31TH DECEMBER

First Question:-

Geek is very fond of patterns. Once, his teacher gave him a pattern to solve. He gave Ram an integer n and asked him to build a pattern.

Help Ram build a pattern.

Example 1:

```
Input: 5

Output:
    *
   ***       *****
  *******
*********
Your Task:
```

You don't need to input anything. Complete the function **printTriangle()** which takes an integer **n** as the input parameter and print the pattern.

CODE SECTION:-

```
//first question
//{ Driver Code Starts
#include <bits/stdc++.h>
 using namespace
std;

// } Driver Code Ends
class Solution {
public:
    void printTriangle(int n) {
        // code here

                          for(int
i=n,k=0;i>0;i--,k++){
```

```
            for(int j=i-1;j>0;j--){
cout<<" ";
            }             for(int
m=0;m<2*k+1;m++){
cout<<"*";
            }
            cout<<endl;

        }


    }
};

//{ Driver Code Starts.
 int main() {
int t;     cin
>> t;     while
(t--) {
int n;
cin >> n;

        Solution ob;
        ob.printTriangle(n);
    }
    return 0;
}
// } Driver Code Ends
```

Second Question:-

Geek is very fond of patterns. Once, his teacher gave him a pattern to solve. He gave Geek an integer n and asked him to build a pattern.

Help Geek to build a pattern.

Example 1:

```
Input: 5

Output:

*********
 *******
  *****
   ***
    *
```

Your Task:

You don't need to input anything. Complete the function **printTriangle()** which takes an integer **n** as the input parameter and print the pattern.

Constraints:

- 1<= **N** <= 20

CODE SECTION:-

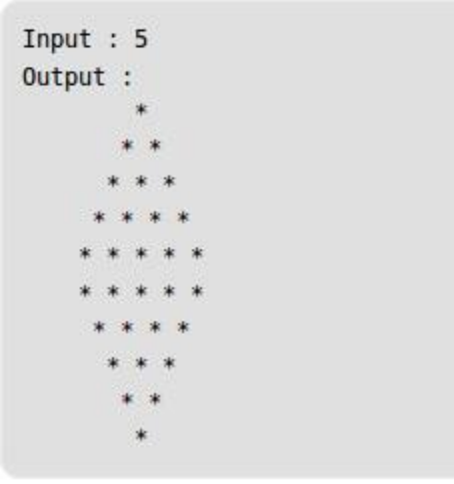
```
//{ Driver Code Starts
#include <bits/stdc++.h>
 using namespace
std;

// } Driver Code Ends class
Solution{ public:
void printTriangle(int n) {
        // code here
               for(int
i=0,s=n;i<n;i++,s--){
                     for(int
k=0;k<i;k++){
cout<<" ";
          }
for(int j=2*s-1;j>0;j--){
cout<<"*";
          }
cout<<endl;

      }

    }
};

//{ Driver Code Starts.

int main() {
int t;     cin
>> t;     while
(t--) {
int n;
cin >> n;

        Solution ob;
        ob.printTriangle(n);
    }
    return 0;
}
// } Driver Code Ends
```

THIRD QUESTION:-

Geek is very fond of patterns. Once, his teacher gave him a star pattern to solve. He gave Geek an integer n and asked him to build a pattern.

Help Geek to build a star pattern.

Example 1:



```
Input : 5
Output :
         *
        * *
       * * *
      * * * *
     * * * * *
     * * * * *
      * * * *
       * * *
        * *
         *
```

Your Task:

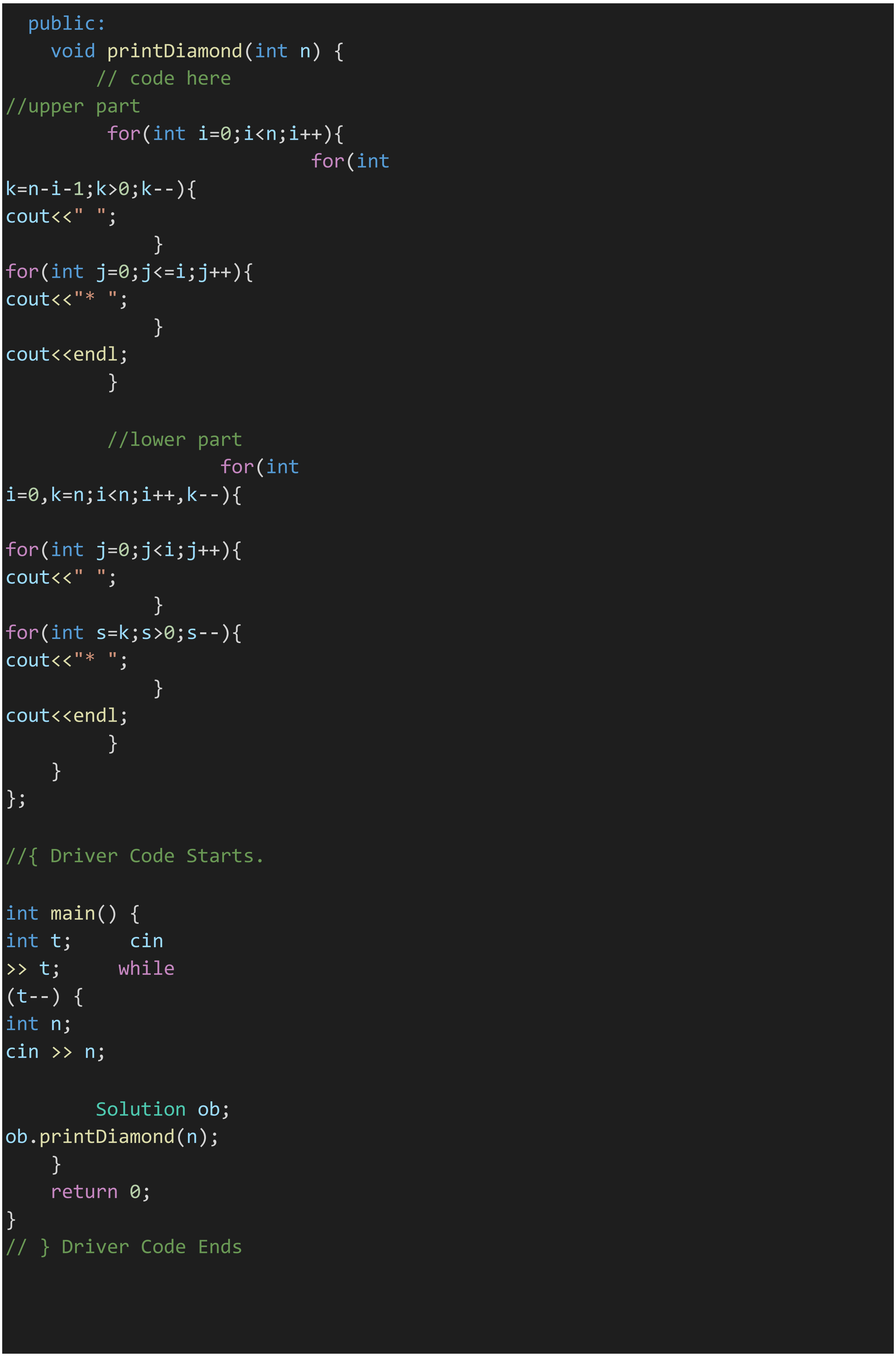
You don't need to input anything. Complete the function **printDiamond()** which takes an integer n as the input parameter and print the pattern.

Constraints:

- 1<= **N** <= 20 **CODE**

SECTION:-

```
//{ Driver Code Starts
#include <bits/stdc++.h>
 using namespace
std;

// } Driver Code Ends
class Solution {
```



```
  public:
    void printDiamond(int n) {
        // code here
//upper part
         for(int i=0;i<n;i++){
                           for(int
k=n-i-1;k>0;k--){
cout<<" ";
             }
for(int j=0;j<=i;j++){
cout<<"* ";
             }
cout<<endl;
         }

         //lower part
                   for(int
i=0,k=n;i<n;i++,k--){

for(int j=0;j<i;j++){
cout<<" ";
             }
for(int s=k;s>0;s--){
cout<<"* ";
             }
cout<<endl;
         }
    }
};

//{ Driver Code Starts.

int main() {
int t;     cin
>> t;     while
(t--) {
int n;
cin >> n;

        Solution ob;
ob.printDiamond(n);
    }
    return 0;
}
// } Driver Code Ends
```

FOURTH QUESTION:-

Geek is very fond of patterns. Once, his teacher gave him a pattern to solve. He gave Geek an integer n and asked him to build a pattern.

Help Geek to build a star pattern.

Example 1:

```
Input: 5

Output:
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Your Task:

You don't need to input anything. Complete the function **printTriangle()** which takes an integer **n** as the input parameter and print the pattern.

Constraints:

- 1<= **N** <= 20

CODE SECTION:-

```
//{ Driver Code Starts
#include <bits/stdc++.h>
 using namespace
std;

// } Driver Code Ends
//Back-end complete function Template for
C++ class Solution{ public:
    void printTriangle(int n) {
        // code here
for(int i=0;i<n;i++){
for(int j=0;j<=i;j++){
cout<<"* ";
            }
            cout<<endl;
        }
for(int i=n;i>1;i--){
                           for(int
j=i-1;j>0;j--){
cout<<"* ";
        }
cout<<endl;
        }
    }
};

//{ Driver Code Starts.

int main() {
int t;     cin
>> t;     while
(t--) {
int n;
cin >> n;

        Solution ob;
        ob.printTriangle(n);
    }
    return 0;
}
// } Driver Code Ends
```

FIFTH QUESTION:-

Find minimum number of Laptops required

There are **N** jobs and the start and finish time of the jobs are given in arrays **start[]** and **end[]** respectively. Each job requires one laptop and laptops can't be shared. Find the minimum number of laptops required given that you can give your laptop to someone else when you are not doing your job.

Example 1:

```
Input: N = 3
start[] = {1, 2, 3}
end[] = {4, 4, 6}
Output:
3
Explanation:
We can clearly see that everyone's supposed to
be doing their job at time 3. So, 3 laptops will
be required at minimum.
```

Example 2:

```
Input: N = 3
start[] = {1, 5, 2}
end[] = {2, 6, 3}
Output :
1
Explanation:
All jobs can be done using 1 laptop only.
```

Your Task:
You don't need to read input or print anything. Your task is to complete the function **minLaptops()** which takes an integer N and two arrays start and

end denoting starting and ending time of N jobs and returns minimum laptops required.

Expected Time Complexity: O(N*logN)
Expected Auxiliary Space: O(N)

Constraints:

1 ≤ N ≤ 10^5

1 ≤ start[i] < end[i] ≤ 10^9

CODESECTION:-

```
//{ Driver Code Starts
// Initial Template for C++

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

// } Driver Code Ends
//User function Template for C++

class Solution {
public:
    int minLaptops(int N, int start[], int end[]) {
        // Code here
int minlappy=0;
int count=0;
vector<pair<int,int>>v;
               for(int
i=0;i<N;i++){
           v.push_back({start[i],1});
           v.push_back({end[i],-
1});           }
sort(v.begin(),v.end());
for(auto i:v){
count=count+i.second;
```

```
           minlappy=max(count,minlappy);
       }

return minlappy;


    }
};

//{ Driver Code Starts.
 int main() {     int t;
cin >> t;     while (t--) {
int N;         cin >> N;
int start[N], end[N];
for(int i=0; i<N; i++)
cin>>start[i];
for(int i=0; i<N; i++)
cin>>end[i];

        Solution ob;         cout <<
ob.minLaptops(N, start, end) << endl;
    }
}
// } Driver Code Ends
```