Indian Institute of Engineering Science and Technology, Shibpur
B.Tech. 2nd Semester (ARCH/CST/EE/ETC/IT) Midterm Examinations, 2019
Introduction to Computing (CS - 1201)

Full Marks: 50 **Time: 2 Hours**

[For the programming problems use C language. Two marks are reserved for neatness.]

SECTION - A
Answer any six questions.

$[(3 \times 6) = 18]$

1. (a) Explain the function of Random Access Memory (RAM) in a digital computer.

   (b) Write the decimal value of the two 4-bit numbers, 1111 and 0000, assumed to be in i) sign-magnitude, ii) 1's complement and, iii) 2's complement form.

   (c) Convert the following numbers from the given base to the desired base.

   i. $(101.1)_{10}$ to (base 2)

   ii. $(AB.CD)_{16}$ to (base 8)

   (d) Subtract $(111)_2$ from $(1110)_2$ using 2's complement method of subtraction.

   (e) Draw the minimised logic circuit diagram for the following boolean expression using two-input logic gates only.

   $$f(a,b,c) = (a+bc)(ac+b)(c+ab)$$

   (f) Find out the dual and complement of the following logic expression.

   $$f(a,b,c) = (a' + bc')(ac' + b')$$

   (g) Show that NAND gate is a universal logic gate.

   (h) Explain with example how to write single-line and multiple-line comments in a C program.

SECTION - B
Answer any five questions.

$[(6 \times 5) = 30]$

2. Write down the truth table and logic expression for a full-adder. Also, show that a full-adder can be made using two half-adders plus an extra logic gate.

3. Write a program to print the following pattern for n ($3 \leq n \leq 5$) number of rows. Here n is a user input.

```
A
B  C
D  E  F
G  H  I  J
```

4. Write a program to generate the Fibonacci sequence up to n^{th} term where n is a user-input.

   $$0\ 1\ 1\ 2\ 3\ 5\ 8\ up\ to\ n^{th} term\ \ldots$$

   The Fibonacci numbers are generated by initializing $F_0 = 0$, $F_1 = 1$ and using the following formula.

   $$F_n = F_{n-1} + F_{n-2}$$

5. Write a program to evaluate the following series. $S = 1! - 2! + 3! - 4! + \ldots$ *up to n^{th} term*. Here n is a user-input.

6. Write a program that can accept n number of digits (0 to 9) and store them in an array. Here, n is a user-input. Next count the frequency of each individual digit present in this array.

7. Write a program to read the elements of a 4×4 matrix of integers. Next, find the sum and average of all the elements of the two diagonals of the matrix

8. Write a function; *int evenORodd(int n)* which returns 0 if n is even and 1 if it is odd.

Indian Institute of Engineering Science and Technology, Shibpur

B.Tech. 2nd Semester (ARCH/CST/EE/ETC/IT) Examinations, 2019

Introduction to Computing (CS 1201)

Full Marks: 35 **Time: 2 Hours**

Answer question no. 1 and any four from the rest.
For the programming problems use C language.

1. ***Answer any five questions.*** [(3 × 5) = 15]

(a) Draw the logic circuit diagram to convert a 4-bit binary number ($b_3b_2b_1b_0$) to its corresponding number $g_3g_2g_1g_0$ in Gray code. Also, draw the truth-table of this converter.

(b) Why 8-bit is enough to encode (say, ASCII encoding) English alphabets, numerals and punctuations? Assume that you are creating your own character code and the character code of the Alphabet 'J' is (chosen to be) n then what it should be for 'C' and 'X'?

(c) Compare the advantages and disadvantages of arrays (1-D) and singly linked list?

(d) What is the format for the IEEE 754 32-bit representation? Also, explain the biased exponent.

(e) Draw the circuit diagram for the following Boolean expression using only NAND gates:

$$f(x, y) = xy + x'$$

(f) Describe any three file opening modes used with the function *fopen()*.

(g) Minimize the following Boolean expression using Karnaugh-Map methdod and hence draw the corresponding logic circuit diagram.

$$f(a, b, c) = a'bc + ab'c' + ab'c + abc' + abc$$

$bc + ab' + ac'$

2. Define a structure to represent complex numbers. Next write a function that will accept two complex numbers as arguments, add these two complex numbers and return the sum to the calling function. Also write *main()* to demonstrate the calling of the function. [5]

3. Write a recursive function to find the highest common factor of two non-negative integers. [5]

4. Write a function that accepts two string arguments and compares the strings byte by byte. It returns the difference of the character values of the first mismatched characters; otherwise zero if there is no mismatch. [5]

5. Write a program that can accept n number of integers and store them in an array. Here, n is a user input. Next sort the numbers in ascending order. [5]

6. Write a program to accept as many numbers (positive integers) as the user likes to input as command line arguments. The program then find the sum and average of these numbers. [5]

7. Write a program to find the solution to the following brain-teaser:

```
   BC
 x BC
 ----
  ABC
```

Note that A, B, and C are digits (0 to 9) and 'x' denotes multiplication. [5]