Department of Computer Science and Engineering

Exam: **Mid Term**   Year: **2021**   Trimester: **Summer**   Course: **CSE 1111/CSI 121**
Title: **Structured Programming Language**   Marks: **20**   Time: **1 hr + 15 min**

Answer all of the Questions given in the **Section-A** and **Section-B**. At first complete all the Questions in **Section-A** and then **Section-B**. Numerical figures in the right margin indicate full marks.

Section-A

Show the **manual tracing** for each of the programs (assume they are syntactically correct) given below. In the programs, LAST_**FOUR**_DIGITS_OF_YOUR_STUDENT_ID (or some other variations) are used. For example, your STUDENT ID is 011202017 and therefore, the value of LAST_ **THREE**_DIGITS_OF_YOUR_STUDENT_ID is 017, and you will use 17. Below, **Use your own student ID**.

1.   **Find** the values of the following variables *a, b, c, d,* and *e*.   **[2.5]**

```
int a = LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID / 7;

int b = LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID % 7;

float c = (float)LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID / 7;

float d = (float)(LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID / 7);

int e = (a-b)<0 && c;
```

2.   **Manual trace** the values of *i,* and ***value_final*** every time their value change.   **[2.5]**

```
int n = LAST_THREE_DIGITS_OF_YOUR_STUDENT_ID % 66;

int i = 0, value_final = 0;

for( i=n-4; i <= n; i++ ){

    value_final += i*i;

    ++i;

}
```

3. In the **manual tracing**, show the value of variable *i* every time its value changes.   **[2.5]**

```
int a = LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID % 3 + 1;
int b = LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID % 2 + 1;
int c = LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID % 3 + 2;
for(int i=0; i<8; i++){
    switch(i){
        case 0:
            i += a;       break;
        case 1:
            i += b;       break;
        case 2:
            i += a;       break;
        case 3:
            i += c;       break;
        case 4:
            i += b;       break;
        case 5:
            i += c;       break;
        default:
            i+=4;        break;
    }
}
```

4. Show the **manual tracing** for the array *A* elements.   **[2.5]**

```
int A[4]={0};
int i, n;
n = LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID;
for(int i=0; i<4; i++){
        A[i] = n+i;
        if(A[i]%2 != 0){
                A[i] *= 2;
        }
}
```

Section-B

5. Write a program to perform the following operations: **[2.5]**

    a) Assign LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID + 3 to integer variable *a*.
    b) Add 11 to *a* and assign that to integer variable *b*.
    c) Divide **b** by 15 and assign that back to **b**.
    d) Add *a* and *b* and assign the sum to a new float variable *c*.
    e) Increment the value of *c* by 1.
    f) Print the values of *a, b* and *c*.

6. Write a program to perform the following operations. Write the "**then**" conditions   **[2.5]**
    using **nested conditional statements**.

    a) Assign (LAST_THREE_DIGITS_OF_YOUR_STUDENT_ID % 17) + 3 to integer
        variable *a*.
    b) Assign (LAST_TWO_DIGITS_OF_YOUR_STUDENT_ID % 21) + 5 to integer variable
        b.
    c) Take an integer variable *c* from user.
    d) If *c* is larger than *a* and *b*, **then** check if *c* is even or odd. If even, then print: "c is
        the largest and even". Otherwise print: "c is largest and odd".
    e) If *c* is larger than *a* only, **then** check if divisible by 3 or not. If divisible, then print:
        "c crossed a and divisible by 3". Otherwise print: "c is not divisible by 3".
    f) If *c* is larger than *b* only, **then** check if divisible by 5 or not. If divisible, then print:
        "c crossed b and divisible by 5". Otherwise print: "c is not divisible by 5".

7. Write a program to compute the summation of the following series.   **[2.5]**

    $$(-j) + (i) + (2i+j) + (3i+2j) +.....+ (ni+(n-1)j)$$

    If the sum is an even number, display your student id; otherwise display your name.

8. Write a program to perform the following operations.   **[2.5]**

    a) Assign (LAST_TWO_DIGITS_OF_YOUR_STUDENT_ID % 21) + 5 to integer variable
        b.
    b) Declare a one-dimensional integer array *A* of size 10.
    c) Initialize the array values with a%7 + 3i, Where a =
        LAST_FOUR_DIGITS_OF_YOUR_STUDENT_ID and i = array index.
    d) Find the sum of the numbers that are stored in even numbered indices in the
        array.