Programming Fundamentals Mock Fall 2024

By MLSA FAST Karachi 2024-25

Max Marks: 100

Date: 30 November 2024	Time: 3 Hours
Compiled By:	Muhammad Ali Hadi, Krish Talreja, Usman Ahmed, Anas Khan

Question-1: **Marks: 20**

(a) You must predict the outputs of following code snippets: [12]

(i)

```
#include <stdio.h>

int main() {
   int x = 7;
   printf("%d\n", (x > 5) && (x < 10));
   printf("%d\n", (x < 5) || (x == 7));
   return 0;
}
```

(ii)

```
#include <stdio.h>

int main() {
   char str[] = "Programming";
   char *ptr = str;
   printf("%c\n", *(ptr + 4));
   printf("%s\n", ptr + 5);
   return 0;
}
```

(iii)

```
#include<stdio.h>

int foo(int num)
{
        if(num<10)
        return num;
        int sum=num/10+foo(num%10);
        if(sum>9)
        return sum/10+foo(sum%10);
        else
        return sum;
}

int main()
{
        printf("Sum : %d\n",foo(456));
        printf("Sum : %d",foo(97));
        return 0;
}
```

(b) You must write the corrected code if you find any error in the following codes:[08]

(i)

```
#include <stdio.h>

int main() {
   int i, sum = 0;
   for (i = 0; i <= 10; i++) {
      if (i % 2 = 1)
         sum += i;
   }
   printf("Sum of odd numbers = %d\n", sum);
   return 0;
}
```

(ii)

```
#include <stdio.h>

int main() {
   int arr[5] = {1, 2, 3, 4, 5};
   for (int i = 0; i <= 5; i++) {
      printf("%d\n", arr[i]);
   }
   return 0;
}
```

Question-2: Marks: 24

Imagine you are tasked with developing a management system for a library's CD collection. Each CD has a unique ID, title, artist name, genre, and the number of tracks. The system needs to allow users to:

- Add new CDs dynamically.
- Delete CDs by their ID.
- Search for CDs by title or artist name.
- Update CD details.

To enhance functionality, the system should also allow users to:

- Sort CDs by title, artist name, or number of tracks using a simple sorting algorithm
- You must use structures to represent individual CDs and dynamic memory allocation to manage a collection of CDs that grows or shrinks based on user actions.
- Also write a method to print the data set in following manner after any operation is performed on the dataset.

Sorting Task:

Write a program that sorts the collection by:

- Title (alphabetically).
- Artist name (alphabetically).
- Number of tracks (ascending).

Searching Task:

After sorting, implement a simple search algorithm to find a CD by its title or artist name. If a match is found, display the CD details; otherwise, return a message indicating no match.

Question-3: Marks: 20

Implement a recursive function given below to convert a non-negative integer number to its English word representation:

[Hint: You can create global variables or arrays to implement]

Here, “num” ranges between 0 to 10000 so handle cases according to it.

Sample Input: num= 798

Output: Seven Hundred Ninety Eight

Function: (Implement Function Definition given below)

```
char* numToWords(int num)
{


}
```

Question-4: Marks: 18

You are required to make a basic student management system in which you have to do the following things.

1- Create a structure with name Student , which will have the following data members:

a- Name

b- Roll Number (char array)

c- GPA

Note : Name and roll number should not have spacing.

2- Implement a bubble sort algorithm which sorts the student by their name

3- Save the data into a "student.txt" file

4- Retrieve the data from the file and display it.

Note: you can use string functions for the sorting algo.

Question-5: Marks: 18

Advanced Scenario with Functions, Structures, and File Handling

Scenario:

A library wants to automate its book borrowing system. Write a program that:

1. Defines a structure **Book** with fields:
    - **id** (integer)
    - **title** (character array of size 100)
    - **author** (character array of size 50)
    - **isAvailable** (integer: 1 for available, 0 for borrowed)
2. Provides the following functionalities using functions:
    - **Add Books**: Adds details of n books to a file named **"library.txt"**.
    - **Borrow Book**: Marks a book as borrowed if it's available.
    - **Return Book**: Marks a book as returned.
    - **List Books**: Displays the list of all books with their availability status.