

CSE 1111: Structured Programming Language Assignment -1

CSE 1111-Set A

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 using **nested conditional statements**. [2.5]

- 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]

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

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.

CSE 1111-Set B

1. (a) There are few errors in the following code. Rewrite the code correctly. Do minimum changes. Try not to introduce any new error. [3]

```
#Include<stdlib.h>
int Main(){
    int a, b, float sum;
    scanf("%f%d", a, b);
    sum += a;
    print("%d", &sum);
    return 0;
}
```

- (b) Find the outputs when input values of x are (i) 3, (ii) 10, (iii) 17 and (iv) 21, respectively. [3]

```
scanf("%d", &x);
switch (x){
    case 0: printf("Good"); break;
    case 3: printf("Morning");
    case 7: printf("Hello"); break;
    case 11: printf("World");
    case 17: printf("Best"); break;
    case 21: printf("Wishes!");
    default: printf("Invalid answer");
}
```

- (c) Suppose, Team A and Team B are playing a football match. Write a program that will take two integers as input from the keyboard and print the outcome of the match. The two input integers represent the goals scored by Team A and Team B in order. The team with the higher score wins the match. If both scores are equal, the match is declared a draw. The program displays the result of the match accordingly. Few sample inputs and outputs are given below. [4]

Sample Input	Sample Output
0 0	Draw!
0 2	Team B wins!
3 2	Team A wins!

2. (a) Show the **manual tracing** of variables i , j and sum of the following program. [3]

```
#include<stdio.h>
void main(){
    float sum=0;
    for(int i=10, j=1; i>0; i=i+2, j++){
        sum += i-j;
    }
}
```

- (b) Draw a flowchart that takes an integer n as input from the user and computes the sum of the following series. [3]

$$1^2 - 2^2 + 3^2 - 4^2 + \dots \text{upto } n \text{ terms}$$

(c) Write a program that takes an integer n as input from the user and prints the following pattern using [4] nested loop.

Sample Input, n	Sample Output
3	ABC AB A
5	ABCDE ABCD ABC AB A

4. (a) For the following program,
 (i) Show the manual tracing of variables n and m . [3]
 (ii) Show the output. [3]

```
int main() {
    int arr1[5] = {1,2,3,4,5};
    int arr2[5] = {5,4,3,2,1};
    int arr3[5];

    int m=0, n=0;
    int i, j;
    for(i=0; i<5; i++) {
        for(j=0; j<5; j++) {
            if(arr1[i]==arr2[j]) {
                n++;
                m = arr1[i];
                arr3[i]=m;
            }
        }
        // printf("Output - 1")
        printf("%d : \n", n);
    }
    //printf("Output - 2");
    for(i=4; i>=0; i--) {
        printf("%d : \n", arr3[i]);
    }
    return 0;
}
```

(b) Write a program in C to take input into an integer array of size 50 from user. Find and display the minimum element and its index. Name the array with your firstname. [4]

(c) Write a program to calculate the online average of 4 positive floating point numbers taken from keyboard as inputs. Follow the sample input and output given below for understanding the logic.

Sample Input	Processing	Output on Monitor
num = 10.0	10.0/1=10.0	Average=10.0
num = -5.0	-	-
num = 20.0	(10.0+20.0)/2=15.0	Average=15.0
num = -18.6	-	-
num = 15.6	(10.0+20.0+15.6)/3=15.2	Average=15.2
num = 15.2	(10.0+20.0+15.6+15.2)/4=15.2	Average=15.2

-END-

CSE 1111-Set C

- 1 a) Rewrite the following code after correcting the errors. [2]
- ```
#include<studio.h>
#include<math.H>
int Main (){
 Float n_, m = 5;
 scanf("%d", n_)
 float p = (n_%m)/sqrt(6;
 printf("%d", P);
}
```
- b) Identify the invalid variable names from the following. Mention the reasons that make them invalid. [2]
- Num1, 1time, \$VALUE, "myValue", num6, first\_name, last-name, flx
- c) Compute the values of the variables a, b, c, and d. [2]
- ```
int a = 10.0/3*10;
float b = (int)23.0%11;
int c = (10 > 9 && 21 <= 19)*5;
float d = 7/2;
```
- 2 a) Find the outputs when the input values of variable b are 4, 5, 10 and 12, respectively. [2]
- ```
int b;
scanf("%d", &b);
printf("Begin\n");
if (b>=5)
 printf("UIU\n");
else if(b<=5)
 printf("CSE\n");
else if ((b>=2)&&(b<10))
 printf("COMPUTER\n");
else if ((b>2)&&(b<=10))
 printf("NICE\n");
else
 printf("Bye\n");
printf("End");
```
- b) In a factory there are three categories of employees: X, Y, Z. The manager announced a bonus for the employees who have [4]
- 12 years or more work experience and more than 5 family members,
  - OR
  - Less than 1000.50 BDT total family income per month
- He is also generous to his bonus deprived employees who have a larger family. Thus, He has declared the bonus for
- For the employees of 'Y' and 'Z' categories who have more than 8 family members and has less than 1100.78 BDT total family income per month.
  - However, if an employee is from 'X' category, he can avail the bonus having more than 6 family members.
- Now you need to automate the system by writing a program to take following inputs from user (employee) and notify him whether he is eligible for the bonus or not.
- Category (character)
  - Years of work experience (integer)
  - Number of family members (integer)
  - Total family income per month (float)



- 3 a) Manually trace (show the values of all the variables in each step) the following code segment. [3]

```
int n = 5, sum = 0, i, a = 3, sign = 1;
for(i = 1; i <= n; i++)
{
 sum = sum + a*sign;
 if(i%2==0)
 a = a + 6;
 else
 a = a + 4;
 sign = -sign;
}
printf("\n%d", sum);
```

- b) Write a program to determine whether a number is a **deficient number** or not. A number is deficient if the summation of its factors is less than double of the number. For example, 15 is a deficient number as the sum of its factors (1+3+5+15=24) is less than 30 (double of 15). On the other hand, 24 is not a deficient number. Because the sum of the factors of 24 (1, 2, 3, 4, 6, 8, 12, 24) is 60 which is larger than double of 24. [3]

- 4 a) Show the manual tracing (show the values of all the variables and array elements in each step) for the following code segment [2]

```
int F[6]={0};
int i;
F[0]=1;
F[1]=1;
for(i=2; i<=5; i++){
 F[i]=F[i-1]+F[i-2];
 printf("%d %d %d\n", F[i-2], F[i-1], F[i]);
}
printf("%d %d %d", F[i-2], F[i-1], F[i-1]+F[i-2]);
```

- b) Write a Program that will take n integer numbers into an array, and then find the maximum - minimum among them with the index positions. [4]

| Sample input     | Sample output                        |
|------------------|--------------------------------------|
| 5<br>1 2 3 4 5   | Max: 5, Index: 4<br>Min: 1, Index: 0 |
| 6<br>2 8 3 9 0 1 | Max: 9, Index: 3<br>Min: 0, Index: 4 |

- 5 a) Draw a flowchart that always displays the following menu. It also takes a choice from the user and handle according to the menu. [2]

Enter 1, to display "SPL is fun!"  
Enter 2, to display "(On campus) University life is the Best!"  
Enter q, to quit.

- b) Write a C program that takes an integer value **n**, and draw the pattern below. [4]

| Sample input, n | Sample output                 |
|-----------------|-------------------------------|
| 3               | x x<br>x<br>x x               |
| 5               | x x<br>x x<br>x<br>x x<br>x x |

## CSE 1111-Set D

**Q.1** a) Rewrite the following code after **correcting the errors**.

```
include <stdio.h>
void main() {
 int num1 = 5, float num2, char chr = 'q';
 scanf("%d", num2);
 num1 = num2 % chr;
 printf("Result is = %f ", num1);
}
```

b) **Identify** the **invalid variable names** from the following. **Mention the reasons** that make them invalid.

largest\_val, smallest-val, while, 2ndNum, !New, avg marks, val9

c) **Compute** the values of the variables **a, b, c, and d**.

float a=5\*(5/2), int b=5\*(5/2), float c=5\*(5.0/2), int d=5\*(5.0/2)

**Q.2** a) Write down the **output** of the following C program, for **num = 1** and **num = 3**.

```
#include <stdio.h>
int main() {
 int num;
 int sum = 0, i = 10, j = 5;
 scanf("%d", &num);
 switch(num) {
 case 1:
 sum = 2*i++;
 j++;
 case 2:
 sum = 2*j--;
 i++;
 break;
 case 3:
 sum = ++i*j--;
 case 4:
 sum = i+++j--;
 default:
 sum=0;
 i=0;
 j=0;
 }
 printf("%d %d %d", i, j, sum);
 return 0;
}
```

b) **Manually trace** the following code segment and show the **change of values of the variables i, sum, b, a, y, x** in each step.

```
int sum=0, i, a = 1, b, x = 1, y = 1;
for(i=1; i<=5; i++) {
 sum = sum + a;
 b = 6*x + 1;
 a = a + b;
 y++;
 x = x + y;
}
```



- Q.3** a) **Replace** the nested “for” loop in the following code using nested “do-while” loop **without changing the logical meaning** of the program:

```
void main() {
 int n = 3, i, j, sum = 0;
 for(i = 0; i < n; i++) {
 for(j = 0; j <= i; j++) {
 if(i == j) sum += i + j;
 else if(i > j) sum += i + n;
 else sum += n - j;
 }
 }
}
```

- b) **Write a program** to find the **online average** of the **positive numbers** given as inputs by the user. To solve this problem, you should do the following:
- Write an **infinite loop** that will terminate if the user gives 0 as input.
  - If the user gives a **positive number** as input, you should keep adding it.
  - You should also **keep track** of how many positive numbers are given as inputs.
  - Finally, when the loop terminates, you should **calculate the average** by dividing the sum of the positive numbers by the total positive numbers.

- Q.4** a) Show the **manual tracing** (show the values of all the variables and array elements in each step) for the following code segment.

```
int F[6] = {0};
int i, n;
n = 3;
for(i = 0; i < 6; i++){
 F[i] = n+i;
 if(F[i]%2 == 0){
 F[i] *= 2;
 }
}
```

- b) **Write a program** to perform the following operation:
- Read **n integer numbers** from keyboard and **store** them in an array of size 100, where n is input integer from keyboard.
  - Print** all the array elements with their indices (plural of index) in the following format.

| Index | Value |
|-------|-------|
| 0     | 11    |
| 1     | 7     |
| ..    | ..    |

- Find and print** the **average** of the numbers that are stored in **odd numbered indices** in the array.

- Q.5** a) **Draw a flowchart** to find the **sum** of the following series up to n terms, where n is input integer number from keyboard.

$$1 - 2 + 3 - 4 + \dots \text{ upto } n \text{ terms}$$

- b) **Write a program** that takes an integer **n** as input from the user and **prints** the following **pattern**. Program for n, NOT 3 or 5.

| Sample input, n | Sample output |   |   |   |   |
|-----------------|---------------|---|---|---|---|
| 3               | 6             | 4 | 2 |   |   |
|                 | 4             | 2 |   |   |   |
|                 | 2             |   |   |   |   |
| 5               | 10            | 8 | 6 | 4 | 2 |
|                 | 8             | 6 | 4 | 2 |   |
|                 | 6             | 4 | 2 |   |   |
|                 | 4             | 2 |   |   |   |
|                 | 2             |   |   |   |   |