the scoreboard to pick up the optimal problem to solve
77. Don't waste much time to follow the scoreboard.
78. \*\*/
79. void Run\_Case() {
80. }
81. int main() {
82. Go\_to\_the\_hell
83. int Tests = 1;
84. cin >> Tests;
85. while (Tests--)
86. Run\_Case();
87. return 0;
88. }