United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

<u>Final Exam:: Trimester: Summer 2024</u>

Course Code: CSE 1111, Course Title: Structured Programming Language

Total Marks: **50** Duration: **2 hours**

[<u>Any examinee found adopting unfair means will be expelled from the trimester/program as per UIU disciplinary rules.</u>]

There are <u>FIVE</u> questions. Answer <u>all</u> the questions. Marks are indicated in the right margin.

Q.1 a) Implement **getSumMean(float marks[ ], int N, int type)** function, which computes **sum and mean** of marks of N students from an input array A.

i. In the **main()** function, **initialize** an array of floats to store the marks of **100 students**. Take the values from the user as input. **[1]**

ii. Take **2 values** as input from user: an integer (**number of elements**) & an integer (**value type**). **[1]**

iii. Call the **getSumMean()** function from the **main()** passing those values. If the value type is 1, then calculate and return the sum of N elements in the array A. If the value type is 2, then calculate and return the average/mean of N elements in the array A. **[2]**

iv. What type of argument passing is used in **getSumMean()** function call? Call-by-value, or call-by-reference? **Justify and explain**. **[1]**

b) Find the **output** of the following program (left). Notice the **local and global contexts**. **[5]**

```
int x = 6, y = 8;
int func (int a, int b) {
   a *= 2;
   printf("a = %d, b = %d.\n", a, b);
   return (--a) * (b--);
}
int sub (int a, int b) {
   b /= 6;
   printf("a = %d, b = %d.\n", a, b);
   return (--a) * (--b);
}
int main() {
   y = func(x, y);
   printf("x = %d, y = %d.\n", x, y);
   x = sub(x, y);
   printf("x = %d, y = %d.\n", x, y);
   return 0;
}
```

C Code for **1(b)**

```
#include <stdio.h>
#include <string.h>

int main() {
    char A[50] = "Structured";
    char B[50] = "Coding";
    int len = strlen(A);
    for (int i=0; i<strlen(B); i++)
      A[len-i-1] = B[i];

    strncat(A, B, 2);

for (int j=0; j<4; j++)
      A[j] = B[strlen(B)-j-1];
    printf("A: %s\n", A);
    printf("B: %s\n", B);
    return 0;
}
```

C Code for **2(a)**

Q.2 a) Show **manual tracing** (every change) of variables i, j, A, and B of the program above at right. **[5]**

b) Write a **C program** to replace **every nth** character of a string with the letter **'Z'**. You **cannot** use any **library functions**. For example, if string contains "**HelloWorld**" and **n=3**, print "**HeZloZorZd**". Another example: if string contains "**Programming**" and **n=2**, print "**PZoZrZmZiZg**". **[5]**

Q.3 a) **Identify** and **correct the errors** of the following code: **[3]**

```
struct Gadget {
     int serialNo,
     char category[],
     float price;
}
int main() {
     struct g1;
     serialNo = 123;
     category = "Tab";
     price = 5500.00;
     struct *gPtr = g1;
     scanf("%s",&gPtr.category);
     scanf("%s",&gPtr.price);

}
```

Q.3 b) Suppose you are developing a **C program** for the **UIU Sports Club** to **store and manage** players' performance for the **VC Cup football Season 3** to be held in 2025. Your task is to create a **C program** that can store, manage and manipulate the data to select **best goal scorer**. Write a C program that will:

i. **Create a structure** named **player** to **store** the following information of each player participating in the tournament as follows (use **appropriate data types and variable names** for all the features): **[1]**
  - Name
  - ID
  - Number of matches played, and
  - An array **G[ ]** of scored goals in each of the matches played.

ii. **Scan information** as input from the user for **120 players** registered in the tournament. **[1]**

iii. Write a function **goalsScored(struct player p, int matches)** that takes a **player** structure and total **number of matches played** and returns the **sum of the elements** of array **G[ ]** to get **total goals scored** by a particular player. **[2]**

iv. Write a function **selectBestScorer(struct player b[ ], int total_players)** that takes an array of **player** structure and total number of players and returns the **index of the player array** representing the player with **highest number of goals** scored. **[3]**

Q.4 a) Write the **output** of the program provided below on the left. What type of argument passing is used in **mystery()** function call? Call-by-value, or call-by-reference? **Justify and explain**. **[5]**

b) Find the **output** of the code provided below on the right. **[5]**

```
int fun(int *a, int *b){
     if (*a > 1)
       return *a + *b;
     else
       return *b - *a;
}

int main(void) {
      int x = -10;
      int y = 20;

      printf("The result is %d",
             fun(&x, &y));
}
```

C Code for **4(a)**

```
void f(char *s1, char *s2, int n, int *a) {
   int i;
   for(i=0; i<n; i++) {
      if(s1[i] > s2[i]) *a = *a+1;
      else if(s1[i]<s2[i]) *(s1+i)=*(s2+i);
      else if(s1[i] == s2[i]) *a = *a-1;
   }
}
void main() {
   int a, b;
   char s1[] = "worldcupBangladesh";
   char s2[] = "BanglaWash";
   a = 0;
   f(s1,s2,13,&a);
   printf("%d\n s1=%s , s2=%s\n",a,s1,s2);
}
```

C Code for **4(b)**

Q.5 a) Write a **C program** that does the following: **[5]**

- Declare an integer array **A** with arraysize 20, a **pointer** variable **aPtr** and **assign** array **A** to **aPtr**.
- Scan the elements of the array **A** using the pointer **aPtr** with **offset**.
- Write a **function swap(int *a, int *b)** to swap any 2 values from array **A[ ]** using **pointers**. Call swap() function from the **main()** to swap array elements **A[5]** and **A[11]**.

b) Suppose, a file "**students.dat**" is created, where each line contains a **student's name** followed by his/her **cgpa** and the file already have information of 5 students. Now, answer the following questions:

i. What is the difference between **read, write** and **append** mode? **[1]**
ii. Which **mode** you should open the file? What will happen if the file **does not exist**? **[1]**
iii. Write corresponding **C program** to open the file, accomplish the above task and close the file. **[3]**