

National Institute of Technology, Calicut
Department of Computer Science and Engineering
CS2094 – Data Structures Lab
Assignment 2 (Advanced)

Submission deadline (on or before):

26th January 2016, 10:00:00 PM (for Advanced batch)

Naming Conventions for submission:

Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar or .tar.gz). The name of this file must be ASSG<Number>A_<ROLLNO>_<FIRSTNAME>.zip

(For example: ASSG2A_BxyyyyyCS_LAXMAN.zip). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive.

The source codes must be named as ASSG<Number>A_<ROLLNO>_<FIRSTNAME>_<PROGRAM-NUMBER>.<extension> (For example: ASSG2A_BxyyyyyCS_LAXMAN_1.c)

Questions:

1. Write a program that reads a sorted array A of integers and an integer k, and checks in $O(n)$ steps whether there are two elements in the array whose sum is k or not.

Input format:

The first line of input contains an integer n, the length of the array

The next line contains n space separated integers.

The next line contains a single integer k.

Output format:

A pair of numbers whose sum equals k, if such numbers exist. If there are more than one such pairs, then print any one pair. Otherwise, print "No two elements sum up to k".

Example 1

Input:

8
32 38 48 54 58 67 86 95
105

Output:

38 67

Example 2

Input:

8
32 38 48 54 58 67 86 95
100

Output:

No two elements sum up to 100

2. Given an array A of integers, a pair (i, j) of indices $0 \leq i, j \leq \text{length}(A)$ is called a *reversed pair* in A, if $i < j$ and $A[i] > A[j]$. Write a program that reads an array of integers, and prints the number of reversed pairs in the array. Your program should run in $O(n \log n)$ time.

Input format:

The first line of input contains an integer n, the length of the array.

The next line contains n space separated integers.

Output format:

The number of reversed pairs in the array.

Example 1

Input:

5

1 2 3 4 5

Output:

0

Example 2

Input:

5

5 4 3 2 1

Output:

10

Example 3

Input:

5

4 2 3 5 1

Output:

6

3. Consider how you would sort a file that was too large to fit in your machine's RAM. External Sorting refers to the sorting of data which is not in the primary memory of the computer. In this problem, you have to implement a miniature version of external sort. The file sizes have been intentionally kept small because of time constraints. The methodology followed has to be the same as for really big files.

- a) First generate a 256 MB file Bigfile containing integers. You can make use of a random number generator, with suitable modifications on the generator output, to generate suitable integers.
- b) Then sort the above file in the following manner
 - Bring in 1024*1024 integers into memory at a time, and sort into an output file, thus generating 64 sorted files of integers.
 - Do an 8 way merge on the contents of these files to generate 8 sorted files.
 - Do a 4-way merge on the above files to generate 2 sorted files.
 - Merge the two files obtained above into Bigfile, to obtain the sorted Bigfile.
