COMP ENG 2SH4

DURATION OF EXAMINATION: 75 Minutes — Instructor: MOHAMED HASSAN
MCMASTER UNIVERSITY Midterm Exam — Nov 2021

SPECIAL INSTRUCTIONS:

- First things first: Update the README file with your information.
- **Closed-Book Exam:** No reference material of any kind is allowed. Only McMaster standard calculator (Casio FX-991 MS or MS Plus) is allowed.
- Exam is three questions, each question is presented in a separate page in this document and has two parts (a) and (b)
- Answer questions in their corresponding source files as instructed in each question.
- Make sure to commit/push at least after each part of the exam and/or every 10 minutes to avoid any problems.
- Students approved by SAS for extra-time, please follow the guidelines given to you in the SAS letter.

1. (20 Marks) **Question1.**
   This question with its parts should be answered in file: Q1.c.
   (a)[10 Marks] Write a function addArrays, that takes as input three arrays: A, B, and C along with a forth parameter N, representing their size (all arrays assumed to be of same size). The function is required to update A with the summation of the corresponding elements from B, and C. The prototype given in Q1.c is incomplete and you are required to complete it along with the function body/definition.

   (b) [10 Marks] Complete the main program in file Q1.c to test your addArrays function.

2. (40 Marks) **Question2.**
This question with its parts should be answered in file: Q2.c.

Usually Graph data structures are defined using compact forms like Compressed sparse row (CSR), where all graph operations are transformed into vector operations and hence can be implemented using arrays. Assume you want to add the elements of two graphs (C and E), where the results are stored in new graph A. However, since graph nodes are not typically stored in contiguous memory locations, we have to define the order you access both graphs, which is given to you as follows: the order (index) to access nodes of graphs A, C, and E (which are represented by Arrays A, C, E respectively) are defined in Arrays B, D, and F, respectively. Figure 1 below gives an illustration of this.

(a) [30 Marks] You are required to write a function addCompactGraphs that takes as input all the six arrays A to F, along with integer N that represents the size of the arrays (all are assumed same size) and **update A as explained earlier to be the summation of C and E: Every element in A indexed by B[i] is the addition of the two elements in C and E that are indexed by D[i] and F[i], respectively**.

Example: For example, in the figure below, A[0]=6 (indexed by B[1]) because it is the addition of C[1] (indexed by D[1]) and E[3] (indexed by F[1]).

(b) [10 Marks] Complete the main program in file Q2.c to test your addCompactGraphs function.

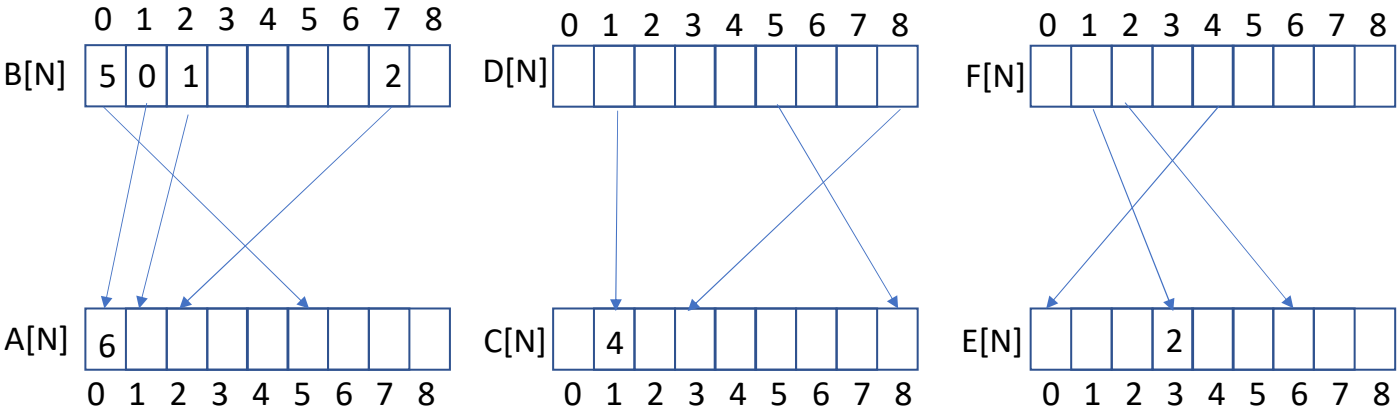

Figure 1: Question 2 example illustration

3. (40 Marks) **Question 3.**
This question with its parts should be answered in file: Q3.c.

(a) [25 Marks] Write a function parseDate that takes as input a Date in the format of a single integer of 6 digits, where it is encoded as: DDMMYY: where least significant two digits are for year, most significant two are for day, and middle are for month. You are asked then to extract these three segments separately. Please note that the caller function needs visibility to these result of extracted segments (In other words, your function results should be accessible by the caller). Build in the prototype given below, please note that the prototype is not complete and you have to modify it:

```
void parseDate(int input_date)
```

(b) [15 Marks] Now, we will make use of the parseDate function from part(a). Write a function dateDifference, which takes as input two dates in the same input format as A (DDMMYY), and then returns the difference between both dates in number of days. Please note that the input dates are not given in order. Please also note that the difference has to be always positive. You are give the function prototype below:

```
int dateDifference(int input_date1, int input_date2)
```

THE END