Programing Fundamentals

Assignment # 04: BCS && BSE && BSAI

Assignment deadline Tuesday, 12 December 2023, 11:00 PM

Problem#01:

Inverse of a matrix A is a matrix A^{-1} such that $\mathbf{A \times A^{-1} = I}$. Where **I**, is an identity matrix. Declare two matrices such that matrix **A** of size **n*m** and another matrix **B** of size **m*n**. Take the input values of both the matrices A and B in **main** program and use proper output statements to get the input of the matrices with the certain dimensions.

You are required to create a user-define function "**checkInverse**" having **four** parameters.

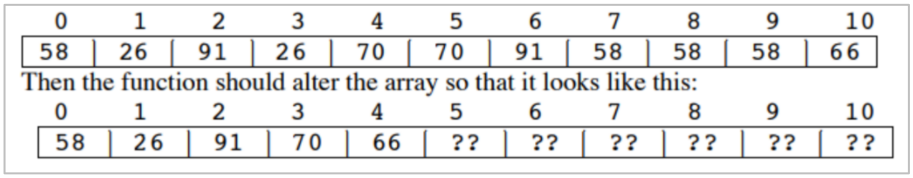
1. 2D matrix A of int type as first parameter,
2. 2D matrix B of int type as second parameter,
3. rowSize and
4. colSize

Function will tell if B is inverse of A or not.

Note: That will return true or false of bool type. Hint: Try matrix multiplication.

Problem#02:

Write a function named "**eliminate_duplicates**" that takes an array of integers in random order and eliminates all the duplicate integers in the array. The function should take two arguments: **(1)** an array of integers**; (2)** an integer that tells the number of cells in the array. The function should not return a value, but if any duplicate integers are eliminated, then the function should change the value of the argument that was passed to it so that the new value tells the number of distinct integers in the array. Here is an example. Suppose the array passed to the function is as shown below, and the integer passed as an argument to the function is **11**.



0	1	2	3	4	5	6	7	8	9	10
58	26	91	26	70	70	91	58	58	58	66

Then the function should alter the array so that it looks like this:

0	1	2	3	4	5	6	7	8	9	10
58	26	91	70	66	??	??	??	??	??	??

and it should change the value of the argument so that it is **5** instead of **11**. The question marks in the cells after the **5th** cell indicate that it does not matter what numbers are in those cells when the function returns.

Problem#03:

Write a user-defined function "**sortNames**" having one parameter (array of string variables). Suppose you've initialized array of Strings in main e.g {**"Ali", "Usman", "Abdullah"**} You have to sort them in ascending order by calling the above function **sortNames.** To decide which string is greater use the following algorithm. Multiply the ascii of each character with its index and add them. The string with higher value would be greater. For Example suppose the string passed are **Ali & Umar** then :

Ali : (ascii of A) * 1 + (Ascii of l) * 2 + (Ascii of i) * 3 = (65) * 1 + (108) * 2 + (105) * 3 = 491

Umar : (ascii of U) * 1 + (Ascii of m) * 2 + (Ascii of a) * 3 + (Ascii of r) * 4 = (85) * 1 + (109) * 2

+ (97) * 3 + (114) *4 = 1050.

Problem#04:

Write a function “**longestWord**” that takes a 2D array of characters as input and displays the longest word in each row and each column and print it out with its length. (void function)

Problem#05:

Write a function named "**g_c_d**" that takes two positive integer arguments and returns as its value the greatest common divisor of those two integers. If the function is passed an argument that is not positive (i.e., greater than zero), then the function should return the value 0 as a sentinel value to indicate that an error occurred. Thus, for example,

```
cout << g_c_d(40,50) << endl;        // will print 10
cout << g_c_d(42,6) << endl;         // will print 6
```

Problem#06: (Pass-by-Reference)

Implement the body of following user-defined functions prototypes and call them in main program.

Void convertDouble*(double &value);*	***Void convertInt****(int &value);*
If the value is positive, compute its factorial. If value is negative, calculate its absolute value. If the value is zero, set it to 42.	If the value is an even, raise it to the power of 3. If the value is odd, replace it with sum of all of its digits. If the value is multiple of 3 then, set it to 100.

Problem#07: File-Handling

Write a C++ program that reads 20 floating point value from an **input.txt file** (unsorted), read them into a floating-point array in main program, pass this array to function that does pass-by-reference and perform Bubble Sort over the array. The resulting sorted array would then be placed in an **output.txt file** and displayed at console after the end of program.

Problem#08: File Handling

Encryption is very important to save data access from unauthorized user. Write a C++ code in which Get employee data in the main program for 5 different employees such as name, age, income and address.

File Writing: Save this data in file named **employee.txt**. Data in file should be stored in encrypted form.
File Reading: Then read all the data from above file and print on console in decrypted form.

User Input Data	**File Writing: Data stored in file**	**File Reading: Output Console**
Name=ali	Name=Bmj	Name=ali
Age=23	Age=34	Age=23
Address=Pakistan	Adress=Qbljtubo	Address=Pakistan
Income=200000	Income=311111	Income=200000

Hint: Use increment form i.e. **str[i]+1**for storing in file and decrement form **str[i]-1**for output.