CODE RANCH

Introduction

Code Ranch is Ureckon's flagship event, pitting the best competitive programmers against each other in an epic battle of wits. Coding Contest consists of question related to logic, mathematics, data structures and algorithms.

Solve, Code and Win

Objective

Code Ranch is the main competitive coding event of Ureckon where participants need to solve a given problem and write the code in their preferred language to solve it correctly and in the shortest amount of time possible to win the competition.

About The Event

Maximum Team Size: 2 member (Solo participation is also allowed)
Duration : 3 hrs
Platform used: CodeChef
Ranking mechanism used: ICPC style
Contest type: Closed

Rules & Regulations:

- All participants must carry their college ID card.
- The use of electronic gadgets during the competition is strictly prohibited.
- Participants are requested to be present at the venue(will be informed later) at least 15-20 minutes prior to the commencement of the event.
- The use of Internet for any unfair means or unfair means of any sort will lead to direct disqualification.
- Co-coordinators decision will be final in all events regarding the competition.

SAMPLE:

Here is a sample problem and a possible solution written in C language.

Problem:

Many of you have played PUBG(relax! question is not specifically about PUBG). We have a player of PUBG with us Prateek usually, other players call Prateek as "**BOT KILLER**" and he gets annoyed by this so he decided to win the next game of PUBG.

Prateek managed to survive till the end and lost his all ammunition and just one player(COSMIC) is left who is having far better RP then Prateek(good player then Prateek). Luckily COSMIC's ammunition is also over except one grenade(multiple explosion grenade). COSMIC has seen Prateek and opens voice chatting and ask Prateek if he will able to save himself from the grenade then COSMIC will surrender himself. Help Prateek to solve the riddle. The behavior of the grenade is described with a recursion depth n and a duration for each level of recursion t_1, t_2, ..., t_n. Once COSMIC launches the grenade in some cell(assume both are in a grid), the grenade starts moving upward. After covering $t_1$cells (including the starting cell), it explodes and splits into two parts, each moving in the direction changed by 45 degrees (see the pictures below for clarification). So, one part moves in the top-left direction, while the other one moves in the top-right direction. Each part explodes again after covering t_2 cells, splitting into two parts moving in directions again changed by 45 degrees. The process continues till the n-th level of recursion when all 2_{n-1} existing parts explode and disappear without creating new parts. After a few levels of recursion, it's possible that some parts will be at the same place and at the same time — it is allowed and such parts do not crash.

Before launching a grenade, Prateek must make sure that he doesn't stand in cells which will be visited at least once by the grenade. Prateek has to tell the count of those cells.

Input:

The first line will be n and the second line will be n-space separated numbers.

Output:

Print one integer, denoting the number of cells which will be visited at least once by any part of the Grenade.

Constraints:

1 ≤ n ≤ 30

1 ≤ t_i ≤ 5

Sample Input:

```
4
4 2 2 3
```

Sample Output:

```
39
```

Sample Explanation:

COSMIC launched the firework from the bottom-most red cell. It covered t_1 = 4 cells (marked red), exploded and divided into two parts (their further movement is marked green). let denote different colors for next set of explosions, there are 4 red, 4 green, 8 orange, and 23 pink cells. So, the total number of visited cells is 4 + 4 + 8 + 23 = 39

Solution:

```
#include<bits/stdc++.h>
using namespace std;
bool a[400][400],b[32][400][400][8];
int t,dx[8]={1,1,0,-1,-1,-1,0,1},dy[8]={0,1,1,1,0,-1,-1,-1};
int main()
{
        int T;
        cin>>T;
        int n,i=0,t=0,d,x,y,z,dd,xx,yy;
        b[0][200][200][0]=1;
        for(scanf("%d",&n);i<n;i++)
        {
                scanf("%d",&d);
                for(x=0;x<400;x++)
                for(y=0;y<400;y++)
                for(z=0;z<8;z++)
                if(b[i][x][y][z])
                {
                    for(dd=0,xx=x,yy=y;dd<d;dd++)
                    {
                        xx+=dx[z],yy+=dy[z];
                        if(!a[xx][yy]) t++;
                        a[xx][yy]=1;
                    }
                    b[i+1][xx][yy][(z+1)&7]=b[i+1][xx][yy][(z+7)&7]=1;
                }
        }
        printf("%d\n",t);
        return 0;
}
```

Coordinators:

Jagjeet Kaur Randhawa (+91 7278722839)
Sagnik Sengupta (+91 9733197523)