```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>
#define N 100                  // should be N 512 or more;  Rule encoding deficiencies
char map[N][N];2
void out(int n)
{
   int i,j,t;
   t=8*n-2;                    //* should be: t=int(pow(2.0,n+2)-2); Difficulty with exponential development
   for(i=0;i<=t+1;i++,printf("\n"))
      for(j=0;j<=t+1;j++)
         printf(map[i][j]);
}
int main()
{
   int i,j,n,T,t,di,dj,level;
   scanf("%d",&T);
   while(T--)
   {
      scanf("%d",&n);
      memset(map,'0',sizeof(map));              // * '0' should be ‘ ’ First exception, strong-but-now-wrong
      di=0,dj=0;
      for(level=n;level>=0;level--)
      {
         t=8*n-2;                    //* should be t=int(pow(2.0,n+2)-2);
         map[di][dj]='+';
         for(j=1;j<=t;j++) {
         map[di][dj+j]=2;map[di][dj+t+1]='+';
         }
```

```
        for(i=1;i<=t;i++) {
         map[di+i][dj]=map[di+i][dj+t+1]='!';                //*  '!' should be '|'          perceptual confusion
        }
        map[di+t+1][dj]=map[di+t+1][dj+(t+1)]='+';
        for(j=1;j<=(t/2+1)/2;j++){
           map[di+t+1][dj+j]=map[di+t+1][dj+(t+1)-j]='-';
        }

        if(level!=0)
        {
           for(i=2;i<t/2;i++)
           {
              map[di+i][dj+t/2+1-i]='/';
              map[di+i][dj+t/2+i]='\\';
           }
           di=di+t/2+1;
           dj=dj+(t/2+1)/2;
        }
        else {
         break;
        }

     }
     out(n);                    // missed printf("\n");              Post-completion error
  }
  return 0;
}
```