CSC 300: Assignment2 War2

Short version

Watch the video: <https://videos.dsu.edu/Watch/Ka68Aqc5>

Start with your War1 code as a starting point for each variation below. In the end, you will develop 3 more programs that follow the same model & guidelines. Yes, this is 3 programs. But each should take a small amount of time so the workload is very reasonable. This is practice & review. You have to do work to review.

Upon completion of War1, a video going through my solution will be available to you. Consult if you are stuck on any sections or just want to see another version of the code. But if you aren’t interested there is nothing else essential in that video.

Due: Sunday 5/31 6pm

Program war2a: multi-dimension arrays

Program war2b: structures

Program war3a: dynamic allocation of parallel arrays

See below for more details on the requirements for each

Overall:

/\*

Start with your code from the A1War1 assignment

Each of the 3 programs below can either evolve from the

initial code or from each other. That is up to you.

\*/

Program war2a: multi-dimension arrays

/\*

Program war2a:

- 2d array

All the rules apply, except for these enhancements:

- 2d array stores the 2 columns of numbers

- adjust declarations & parameters to accept

- function changes: maxA, sumA, searchval

- add another parameter ( last )

1 or 2 to denote which column is to be used

- you do not need to validate the parameter value

- function changes: searchpair, adda, war

- 2d array replaces parameters for individual lists

but you knew that

\*/

Program war2b: structures

/\*

Program war2b:

- structures

All the rules apply, except for these enhancements:

- array of structures stores the 2 columns of numbers

- adjust declarations & parameters to accept

- function changes: maxA, sumA, searchval

- add another parameter ( last )

1 or 2 to denote which column is to be used

- you do not need to validate the parameter value

- function changes: searchpair, adda, war

- array of structures replaces parameters for individual lists

but you knew that

\*/

Program war3a: dynamic allocation of parallel arrays

/\*

Program war3a:

- dynamically allocated arrays

All the rules apply, except for these enhancements:

- array to store the 2 columns of numbers

should be dynamically allocated to the proper size

- very similar to A1War1 otherwise

Add this function & call it from main

int \*allocArray( int n )

\*/