

The due time of Lab2 is 11:00 PM on Mar.12, 2015.

The Lab2 submission instruction:

- (1) Please create a .c file for each of your programs.
 - (2) Please prepare a text (.txt) file, clearly describing how to run all your programs and what input to give. Your file should be named Readme.txt.
 - (3) Zip all the files and Email the zip file to the TA. Subject of Email: **Lab2_LName_UTAID!**
 - (4) Note that all programs should be tested on Omega using gcc before submission.
- Make sure you follow the good programming styles, for example, comments, indentation and white space.

Problem 1:

You are given two sorted arrays, A and B, and A has a large enough buffer at the end to hold B. Write a method to merge B into A in sorted order without using any other array space.

Implementation instruction:

- (1) Define one array of size 18 for A and the other array of size 5 for B.
- (2) Initialize A by inputting 13 integers in the ascending order, and Initialize B by inputting 5 integers in the ascending order. (Note: don't hard code the integers of the arrays.)
- (3) Merge B with A in a way all values are sorted.
- (4) Print out the updated array A, after merging with B.

For example:

If your input for A is

1, 3, 11, 15, 20, 25, 34, 54, 56, 59, 66, 69, 71

and your input for B is

2, 4, 5, 22, 40

Finally, after merging A and B, A becomes

1, 2, 3, 4, 5, 11, 15, 20, 22, 25, 34, 40, 54, 56, 59, 66, 69, 71

Problem 2: Look for pairs of integers

Given a sorted array of 20 integers and a target integer value, your program needs to print out all pairs of integers which have sum equal to the target value. If there is no pair of integers to return, print out "No pair of integers in the given array can be summed to the target value". The array of integers and the target value should be given from the keyboard (taken as user input).

Problem 3:

Read an integer from keyboard and then print the result of the sum obtained by adding the entered integer to the integer formed by reversing the order of the digits.

(For example: entered number = 123. Sum = 123+321 = 444)

Problem 4:

Write a C program that counts the number of repeated character in a phrase entered by the user and prints them.

Note: Assume the length of the string is 10. Space in the input string need not be counted for repetition.

Example Test Cases & Output:

Test Case-1:

Enter the string:

Roof

Output:

o-2

Test Case-2:

Enter the string: roof of th

Output:

o-3, f-2

Problem 5:

Write a C program to reverse a string. The string can be of any length and might be empty. User will enter the string.

For example: if we reverse “I am at UTA” then we will get “ATU ta ma I”

Problem 6:

Read 5 strings of each size 4 into a 2-dimensional character array. Output these strings in the sorted order. To compare the strings, strcmp() can be used.

Example Test Cases & Output:

Enter the 5 strings one by one delimited by new line.

abfg
abcd
xyzw
pqrs
orde

Sorted strings:

abcd
abfg
orde
pqrs
xyzw