PROGRAMMING & DATA STRUCTURES LAB (CS19003) SECTION 11

LAB CLASS #6: LAB TEST #I

Sudeb P Pal & Partha P Chakrabarti

Indian Institute of Technology Kharagpur

January 12, 2022

Instructions: PDS (Sec. 11) Lab Test (Jan 12 2022)

1. We will all start in the General Channel at 2 pm. Channel will be on by 1:50 pm. Please join by 1:55 pm for sure. We will give the instructions (for 10 minutes) and announce the tests.
2. After that each student will move to their Group Channels for the rest of the class and start the test.
3. You be given two test problems for which you will have to write C programs, compile, run, test with data provided and submit through Teams.
4. Each Problem will be of 25 marks, 8 marks for intermediate submission and 17 marks for final submission. Each submission will be evaluated for amount of work completed and correctness.
5. Test Problem 1 will be available by 2:15 pm. Intermediate submission will be till 2:50 pm. Final submission will be till 3:30 pm. File name format is T01_<Roll Number>.c
6. Test Problem 2 will be available by 3:40 pm. Intermediate submission will be by 4:15 pm. Final submission will be by 4:55 pm. File name format is T02_<Roll Number>.c
7. Your TA will not help you in the Test. They will take attendance and keep watch. If you have a problem send a message to the Teacher. Make sure you are connected to the computer, network and have power backup readily available. You have to be present throughout the period.
8. Late submissions or submissions by email will not be accepted or if accepted will be severely penalized.
9. For any sort of plagiarism, you will get 0 for the whole test and also subsequent punishment.

Protocol for File Headers

- Every program must start with a comment containing
  - Section No.
  - Roll No.
  - Name
  - Group No.
  - Test Problem No.
  - A one line description of the assignment
- Type the example header (replace with your name, roll no. assignment no. etc.) at the beginning of each of your C file, even before the #include <stdio.h>

```
/*
*    Section 11
*    Roll No  :  21CS30010
*    Name    :  Your Name
*    Group No.: 7
*    Test Problem No : 01
*    Description : Program to check points
*/
#include<stdio.h>
#include<math.h>
```

Thank you

PDS Section 11, Lab Test1 Problem 01 (Jan 12, 2022)

Write a C Program to read in a positive integer n and print it. Then find the smallest integer m > n such that m has at least 4 factors (other than 1 and itself) and all its digits occur only once in m (that is, none of its digits are duplicated in m). Print m, all the factors of m and all the digits between 0 to 9 that m does not have. If it has all the digits, then print a line saying it has all digits.

For example, if n = 20, the answer is m = 24, since it has factors 2, 3, 4, 6 and 12 and the digits 2 and 4 in 24 occur only once each. The digits that m does not have are 0, 1, 3, 5, 6, 7, 8, 9.

Save the file as T01_<Roll Number>.c (example T01_21AG10002.c). Build, Run and test it for the given data as well as your own data. Then upload the .c file for the Test Problem.

[25 Marks: 8 marks for intermediate submission, 17 marks for final submission]

In particular, you will do the following:

1. Read in a positive integer n. Print n. Create a new line.
2. Print (with a suitable message) the smallest integer m > n such that m has at least 4 factors other than 1 and m, and all its digits occur only once in m. Create a new line.
3. Print all the factors of m other than 1 and itself appropriately. Create a new line.
4. Print all the digits that m does not have. If it has all the digits then print, “All digits present” in a new line.

Arrays allowed. No additional library functions except scanf and printf. No user defined functions allowed.

Test Data:

a) n = 20

b) n = 99

c) n = 100000

PDS Section 11, Lab Test1 Problem 02 (Jan 12, 2022)

Write a C Program to read in two strings A and B and print them. Next find out the frequency of the characters occurring in A and B, respectively. Then find out if every character present in B is present strictly more number of times in A than in B. Treat upper case and lower case as same / non-distinct. You may assume that neither A nor B have any blank characters within them and so can be read in %s format.

For example, if A = IITKharagpur and B = air, then the answer is Yes. On the other hand, if A = IITKharagpur and B = IITKgp, then the answer is No.

Save the file as T02_<Roll Number>.c (example T02_21AG10002.c). Build, Run and test it for the given data as well as your own data. Then upload the .c file for the Test Problem.

[25 Marks: 8 marks for intermediate submission, 17 marks for final submission]

In particular, you will do the following:

1. Read two character strings A and B. Print them appropriately, one in each line. Create a new line.
2. Print each distinct character in A along with its frequency, appropriately. Create a new line.
3. Print each distinct character in B along with its frequency, appropriately. Create a new line.
4. If every character present in B is present in A strictly more number of times than in B, print “Yes”, with a suitable message. Otherwise print “No” with a suitable message. Create a new line.

Arrays allowed. No additional string library function other than scanf, printf. No user defined functions.

Test Data:

a) A = IITKharagpur, B = air

b) A = IITKharagpur, B = IITkgp

c) A = 3$%2$alphalpha$, B = $alpha$

Programming & Data Structures Lab (CS19003) Section 11

LAB CLASS #09: LAB TEST #2

Sudeb P Pal & Partha P Chakrabarti

Indian Institute of Technology Kharagpur

February 16, 2022

Instructions: PDS (Sec 11) Lab Test 2 (Feb 16 2022)

1. We will all start in the General Channel at 2 pm. Channel will be on by 1:50 pm. Please join by 1:55 pm for sure. We will give the instructions (for 10 minutes) and announce the tests.
2. After that each student will move to their Group Channels for the rest of the class and start the test.
3. You be given two test problems for which you will have to write C programs, compile, run, test with data provided and submit through Teams.
4. Each Problem will be of 25 marks, 8 marks for intermediate submission and 17 marks for final submission. Each submission will be evaluated for amount of work completed and correctness.
5. Test Problem 3 will be available by 2:15 pm. Intermediate submission will be till 2:50 pm. Final submission will be till 3:30 pm. File name format is T03_<Roll Number>.c
6. Test Problem 4 will be available by 3:40 pm. Intermediate submission will be by 4:15 pm. Final submission will be by 4:55 pm. File name format is T04_<Roll Number>.c
7. Your TA will not help you in the Test. They will take attendance and keep watch. If you have a problem send a message to the Teacher. Make sure you are connected to the computer, network and have power backup readily available. You have to be present throughout the period.
8. Late submissions or submissions by email will not be accepted or if accepted will be severely penalized.
9. For any sort of plagiarism, you will get 0 for the whole test and also subsequent punishment.

Protocol for File Headers

- Every program must start with a comment containing
  - Section No.
  - Roll No.
  - Name
  - Group No.
  - Test Problem No.
  - A one line description of the assignment
- Type the example header (replace with your name, roll no. assignment no. etc.) at the beginning of each of your C file, even before the #include <stdio.h>

```
/*
*    Section 11
*    Roll No  :  21CS30010
*    Name     :  Your Name
*    Group No.: 7
*    Test Problem No : 03
*    Description : Program to check points
*/
#include<stdio.h>
#include<math.h>
```

Thank you

Test Problem 03

You are to develop a menu driven program to store marks for n students (n<=20) in a 2-D array and perform certain operations based on user options. Each row of the array will store the roll number and marks for Physics and Chemistry subjects for a student (all integers within 100). Then you will (in a suitable continuous loop) choose an option among 0, 1, 2, 3 and 4. In option 1, you will add a new student with roll number and marks for the two subjects. For option 2, you will change the marks of the two subjects of an existing student. For option 3, you will print the merit list and rank based on the total marks obtained (descending order) with ties resolved in favour of marks in Physics. For option 4, you will list the Gold Medal winners, on total as well as subject-wise. For option 0, you will exit from the program. Options can be in any order and can repeat. You will write your own appropriate functions to implement each of options 1, 2, 3 and 4.

Save the file as T03_<Roll Number>.c (example T03_21AG10002.c). Build, Run and Test it for your own data. Then upload the .c file for the Assignment.

[25 Marks: 8 marks for intermediate submission, 17 marks for final submission]

In particular, you will do the following:

1. Write a function to add a new student with roll number and subject marks. The function will take as argument the array name, current number of students and new roll number. The marks will be read within the function. If a duplicate roll number is given, you will give a warning and ask for input again. If number exceeds 20, give a message, go to main menu.
2. Write a function to change marks for an existing student. The function will take as argument the array name, current number of students and a roll number. The new marks will be read within the function. If the roll number does not exist, then you will give a warning and take input again. If there are no students, then give a message and go to main menu.
3. Write a function to calculate and print the merit list with rank, roll number, individual and total marks, row-wise. This function will take as argument the array name and current number of students. Ties in total marks will be resolved by marks in Physics. If there is a tie in all marks then they will get the same rank. For example, if three people get rank 4 then the next rank will start from 7. If there are no students, you will give a suitable message and go to main menu.
4. Write a function to list the winners of three Gold Medals. The three Gold Medals are for highest in total (with preference to Physics as in item 3 above), as well as highest in Physics and Chemistry. If there are ties, all those who are respective toppers will be selected for the specific Gold Medals. If there are no students, give a suitable message, go to main menu.
5. Write a main program that takes in the options of the user and calls the appropriate functions.

[Do not call any Library Functions other than scanf, printf. You may use your own user defined functions]

Test Data: Test it with your own data for different cases

Test Problem 04

You are to read in a character string into an array A. After that you will write a function that takes as one of the arguments, an integer k and forms a new string B with the first k characters of A reversed followed by the rest of the characters of A. For example, if A = abcadg and k = 4, then B will be acbadg. You will then find out the number of character mismatches between A and B. Upper and lower cases of the same character will be treated as the same for matching purposes. In the above example, the number of mismatches is 2. This will be done for all values of k from 1 to the length of the string A. The mismatch result will be printed for all k values along with the relevant partly reversed string.

In order to implement this you will develop both non-recursive and recursive functions for (a) conversion of a string A to B by reversing the first k characters of A and leaving the rest intact (b) Matching two strings to find the number of mismatches. You will call these functions appropriately in the main program to find the solutions as desired above, for the same input, using both non-recursive and recursive function calls.

Save the file as T04_<Roll Number>.c (example T04_21AG10002.c). Build, Run and Test it for the given and your own data. Then upload the .c file for the Assignment.

[25 Marks: 8 marks for intermediate submission, 17 marks for final submission]

In particular, you will do the following:

1. Read in a character string A and print it. Create a new line.
2. Write a non-recursive function krev with appropriate arguments to reverse the first k characters of A to form B.
3. Write a non-recursive function mism with appropriate arguments to find the number of mismatches between A and B.
4. You will call the above functions in the main program for all values of k from 1 to length of string A and for every k neatly print the string B along with the number of mismatches between A and B for that k, in a new line for each k.
5. You will write recursive versions of 2 and 3 above calling these functions rkrev and rmism, respectively.
6. You will repeat step 4 using the recursive versions for the same input A and print the result. You will not read A again.

[Do not call any Library Functions other than scanf, printf. You may use your own user defined functions, including to find the length of a character string.]

Test Data: Test it with the following: A = abcadg, A = $baB$aaaBaaa, A = Malayalam as well as your own data covering various cases.

PROGRAMMING & DATA STRUCTURES LAB (CS19003) SECTION 11

LAB CLASS #12: LAB TEST #3

Sudeb P Pal & Partha P Chakrabarti

Indian Institute of Technology Kharagpur

March 9, 2022

Instructions: PDS (Sec 11) Lab Test 3 (Mar 9 2022)

1. We will all start in the General Channel at 2 pm. Channel will be on by 1:50 pm. Please join by 1:55 pm for sure. We will give the instructions (for 10 minutes) and announce the tests.
2. After that each student will move to their Group Channels for the rest of the class and start the test.
3. You be given two test problems for which you will have to write C programs, compile, run, test with data provided and submit through Teams.
4. Each Problem will be of 25 marks, 8 marks for intermediate submission and 17 marks for final submission. Each submission will be evaluated for amount of work completed and correctness.
5. Test Problem 5 will be available by 2:15 pm. Intermediate submission will be till 2:50 pm. Final submission will be till 3:30 pm. File name format is T05_<Roll Number>.c
6. Test Problem 6 will be available by 3:40 pm. Intermediate submission will be by 4:15 pm. Final submission will be by 4:55 pm. File name format is T06_<Roll Number>.c
7. Your TA will not help you in the Test. They will take attendance and keep watch. If you have a problem send a message to the Teacher. Make sure you are connected to the computer, network and have power backup readily available. You have to be present throughout the period. We will ask you to have your videos on.
8. Late submissions or submissions by email will not be accepted or if accepted will be severely penalized.
9. For any sort of plagiarism, you will get 0 for the whole test and also subsequent punishment. Do not share even after the examination is over and you have submitted.

Protocol for File Headers

- Every program must start with a comment containing
  - Section No.
  - Roll No.
  - Name
  - Group No.
  - Test Problem No.
  - A one line description of the assignment
- Type the example header (replace with your name, roll no. assignment no. etc.) at the beginning of each of your C file, even before the #include <stdio.h>

```
/*
*     Section 11
*     Roll No  :  21CS30010
*     Name     :  Your Name
*     Group No.: 7
*     Test Problem No : 05
*     Description : Program to …….
*/
#include<stdio.h>
#include<math.h>
```

Test Problem 05

Write a program that will read a positive integer n, dynamically allocate two n by n matrices where each matrix element is a lower-case alphabetic string of (upto 4) unique characters ordered alphabetically. The sum of two strings s1 and s2 is defined as the first (upto 4) alphabetically ordered unique characters from the set of all characters (union) in s1 and s2. [For example: agrt + gptu = agpr] The product of two strings s1 and s2 is defined as the first (upto 4) alphabetically ordered unique characters of the common characters (intersection) of s1 and s2 (and could also be a null string). [For example: agrt . gptu = gt] Write functions for Matrix Addition and Matrix Multiplication of two such string matrices based on the above operations, using the sum and product operations on strings as base operators and the standard Matrix addition and multiplication rule. In the main program, read n, dynamically allocate two string matrices A and B, read the matrices A and B and print them row-wise. Then call the functions to Add and Multiply A and B using Matrix Addition and Matrix Multiplication as defined above and print the results again row-wise. If a Matrix cell is a null string, print the string “NULL”.

Save the file as T05_<Roll Number>.c (example T05_21AG10002.c). Build, Run and Test it for your own data. Then upload the .c file for the Assignment.

[25 Marks: 8 marks for intermediate submission, 17 marks for final submission]

In particular, you will do the following:

1. In the main program, read a positive integer n and dynamically allocate two n by n matrices A and B of strings, where each matrix element can be a string of at most 4 unique characters, ordered alphabetically. Read the contents of A and B. Assume that initial elements strings are non-null. Print A and B in row major fashion.
2. Write functions to find the sum and product of two strings and return the result string as per definitions mentioned earlier.
3. Write functions for Matrix Addition and Matrix Multiplication using the string sum and product operations and return the result Matrices.
4. Call these functions appropriately in the main program to obtain the results matrices for Addition and Multiplication of A and B as C and D. Print the result matrices C and D in row major order, appropriately.

[Do not call any Library Functions other than scanf, printf, basic math functions. Use your own user defined functions]

Test Data: Test it with your own data for different cases

Test Problem 06

You are to manipulate a set of polynomials of a single variable where the coefficients and exponents are rational numbers of the form (x, y) depicting the rational number x/y, where x and y are positive integers. Two rational exponents (x1,y1) and (x2,y2) are considered equal if the value of x1/y1 is equal to the value of x2/y2. Write a program that reads two such polynomials P and Q. For each polynomial, read the number of terms and dynamically allocate the polynomial. Read each polynomial term by term. For each term, read pairs of integers each for their coefficients and exponents. The terms in the input may not come in ascending order of x/y values of exponents. Write a function to print a polynomial by printing the (x, y) values of the coefficient and exponent of each term (and not as floating point numbers) in ascending order of the value (x/y) of the exponent. Print the two polynomials P and Q read using this function. Next, write functions for addition and multiplication of two such polynomials to return the result polynomials, respectively. Call these functions in the main program to find R = P + Q and S = P X Q and print the polynomials R and S returned as a result using the ordered print function mentioned earlier.

Save the file as T06_<Roll Number>.c (example T06_21AG10002.c). Build, Run and Test it for the given and your own data. Then upload the .c file for the Assignment.

[25 Marks: 8 marks for intermediate submission, 17 marks for final submission]

In particular, you will do the following:

1. In the main program, read positive integers n and m. Read polynomials P of n terms and Q of m terms. Use dynamic allocation.
2. Write a function to print a polynomial in ascending order of x/y value of the exponent. For every term print the coefficient as a pair followed by the character variable Z followed by the exponent as a pair. Use the + character in between terms.
3. Write functions to Add and Multiply two polynomials and return the result polynomial.
4. In the main program, call the functions Add and Multiply to get R = P + Q and S = P X Q. Use the print function (stated in item 2 above) to print the result polynomials R and S.

[Do not call any Library Functions other than scanf, printf, basic math functions. Use your own user defined functions]

Test Data: Test it with your own data covering various cases.