

National University of Computer and Emerging Sciences, Lahore Campus



Course:	ITC Lab	Course Code:	CS-101
Program:	BS (Computer Science)	Semester:	Fall2017
Duration:	150 Minutes	Total Marks:	40
Paper Date:	04-Dec-2017	Weight	40%
Section:	C	Page(s):	2
Exam:	Final Term	Roll. No.	

Instructions:

1. Understanding the question paper is also part of the exam, so do not ask any clarification.
2. Make sure to switch off your mobile phones before the Exam starts.
3. No USB's are allowed. No INTERNET is allowed. Please see that the area in your threshold is clean. You will be charged for any material which can be classified as 'helping in the paper' found near you.
4. Talking/Discussion is not allowed. It is your responsibility to protect your code and save it from being copied. If you don't protect it all matching codes are considered copy/cheating cases.

Q-1: Here is a simplified version of a word search puzzle. You are given a text file called dictionary.txt with a list of words of length 2, 3 and 4. (4 is the maximum length, and no word can have a length greater than 4). Use this file to solve the problem given below:

Take input text from the user in **data**. Your job is to determine which words given in dictionary.txt file also occur in **data**. Two conditions apply:

1. The characters in the word may **not** occur consecutively in **data**.
2. The order of the characters in each word is preserved.

Store all the words that occur in **data**, based on the conditions given above, in a text file named output.txt.

Sample Input:

Data: catinksaymildentb

Dictionary.txt

```
ink
cat
fun
icy
play
name
salt
farm
tin
bat
dent
```

output.txt

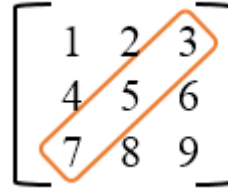
```
ink
cat
name
salt
tin
dent
```

Q-2: Write a C++ program to read elements in a matrix and find the sum of minor diagonal (opposite diagonal) elements. Minor diagonal of a matrix **A** is collection of elements **A_{ij}** Such that $i + j = (N + 1) - 2$. Where N is size of matrix (N x N).

Input

Input elements in array:

1 2 3
4 5 6
7 8 9



Output

Sum of minor diagonal elements = 15

Q-3: Write a function named "shift_right" that takes as its arguments the following:

- (1) an array of floating point values;
- (2) an integer, call it "left", that tells the leftmost cell of the part of the array to be shifted;
- (3) an integer, call it "right", that tells the rightmost cell of the part of the array to be shifted;
- (4) a positive integer, call it "distance" that tells how many cells to shift by.

NOTE: Write a main function to test your function.

The function should make sure that `left` is less than or equal to `right`, and that `distance` is greater than zero. If either of these conditions fails, the function should return the value 1 to indicate an error.

Otherwise it should shift by `distance` cells the contents of the array cells with subscripts running from `left` to `right`. Thus, for example, if the array passed to the function looks like this:

0	1	2	3	4	5	6	7	8	9	10
5.8	2.6	9.1	3.4	7.0	5.1	8.8	0.3	-4.1	8.0	2.7	etc.

and if `left` has the value 3, `right` has the value 7, and `distance` has the value 2, then the function should shift the contents of cells 3, 4, 5, 6, and 7 to the right by 2 cells, so that when the function

returns, the array will have been changed so that it looks like this:

0	1	2	3	4	5	6	7	8	9	10
5.8	2.6	9.1	???	???	3.4	7.0	5.1	8.8	0.3	2.7	etc.

The question marks in cells 3 and 4 indicate that we don't care what numbers are in those cells when the function returns. Note that the contents of cells 8 and 9 have changed, but the contents of cell 10 is unchanged. The function need not take any precautions against the possibility that the cells will be shifted beyond the end of the array (the calling function should be careful not to let that happen).

***** Best ☺ of Luck *****