Bangladesh Army University of Science and Technology (BAUST), Saidpur

Department of Computer Science and Engineering (CSE)

Final Examination, Summer 2024

m: B.Sc. Engg. in CSE	**Course Code: CSE 1101 (Syl. 2021)**	**Level- 1 Term- I**
)3 (Three) hours	**Credit: 3.00**	**Full Marks: 180**

Course Title: Structured Programming Language

- Symbols and abbreviations bear their standard meanings.
- Use separate answer script for each PART.
- Assume reasonable values for any missing data.

Part-A (Marks: 90)

(Answer any three questions including Question No. 1)

	Marks	CLOs
Explain how computers can understand our commands and perform the tasks we want the computers to do.	(10)	1
Describe the basic structure of a C program using an example.	(10)	1
State some reasons why we should learn to program using C programming language.	(10)	1
Differentiate between keywords and identifiers in C.	(5)	
Summarize the rules for naming identifiers. Analyze and determine which of the following identifiers are valid or invalid. If invalid, explain why it is.	(10)	

BaUsT&	1stsemester	Is_Not_Valid_3	my-name	printf

Write a C program that takes three marks as input, calculates and prints the best, worst and average marks.	(15)	
If a year is divisible by four, it is a leap year. But if the year can be divided by 100 as well as four, it is not a leap year. However, if the year is divisible by 400, it is a leap year. Now, write a C program that repeatedly takes a year as input, checks and prints if it is leap year and exit the program only when user gives zero (0) as input.	(10)	
The Fibonacci sequence is a sequence where the next term is the sum of the previous two terms. The first two terms of the Fibonacci sequence are 0 followed by 1. Now, write a C Program to display the Fibonacci sequence of first **n** numbers, where **n** is a positive integer less than 100.	(10)	
Provide an example of infinite loop. Demonstrate how break and continue is used to control the flow of a loop.	(10)	
Write a C program that takes 10 numbers as input, save them in an array, sort the numbers inside the array in descending (large to small) order and print the array.	(10)	
Write a C program to print a pyramid of stars that looks like as follows:	(10)	

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

Explain the purpose, syntax, and the associated header file for the following string functions: strcpy(), strlen(), strrev(), strcat(), and strncmp().	(10)	

Part-B (Marks: 90)

(Answer any three questions including Question No. 5)

5. a) Provide suitable array definitions for the given scenarios:
   - i) Create a one-dimensional integer array named ***data*** with 10 elements. Assign ***10, 20, 30 … 100*** values to the array elements. Access the last element from the array.
   - ii) Establish a one-dimensional character array named ***point*** and assign the string ***"NORTH"*** to the array elements. Conclude the string with the null character.
   - iii) Formulate a one-dimensional character array named ***letters*** with four elements. Assign the characters ***'N'***, ***'E'***, ***'W'***, and ***'S'*** to the array elements.

   b) Write a C program to demonstrate the use of 2D array by performing the following tasks:
   - i) Define a 2D array of integers with size ***m * n***, where m and n are provided by the user.
   - ii) Prompt the user to input ***m* n*** elements for the 2D array and
   - iii) Implement the following operations:
     - Display all elements of the 2D array in a matrix format.
     - Calculate and display the sum of the diagonal elements in a ***file***.
     - Find and display the maximum or minimum elements in the 2D array using ***function*** and ***file*** (. e.g., ***find_MAX()*** or ***find_MIN()***)

6. a) Write a C program that demonstrates the use of pointers with arrays to perform the following tasks:
   - i) Declare an integer array of size 5 and initialize it with some values.
   - ii) Declare a pointer variable to point to the first element of the array.
   - iii) Access and print the values of the array elements using both array notation and pointer arithmetic.
   - iv) Modify the values of the array elements using pointer dereferencing. Ensure that the program compiles without errors and produces the correct output.

   b) Consider the following C code which demonstrates the relationship between arrays and pointers in C, and how pointer arithmetic can be used to access array elements. Also analyze the code and write the output produced by each line of the printf statements.

```
                         int a[5]={10,20,30,40,50};
                         int *p=a;   // Not int *p=&a;
printf("\n a =%x p = %x ",a,p);                printf("\n a[1]=%x p[1]=%x",&a[1], &p[1]);
printf("\n a =%x  p= %x ",&a,&p);              printf("\n a[1]=%x p[1]= %x ", a+1,p+1);
printf("\n a[0] =%x p[0] =%x ",a[0],p[0]);     printf("\n a[1]=%x p[1]=%x ", a[1],p[1]);
printf("\n a[0] =%d p[0] =%d ",a[0],p[0]);     printf("\n a[1]=%d p[1]= %d ", *(a+1),*(p+1).
```

7. a) Explain the concepts of "Call by Value" and "Call by Reference" in C functions. Provide a code example for each to illustrate the difference between these two parameter-passing methods.

   b) Define ***malloc(), calloc(), realloc(),*** *and* ***free()*** functions with suitable examples. Compare and contrast ***malloc***() with ***calloc***().

   c) Write a C program to check if a given string is a palindrome without using any standard library functions. A palindrome is a string that reads the same forwards and backwards (***e.g., "MADAM" or "RACECAR"***).

   The program should:
   - Prompt the user to enter a string (up to a specified maximum length).
   - Dynamically allocate memory for the string using ***malloc() or calloc()***.
   - Implement a function to check if the string is a palindrome.
   - Release the allocated memory using ***free()***.

   Sample Input and Output:
   - ***Sample Input 1:*** *Enter a string: MADAM*
   - ***Sample Output 1:*** *The string "MADAM" is a palindrome.*

- ***Sample Input 2:*** *Enter a string: HELLO*
- ***Sample Output 2:*** *The string "HELLO" is not a palindrome.*

Imagine you are designing a student records file with the following four attributes: a unique long integer named ID, a character array (string) named name, a character representing sex, and a double-precision number named CGPA. Now, follow these tasks: (30)

i) Define both a structure and a union to represent this data structure as described. Declare structure variable 'x' and union variable 'y' for this purpose.

ii) Examine the memory requirements of variables 'x' and 'y.' Do they consume the same amount of memory? If not, please explain the reason.

iii) Declare a structure variable named 'p' and a union variable named 'q' with the same attributes as described earlier. Initialize both 'p' and 'q' with the following values during declaration: ***ID = 802320105101039, name = "" sex = 'M' and CGPA = 3.95.***

iv) Discuss the pros and cons of using structures versus unions for storing above student record.