1048 – Conquering Keokradong

This winter we are going on a trip to Bandorban. The main target is to climb up to the top of Keokradong. So, we will use a trail. The trail is a continuous marked footpath that goes from Bandorban to Keokradong.

Part of the experience is also the route planning of the trip. We have a list of all possible campsites that we can use along the way and we want to do this trip so that we only stop **K** nights to camp. We also know in advance the distance between consecutive campsites and we are only allowed to camp at a campsite. Our goal is to plan the trip so that we minimize the maximum amount of walking done in a single day. In other words, if our trip involves 2 nights (3 days of walking), and we walk 9, 10, 5 miles on each day respectively, the cost (maximum amount of walking done in one day) is 10. Another schedule that involves walking 9, 6, 9 miles on each day has cost 9.

Given the distances between **N** consecutive campsites of a trail and given the number of nights for your trip, **K**, your task is to devise a camping strategy for the specified trail such that it minimizes the maximum amount of walking done in a single day. Note that the first distance value given is the distance from our start-point of the trail to our 1^{st} campsite, and the last distance value given is the distance from our $\mathbf{N}^{th}$ campsite to our end-point of the trail.

Input

Input starts with an integer **T (≤ 200)**, denoting the number of test cases.

Each case contains of two integers, the number of campsites, **N (1 ≤ N ≤ 1000)** and the number of nights of the trip, **K (1 ≤ K ≤ min(N, 300))**. The following **N + 1** lines indicate the distance in miles between consecutive campsite locations. All the integers will be positive and less than **10000**.

Output

For each case of input you have to print the case number and the minimized cost as described above. Then print **K+1** lines, each containing the amount of distance covered in $\mathbf{i}^{th}$ day. As there can be many solutions, the primary target is to find the one which ensures that each day we have to walk some distance. For ties, print the one where the distance covered in first day is maximum, then the distance covered in second day is maximum and so on.

Sample Input	Output for Sample Input
1 4 3 7 2 6 4 5	Case 1: 8 7 8 4 5