[35 points] In order to maintain the max-heap property, write a function
void maxHeapify(int *A, int i, int arrLen).
When it is called, maxHeapify assumes that left and right binary sub-trees are maxheaps, but that A[i] might be smaller than its children, thus violating the max-heap property. maxHeapify lets the value at A[i] float down" in the max-heap so that the subtree rooted at index i obeys the max-heap property.
Write a routine void heapSort(int *A, int arrLen) to sort the array A by using maxHeapify function only.

[50 points] Write a program to merge k sorted lists (use heapSort) into one sorted list, where n is the total number of elements in all the input lists (O(nlogk)). Your program should work for any number of list (i.e., k) and elements (n). Note that, each list may not contain equal number of elements. You need to take the input file as a command line argument. Storage for each individual arrays need to allocated dynamically.
[15 points] for good coding practises.
File Format:
k n1 n2 n3 . . . nk *//k* is the number of lists, n1+n2+. . .+nk = n
space separated n integers

[35 points] In order to maintain the max-heap property, write a function
void maxHeapify(int *A, int i, int arrLen).
When it is called, maxHeapify assumes that left and right binary sub-trees are maxheaps, but that A[i] might be smaller than its children, thus violating the max-heap property. maxHeapify lets the value at A[i] float down" in the max-heap so that the subtree rooted at index i obeys the max-heap property.
Write a routine void heapSort(int *A, int arrLen) to sort the array A by using maxHeapify function only.

[50 points] Write a program to merge k sorted lists (use heapSort) into one sorted list, where n is the total number of elements in all the input lists (O(nlogk)). Your program should work for any number of list (i.e., k) and elements (n). Note that, each list may not contain equal number of elements. You need to take the input file as a command line argument. Storage for each individual arrays need to allocated dynamically.
[15 points] for good coding practises.
File Format:
k n1 n2 n3 . . . nk *//k* is the number of lists, n1+n2+. . .+nk = n
space separated n integers

[35 points] In order to maintain the max-heap property, write a function
void maxHeapify(int *A, int i, int arrLen).
When it is called, maxHeapify assumes that left and right binary sub-trees are maxheaps, but that A[i] might be smaller than its children, thus violating the max-heap property. maxHeapify lets the value at A[i] float down" in the max-heap so that the subtree rooted at index i obeys the max-heap property.
Write a routine void heapSort(int *A, int arrLen) to sort the array A by using maxHeapify function only.

[50 points] Write a program to merge k sorted lists (use heapSort) into one sorted list, where n is the total number of elements in all the input lists (O(nlogk)). Your program should work for any number of list (i.e., k) and elements (n). Note that, each list may not contain equal number of elements. You need to take the input file as a command line argument. Storage for each individual arrays need to allocated dynamically.
[15 points] for good coding practises.
File Format:
k n1 n2 n3 . . . nk *//k* is the number of lists, n1+n2+. . .+nk = n
space separated n integers

[35 points] In order to maintain the max-heap property, write a function
void maxHeapify(int *A, int i, int arrLen).
When it is called, maxHeapify assumes that left and right binary sub-trees are maxheaps, but that A[i] might be smaller than its children, thus violating the max-heap property. maxHeapify lets the value at A[i] float down" in the max-heap so that the subtree rooted at index i obeys the max-heap property.
Write a routine void heapSort(int *A, int arrLen) to sort the array A by using maxHeapify function only.

[50 points] Write a program to merge k sorted lists (use heapSort) into one sorted list, where n is the total number of elements in all the input lists (O(nlogk)). Your program should work for any number of list (i.e., k) and elements (n). Note that, each list may not contain equal number of elements. You need to take the input file as a command line argument. Storage for each individual arrays need to allocated dynamically.
[15 points] for good coding practises.
File Format:
k n1 n2 n3 . . . nk *//k* is the number of lists, n1+n2+. . .+nk = n
space separated n integers