Problem: Cavity Map

You are given a square map of size . Each cell of the map has a value denoting its depth. We will call a cell of the map a *cavity* if and only if this cell is not on the border of the map and each cell adjacent to it has *strictly smaller depth*. Two cells are adjacent if they have a common side (edge).
You need to find all the cavities on the map and depict them with the uppercase character **X**.

Input Format

The first line contains an integer, , denoting the size of the map. Each of the following lines contains positive digits without spaces. Each digit (1-9) denotes the depth of the appropriate area.

Constraints

Output Format

Output lines, denoting the resulting map. Each cavity should be replaced with character `X`.

Sample Input

```
4
1112
1912
1892
1234
```

Sample Output

```
1112
1X12
18X2
1234
```

Explanation

The two cells with the depth of 9 fulfill all the conditions of the Cavity definition and have been replaced by X.

Solution:

```
int main()
  {
      int size;
      cin>>size;
      string map[size];
      for(long i=0; i<size; i++)
        {
          cin>>map[i];
        }

       /*processing the map*/
       for(long i=0; i<size; i++)
         {
           for(long j=0; j<size; j++)
            {
              if(i!=0 && i!=size-1 && j!=0 && j!=size-1)
               {
                 long temp=0;
                 long depth=map[i][j];
                 /*check for cavity*/
                 (map[i+1][j]<depth && 0<=i && i<size && j<size && 0<=j ? temp+=1: temp+=0);
                 (map[i-1][j]<depth && 0<=i && i<size && j<size && 0<=j ? temp+=1 :temp+=0);
                 (map[i][j+1]<depth && 0<=i && i<size && j<size && 0<=j ? temp+=1 :temp+=0);
                 (map[i][j-1]<depth && 0<=i && i<size && j<size && 0<=j ? temp+=1 :temp+=0);

                  if(temp==4)
                  {map[i][j]='X';}
              }
             cout<<map[i][j];
            }
           cout<<endl;
         }
        return 0;
       }
```

`’Anshul AgGarwal