**GCDSEQ**

* dp[i][j] = ( 0 || 1) là số i có chia hết cho j hay không
* với mỗi truy vấn u v c ta sẽ cập nhật đoạn dp[u][c]++,

dp[v + 1][c]--;

sau đó cập nhật lại dp[i][j] += dp[i – 1][j]

ta sẽ có a[i] = lcm(j) với dp[i][j] > 0

* Ta sẽ dùng ctdl là segment tree để kiểm tra lại điều kiện ban đầu có thỏa mãn ko
* Đưa ra dãy a[i]

**SWAP**

-Ta sẽ cho xâu ban đầu lại thành xâu x1..x1x2..x2..xp..xp

Với theo thứ tự tần suất của các kí tự từ lớn đến bé

-Nếu số lượng x1 > n / 2 thì là không thể có đáp án

-Nếu sai thì xâu khớp vs xâu trên sẽ là x2..x2..xp..xpx1..x1

Ta có thể lưu thứ tự để dễ dàng lấy kết quả

x1..x1x2..x2..xp..xp

x2..x2..xp..xpx1..x1

ví dụ ở xâu abaabc ta sẽ đưa về xâu

aaabbc khớp với

bbcaaa

kết quả bài toán sẽ là babcaa

**SNP**

Sub1 : gọi f[i][j] là số cách để có được tổng i khi dùng nx số max nhất là j . f[i][j] = (f[i – 1][j] + f[i – j][j – 2]) % MOD

Sub2:

* Ta có nhật xét là số số hạng để tạo được tổng n là không quá căn n
* Gọi f[i][j] là số cách dc tổng i khi dùng j số
* Nếu a[1] = 1 thì f[i][j] = f[i - 2\* j + 1][j - 1]
* Nếu a[1] > 1 thì f[i][j] = f[i – 1][j]
* (f[i][j]+ = (f[i - j][j] + dp[i - 2\* j + 1][j - 1])) % MOD;

**SEED**

Ta sẽ tạo cây trie là những xâu thỏa mãn

* Ta sẽ xây dựng mảng nx[][] là khi thêm kí tự c vào xâu s có kết thúc là node thì sẽ dc xâu mới có điểm kết thúc là mấy
* Hàm Insert( ) tạo cây trie cho xâu str;
* Hàm Dfs(int node , string s)

Mỗi lần ta sẽ tạo mảng nx cho nx[node][]

* Hàm makenext() thêm kí tự c vào xâu s xóa kí tự đầu tiên cho đến khi tìm dc xâu thỏa mãn trên trie
* f[i][node] là số cách xây đến i và chứa xâu có điểm kết thúc là node : f[i + 1][nx[node][c]] += f[i][node];

kết quả bài toán là f[n][i] với i là những node của kí tự cuối cùng của xâu thỏa mãn