IUST - 8 silandr - Notebook

Contents

1 Template - SublimeConfig

1.1 Sublime Config

```
"shell_cmd": "g++ ${file_name} && gnome-terminal -- bash -c './a.out;
    read'"
```

1.2 Template

```
#pragma GCC optimize("Ofast,unroll-loops,fast-math")
#include<bits/stdc++.h>
using namespace std;
typedef long long ll ;
#define pll pair<ll , ll >
#define all(x) (x).begin(),(x).end()
#define SZ(x) (ll)(x).size()
#define X   first
#define Y   second
#define mp  make_pair
#define pii pair<int , int>
#define vec vector
#define file_io freopen("input.txt", "r", stdin);freopen("output.txt",
     "w", stdout);
#define migmig ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);
#define pb push_back
#define ld long double
// BIG p : 1000000000000037 , 100000000003
ll poww(ll a, ll b, ll md) {
    return (!b ? 1 : (b & 1 ? a * poww(a * a % md, b / 2, md) % md :
        poww(a * a % md, b / 2, md) % md));
}
const int maxn = 1000*100+5 ;
const ll inf = 9223372036854775807 ;
const ll mod = 1e9 + 7 ;
const int lg = 20 ;

int main()
{
        migmig ;

}
```

2 Implementation

2.1 Token

```
vector<string> token(string& s, string& del)
{
    // string inp = "Hello,World;This|is.GeeksForGeeks pam";
    // string del = ",;.| ";
    // vector<string> ans = token(inp, del);
    vector<string> ans;
    int start = 0, last = 0;
    while ((last = s.find_first_of(del, start)) != string::npos)
        {
        if (last != start)
            ans.pb(s.substr(start, last - start));
        start = last + 1;
    }
    if (start != s.length())
        ans.pb(s.substr(start));
    return ans;
}
```

2.2 Convert(String-Double)

```
double d = 3.0;
string str = to_string(d);
double D = stod(str)
```

3 Number Theory

3.1 Most Divisors

```
<= 1e2: 60 with 12 divisors
<= 1e3: 840 with 32 divisors
<= 1e4: 7560 with 64 divisors
<= 1e5: 83160 with 128 divisors
<= 1e6: 720720 with 240 divisors
<= 1e7: 8648640 with 448 divisors
<= 1e8: 73513440 with 768 divisors
<= 1e9: 735134400 with 1344 divisors
<= 1e10: 6983776800 with 2304 divisors
<= 1e11: 97772875200 with 4032 divisors
<= 1e12: 963761198400 with 6720 divisors
<= 1e13: 9316358251200 with 10752 divisors
<= 1e14: 97821761637600 with 17280 divisors
```

```
<= 1e15: 866421317361600 with 26880 divisors
<= 1e16: 8086598962041600 with 41472 divisors
<= 1e17: 74801040398884800 with 64512 divisors
<= 1e18: 897612484786617600 with 103680 divisors

//number of primes
30: 10
60: 17
100: 25
1000: 168
10000: 1229
100000: 9592
1000000: 78498
10000000: 664579
```

4 String

4.1 Z-Function

```
const ll maxn = 2e5 + 10;
ll z[maxn];
//  Pattern matching : maximum character matching of start at index i
    to prefix
//  text && pattern -----> s = pattern + '$' + text
// z[i] = number of match prifix start index i

void z_function(string s)
{
    ll left = 0, right = 0;
    for(int i = 1; i < s.size(); i++)
        if(i <= right && z[i - left] < right - i + 1)
            z[i] = z[i - left];
        else
        {
            if(i <= right)
                left = i;
            else
                left = right = i;
            while(right < s.size() && s[right] == s[right - left])
                right++;
            z[i] = right - left;
            right--;
        }
}
```

5 Graph

5.1 2sat

```
//harki mesle i khodesh 2 * i notesh 2 * i + 1 he
//yale u -> v yani agar u yek baseh v ham bayad yek bashe
vector<int> out[maxn * 2] , in[maxn * 2] , topol , adj[maxn];

bool visited[maxn * 2];

int color[maxn * 2] , c , val[maxn] , r[maxn] ;

void add_edge(int v , int u)
{
        out[v].pb(u);
        in[u].pb(v);
}

void dfs(int v)
{
        visited[v] = 1;
        for(auto u : out[v])
                if(!visited[u])
                        dfs(u);
        topol.pb(v);
}

void sfd(int v)
{
        visited[v] = 1;
        color[v] = c;
        for(auto u : in[v])
                if(!visited[u])
                        sfd(u);
}
for(int i = 0; i < 2 * n; i++)
                if(!visited[i])
                        dfs(i);
reverse(topol.begin() , topol.end());
memset(visited , 0 , sizeof visited);
for(int v = 1 ; v <= 2 * m + 1 ; v ++ )
                if(!visited[v])
                        sfd(v) , c++;
        for(int i = 1; i <= m; i++)
                if(color[2 * i] == color[2 * i + 1])
                        return cout << "NO" << endl , 0;
```