1. A *mode* is a value in an array that is larger than both the value immediately before it in the array and the value immediately after it. In other words, a mode occurs at index `k` in the array `A` if `A[k] > A[k - 1]` and `A[k] > A[k + 1]`. The array is *unimodal* if the values increase until they reach a mode, then decrease, so that there is only one mode. For example, the array `A` shown below is unimodal with its mode occurring at index `4`. Assume that the mode does not occur at the first or last entry in the array.

Index k	A[k]	
0	3	
1	5	
2	9	
3	10	
4	12	← mode
5	11	
6	9	
7	4	

(a) Write function `IsMode`, as started below. `IsMode` returns `true` if `data[k]` is larger than `data[k - 1]` and larger than `data[k + 1]`; otherwise, it returns `false`. In the example above, the call `IsMode(A, 4)` returns `true` and the call `IsMode(A, 5)` returns `false`.

Complete function `IsMode` below.

```
bool IsMode(const apvector<int> & data, int k)
// precondition:  0 < k < data.length() - 1
```

(b) Write function `ModeIndex`, as started below. `ModeIndex` returns the index of the mode of `data`. You may assume that `data` is unimodal and the mode occurs at an index `k`, where `0 < k < data.length() - 1`. In the example above, the call `ModeIndex(A)` returns `4`.

In writing `ModeIndex`, you may call function `IsMode` specified in part (a). Assume that `IsMode` works as specified, regardless of what you wrote in part (a).

Complete function `ModeIndex` below.

```
int ModeIndex(const apvector<int> & data)
// precondition:  data is unimodal and data.length() ≥ 3
```

(c) Write function `PrintHistogram`, as started below. `PrintHistogram` prints a character histogram of a unimodal array of nonnegative values, `data`, such that the longest bar of the histogram (the mode) has `longestBar` characters `barChar`, and all other bars have a number of `barChar` characters proportional to the corresponding value in the array `data` (rounding down).

For example, assume that `apvector data` contains the values shown below.

The call `PrintHistogram(data, 20, 'x')` will print the histogram shown in the Output column below.

Index `k`	`data[k]`	Length of bar	Output of call `PrintHistogram (data, 20, 'x')`
0	3	5	xxxxx
1	5	8	xxxxxxxx
2	9	15	xxxxxxxxxxxxxxx
3	10	16	xxxxxxxxxxxxxxxx
4	12	20	xxxxxxxxxxxxxxxxxxxx
5	11	18	xxxxxxxxxxxxxxxxxx
6	9	15	xxxxxxxxxxxxxxx
7	4	6	xxxxxx

In writing `PrintHistogram`, you may call functions `IsMode` and `ModeIndex` specified in parts (a) and (b). Assume that `IsMode` and `ModeIndex` work as specified, regardless of what you wrote in parts (a) and (b).

Complete function `PrintHistogram` below.

```
void PrintHistogram(const apvector<int> & data,
                    int longestBar, char barChar)
// precondition:  data is unimodal and data.length() ≥ 3;
//                data[k] ≥ 0 for 0 ≤ k < data.length()
```