* **[Problem](https://www.hackerrank.com/challenges/winning-lottery-ticket/problem)**
* [Submissions](https://www.hackerrank.com/challenges/winning-lottery-ticket/submissions)
* [Leaderboard](https://www.hackerrank.com/challenges/winning-lottery-ticket/leaderboard)
* [Discussions](https://www.hackerrank.com/challenges/winning-lottery-ticket/forum)
* [Editorial](https://www.hackerrank.com/challenges/winning-lottery-ticket/editorial)

The SuperBowl Lottery is about to commence, and there are several lottery tickets being sold, and each ticket is identified with a ticket ID. In one of the many winning scenarios in the Superbowl lottery, a winning pair of tickets is:

* Concatenation of the two ticket IDs in the pair, in any order, contains each digit from  to  at least once.

For example, if there are  distinct tickets with ticket ID  and ,  is a winning pair.

NOTE: The ticket IDs can be concantenated in any order. Digits in the ticket ID can occur in any order.

Your task is to find the number of winning pairs of distinct tickets, such that concatenation of their ticket IDs (in any order) makes for a winning scenario. Complete the function winningLotteryTicket which takes a string array of ticket IDs as input, and return the number of winning pairs.

**Input Format**

The first line contains  denoting the total number of lottery tickets in the super bowl.  
Each of the next  lines contains a string, where string on a  line denotes the ticket id of the  ticket.

**Constraints**

* length of
* sum of lengths of all
* Each ticket id consists of digits from

**Output Format**

Print the number of pairs in a new line.

**Sample Input 0**

5

129300455

5559948277

012334556

56789

123456879

**Sample Output 0**

5

**Explanation 0**

Pairs of distinct tickets that make for a winning scenario are :

| **Ticket ID 1** | **Ticket ID 2** | **Winning Pair** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Notice that each winning pair has digits from  to  atleast once, and the digits in the ticket ID can be of any order. Thus, the number of winning pairs is .

#include<bits/stdc++.h>

#define pp pop\_back

#define pb push\_back

#define int long long int

#define INF 1e18

#define vec vector<int>

#define pii pair<int,int>

#define REP(i,a,b) for(i=a;i<b;i++)

using namespace std;

int32\_t main()

{

  ios\_base::sync\_with\_stdio(false);

  cin.tie(NULL);

  cout.tie(NULL);

  int t=1;

  //cin>>t;

  while(t--)

  {

    int n,i;

    cin>>n;

    string s[n];

    set<int> v[n];

    map<set<int>,int>m;

    for(i=0;i<n;i++)

    {

      for(int j=0;j<s[i].length();j++)

        v[i].insert((int)s[i][j]-48);

        m[v[i]]++;

    }

  }

}