Problem Statement: Array Pairs
Solution:

C++:

```
#include <bits/stdc++.h>
using namespace std;

#define rep(i,n) for(int i=0;i<n;i++)
#define ll long long int
#define f first
#define s second
#define pi pair<ll,ll>
#define pii pair<pi,ll>
#define f first
#define s second
#define pb push_back
#define mod 1000000007
#define mp make_pair
#define pb push_back
#define rep(i,n) for(int i=0;i<n;i++)

int N;
int A[1000011];
int L[1000011];
int R[1000011];
vector<int>g[1000011];
ll bt[1000011];
int maxn;
void update(int ind, int val) {
   while(ind <= maxn) {
      bt[ind] += val;
      ind += (ind & -ind);
   }
}
ll query(int ind) {
   ll ans = 0;
   while(ind > 0) {
      ans += bt[ind];
      ind -= (ind & -ind);
   }
   return ans;
}
vector<int>V;
int find_ind(int x) {
   if(V.back() <= x) return V.size();
   return upper_bound(V.begin(), V.end(), x) - V.begin();
```

```
}
int main() {
   ios_base::sync_with_stdio(0);
   cin.tie(0);

   cin >> N;
   set<int>S;
   unordered_map<int, int>M;

   for(int i = 1; i <= N; i++) {
      cin >> A[i];
      assert(A[i] >= 1 and A[i] <= 1000000000);
      S.insert(A[i]);
   }
   vector<pi>window;
   for(int i = 1; i <= N; i++) {
      while(window.size() > 0 and window.back().f < A[i]) window.pop_back();
      if(window.size() == 0) L[i] = 1;
      else {
         L[i] = window.back().s + 1;
      }
      window.pb(mp(A[i], i));
   }
   window.clear();
   for(int i = N; i >= 1; i--) {
      while(window.size() > 0 and window.back().f <= A[i]) window.pop_back();
      if(window.size() == 0) R[i] = N;
      else {
         R[i] = window.back().s - 1;
      }
      window.pb(mp(A[i], i));
   }

   for(int i = 1; i <= N; i++) {
      if(i - L[i] <= R[i] - i) {
         for(int j = L[i]; j < i; j++) {
            g[i - 1].pb(-A[i] / A[j]);
            g[R[i]].pb(A[i] / A[j]);
            //S.insert(A[i]/A[j]);
         }

         g[i].pb(-1);
         g[R[i]].pb(1);
      } else {

         for(int j = i + 1; j <= R[i]; j++) {
            g[L[i] - 1].pb(-A[i] / A[j]);
```

```
            g[i].pb(A[i] / A[j]);
            //S.insert(A[i]/A[j]);
        }

        g[L[i] - 1].pb(-1);
        g[i - 1].pb(1);
    }
  }
  maxn = S.size() + 2;
  int cnt = 1;
  for(set<int>::iterator it = S.begin(); it != S.end(); it++) {
    M[*it] = cnt++;
  }
  ll ans = 0;
  int r;
  V = vector<int>(S.begin(), S.end());
  for(int i = 1; i <= N; i++) {
    update(M[A[i]], 1);
    for(int j = 0; j < g[i].size(); j++) {
      r = find_ind(abs(g[i][j]));
      if(g[i][j] < 0) {
        ans -= query(r);
      } else {
        ans += query(r);
      }
    }
  }
  cout << ans;
}
```

By: parkhiapurva2

Problem Statement: Cycle Detection
Solution:

C++:

```
bool has_cycle(SinglyLinkedListNode* head) {
SinglyLinkedListNode *p1 = head,*p2 = head;

  while(p2!=NULL && p2->next != NULL)
  {
    p1 = p1->next;
    p2 = p2->next->next;
    if(p1 == p2 )  return true;
  }
  return false;
}
```

By: kaiwalya_aney

Problem Statement: Find the Path 6
Solution:

C++ :

```
#include <bits/stdc++.h>

using namespace std;
#define pii pair<int,pair<int,int>>
int helper(vector<vector<int>>& grid)
   {
      // Code here
      vector<vector<int>> moves={{0,-1},{0,1},{1,0},{-1,0}};
      int dp[1001][1001];
      for(int i=0; i<1001; i++)
      for(int j=0; j<1001; j++)
         dp[i][j] = INT_MAX;
      dp[0][0]=grid[0][0];
      priority_queue<pii,vector<pii>,greater<pii>> pq;
      pq.push({grid[0][0],{0,0}});
      int n=grid.size();
      while(pq.size()){
         auto node=pq.top();
         pq.pop();
         int x=node.second.first;
         int y=node.second.second;
         for(int k=0;k<4;k++){
            int i=x+moves[k][0];
            int j=y+moves[k][1];
            if(i>=0 && i<n &&j>=0 && j<n && dp[x][y]+grid[i][j]<dp[i][j]){
               dp[i][j]=dp[x][y]+grid[i][j];
               pq.push({dp[i][j],{i,j}});
            }
         }
      }
      return dp[n-1][n-1];
}

int main()
{
   // Write your Program Here
   int n;
   cin>>n;
   vector<vector<int>> mat(n,vector<int>(n));
   for(int i=0;i<n;i++) for(int j=0;j<n;j++) cin>>mat[i][j];
```

```
    cout<<helper(mat);

    return 0;
}
```
By: sudarshanmaskare