There are two parts in this assignment, and you have to submit separate files for each part. One part is a special case of the other but can be done more efficiently.

Let n be a positive integer and k the largest positive integer such that $\frac{k(k+1)}{2} \leq n$. It can be proved that any sequence of n distinct numbers can be partitioned into at most k subsequences, such that each subsequence is either strictly increasing or strictly decreasing. You have to write a program to find such a partitioning.

The second part is a generalization of the previous statement. Let G be any directed graph with n vertices. A subset of vertices is called a chain if there is a path in the graph that includes all vertices in the subset and no other vertices. A subset of vertices is an anti-chain if there is no edge in the graph directed from a vertex in the subset to another vertex in the subset. (This is also called an independent set). The vertex set of G can be partitioned into at most k parts, such that each part is either a chain or an anti-chain. You have to write a program to find such a partitioning. Note that k is the smallest possible value such that there exists such a partitioning for every graph with n vertices.

Submit two separate files Roll_No_8_1.cpp and Roll_No_8_2.cpp for the two parts.

Input Format
For the first part, the first line of input will specify the number n, $1 \leq n \leq 10^5$. The next line will contain n distinct numbers, separated by a single space. Each number will be less than n, so the sequence is just a permutation of $\{0, \ldots, n-1\}$. For the second part, the first line of input will specify the number n of vertices and the number m of edges in the graph. The next m lines will contain two integers each, specifying the edges in the graph. It can be assumed that there are no loops or multiple edges in the graph, and $1 \leq n \leq 10^4$, $1 \leq m \leq 10^7$. Use fast I/O for reading.

Output Format In both problems, the first line of output should specify the number p of parts obtained, which must be at most k. The next p lines should specify the parts. In each line, the first number must be +1 if the part is an increasing subsequence or a chain. The next number must be the number of elements in the part, followed by the actual elements. For subsequences, the elements in the part must be listed in increasing order, and for chains they must be listed in the order in which they appear in the path. If the part is a decreasing subsequence or an anti-chain, the first number must be -1, and the next, the number of elements in the part. Again, list the elements in the subsequence in decreasing order for such a part. For an anti-chain they can be listed in any order.

Sample Input	Sample Output
6 0 2 1 5 4 3	3 +1 3 0 2 5 +1 1 1 -1 2 4 3
6 6 0 1 2 1 2 3 4 3 4 5 0 5	2 -1 3 0 2 4 -1 3 1 3 5