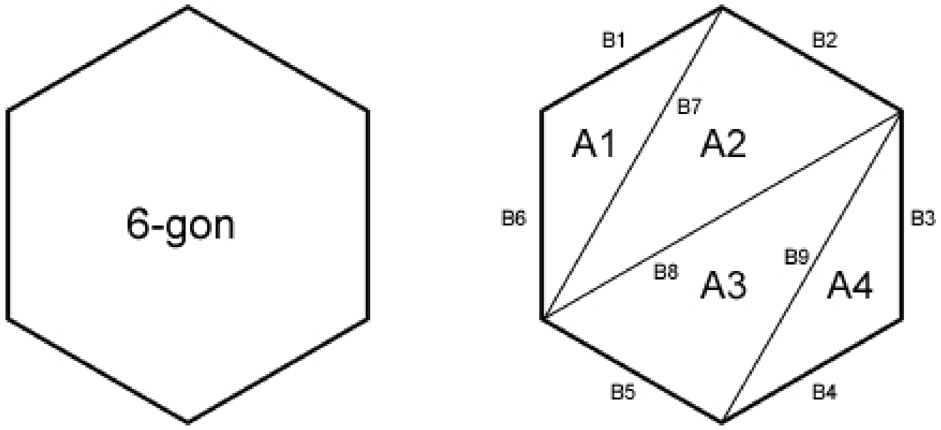
Polygon Triangulation

Time Limit : 1 Second
Memory Limit : 512 MB

In computational geometry, polygon triangulation is a problem of decomposing a polygon into a set of pairwise non-intersecting triangles whose total area is the same as the polygon's area. It is closely related with finding a maximal set of non-intersecting diagonals inside a polygon (this set of diagonals defined the triangles' edges). This problem is considered as important in many graphical applications.

In this problem, you are given an integer N which represents a regular polygon of N sides. Your task is to compute the number of triangles and edges (straight lines) in a polygon triangulation of the given polygon. Note that no matter how you triangulate a polygon of N sides, you will end up with the same number of triangles and edges. So, the output is only determined by N.

For example, consider a 6-gon (N = 6).



In this example, we will get 4 triangles (A1..A4 in the picture) and a total of 9 edges (B1..B9 in the picture).

Input

Input begins with an integer: T (1 ≤ T ≤ 100,000) denoting the number of cases.

Each case contains the following input block: Each case contains one integer: N (3 ≤ N ≤ 2,000,000,000) in a single line.

Output

For each case, output in a line "`Case #X: A triangle(s) with B edges.`" where `X` is the case number (starts from 1) and `A` is the number of triangles and `B` is the number of edges.

Examples

input **Example #1**

```
4
6
12
13
8
```

output

```
Case #1: 4 triangle(s) with 9 edges.
Case #2: 10 triangle(s) with 21 edges.
Case #3: 11 triangle(s) with 23 edges.
Case #4: 6 triangle(s) with 13 edges.
```

End of Problem