DATABASE SYSTEMS
Assignment 2

In this assignment you have to implement the **two -phase merge sort** algorithm to sort a large number of records.

Implementation Specs:

1. The metadata file will contain the information about the size of the different columns (in bytes).
2. The datatype for the column will be string only.
3. The number of columns can range from 1 to 20.
4. Your program should be capable of sorting in both ascending and descending order.
5. Your program should run for different values of main memory usage allowed and different size of files(MBs-GBs).

If we found your program is using main memory more than it is specified, you will lose marks.

Input Format:

Metadata.txt
Containing information about the columns.
<ColumnName 1>,<Size of the column>
<ColumnName 2>,<Size of the column>
......
<ColumnName n>,<Size of the column>

Input.txt
Containing the records with the column values separated by the space. All the values will be strings only.

Data:
To generate input file, visit http://www.ordinal.com/gensort.html and download **gensort** code to generate data.
The following command generates the first 100 tuples of three columns.

```
$> ./gensort a 100 input.txt
```

For generating input files with different sizes, you'll have to compute the size of each tuple and calculate for what number of tuples a certain file size can be achieved. For instance,

If **tuple size** = 100 bytes (sum of size of all columns),
file size = (100*total_tuples) bytes.

To validate your algorithm you can use the **valsort** available on the same link.

Command Line Inputs

1. Input file name (containing the raw records)
2. Output file name (containing sorted records)
3. Main memory size (in MB)
4. Order code (asc/desc) asc means to sort in ascending order and **desc** means to sort in descending order.
5. ColumnName 1
6. ColumnName 2
7. …...

Example

- ./sort input.txt output.txt 50 asc c0 c1
- ./sort input.txt output.txt 100 desc c0

As in the first example, the file input.txt to be sorted in ascending order with 50 MB space based on column c0 and if any column have same value for c0 then on the basis of c1.
(Similar to ORDER BY clause in SQL).
In case if the value is same for all the available columns, then the order should be the same as in the input file.

Record Observations:

1) Varying FileSize with constant Memory:
Take the main memory size as 100 MB and run you algorithm for file size 5MB, 50MB, 500MB, 1GB, 2GB, 3GB and note the **time taken** for each run.

2) Varying Memory Size with constant FileSize:
Take a file of size 512MB and run your code with main memory argument as 25 MB, 100 MB, 250 MB, 512 MB and note the **time taken** for each run..

Record your inferences and what in your opinion is the best explanation of the same.
A graphical interpretation will also suffice.

Submission:

Create a folder with the name rollno_Assign2 and put the following into it:
a) Your source code in the folder named as **Code** .
b) A pdf file with the name **Analysis.pdf** containing following information:
i) Configuration of the System
ii) Observations (in tabular format)
iii) Explanation
Compress the folder and upload the rollno_Assign2.zip

Error Handling:
Check feasibility condition for two -way merge sort for provided memory constraints. If not satisfied, provide error message and halt.

Important Instructions:
1. You are not allowed to use any external library/jar files in this assignment.
2. Plagiarism will not be tolerated. Copy from any source and in any form(friends/seniors/internet) will fetch you straight 0 marks in all the assignments.
3. Languages allowed to code are C/C++/Java.

DEADLINE: 24th September 2018