

Indian Institute of Technology Kharagpur

AUTUMN Semester, 2015

COMPUTER SCIENCE AND ENGINEERING

CS19001: Programming and Data Structure Laboratory

Assignment – 10

Full Marks: 10

Time allowed: 3 hours

INSTRUCTIONS: Please see the questions and write C programs step by step. Ensure proper indentations to improve the readability of your code. All these features are necessary and absence will lead to deduction of marks.

Please do not forget to upload files to *Moodle* before you leave.

Pointers to Structures in C

1. Write a complete C program with the following: define a structure `student` which contains the name (maximum 10 characters), age and the roll number of an IIT-KGP student (hence roll number is a string). Declare two variable of the above structure type, and then populate the fields of the two variables by collecting values from the user. Then, create a new variable, again of type `student`, but with age being the sum of the two ages, the name being the lexicographically higher one, and the roll number being the lexicographically lower one, using a function `student merge_students (student *ptr1, student *ptr2)`. Display the fields of the new variable from inside the `main()` function. (3 marks)
 2. Write a complete C program with the following: define a structure `student` which contains the name (maximum 10 characters) and the marks obtained in PDS for a student. Then, declare an array capable of holding information for 3 such students, and populate the fields in the array entries by collecting the proper data from the user. Finally, calculate the sum total of the marks obtained by the students by calling a function `void sum_marks (int *sum_ptr, student *student_ptr)`, using pointers to individual structure elements in the array. Display the average of the PDS from inside the `main()` function. (3 marks)
 3. In an earlier assignment you had implemented polynomial multiplication using structures; in this problem you would reimplement the same using pointers to structures. Recall that a polynomial can be represented as an array of structure in the following manner: each element of the array represents one term of the polynomial, and is a structure that contains the coefficient (a float) and the exponent (an integer). Let such a structure be named `PolyTerm`. For convenience, you can store the terms of the polynomial in decreasing order of exponent in the array. Given two such polynomials (entered by the user), write a complete C program with a function `void mult_poly (PolyTerm *poly1_ptr, int max_exp_poly_1, PolyTerm *poly2_ptr, int max_exp_poly_2, PolyTerm *polyresult_ptr)` to calculate and store the product polynomial (for the polynomials `poly_1` and `poly_2`) in the array `poly_result`. Your program should print the coefficients of the terms of the product polynomial, in decreasing order of exponent, e.g. if the product polynomial is $x^3 + 3x^2 + 3$, the program should print: 1 3 0 3. Use dynamic memory allocation to ensure just sufficient memory is allocated to hold the result polynomial, and make sure to free all dynamically allocated memory before your program exits. (4 marks)
-