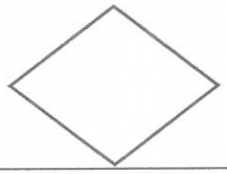
Set A

Scoring Indicators

COURSE NAME: PROBLEM SOLVING AND PROGRAMMING

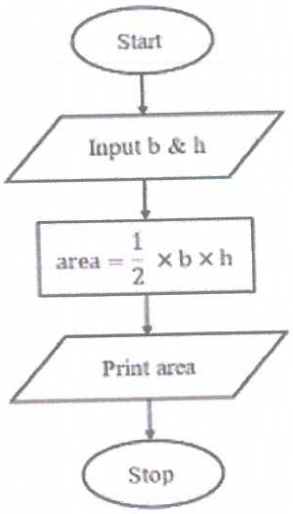
COURSE CODE: 2131 **QID: 2106220124**

Q No	Scoring Indicators	Split score	Sub Total	Total score
	PART A			**9**
I. 1		1 mark	1 mark	1 marks
I. 2	#include, #define, #error,#endif (any two)	½ mark each	1 mark	
I. 3	++ and --	½ mark each	1 mark	
I. 4	for(initialization;condition;updation)	1 mark	1 mark	
I. 5	11	1 mark	1 mark	
I. 6	A function is a group of statements that together perform a task. Every C program has at least one function, which is main().	1 mark	1 mark	9 marks
I. 7	return	1 mark	1 mark	
I. 8	int arr[3][2]={{3,4},{1,2},{0,6}}	1 mark	1 mark	
I. 9	10	1 mark	1 mark	
	PART B			24
II. 1	1. Analyze the problem. 2. Design the program. 3. Code the program 4. Debug the program. 5. Formalize the solution. 6. Document and maintain the program	½ mark each	3 marks	3 marks
II. 2	ans_1=7 ans_2=-1	1½ mark each	3 Marks	3 marks

II. 3	**Break** / **continue** It is used to terminate the enclosing loop like while, do-while, for, or switch statement where it is declared. / It terminates only the current iteration of the loop. It causes early termination of loop / It causes early execution of the next iteration. It is used with 'switch' and 'label' since it is compatible. / It can't be used with 'switch' and 'label' since it is not compatible.	½ mark each	1½+1½=3 Marks	3 marks
II. 4	• input and output • logic	1 mark 2 marks	3 marks	3 marks
II. 5	0 1 2 3 4 5 6 7 8 9 10	3 marks	3 marks	
II. 6	1. function declaration 2. function call 3. function definition	1 mark 1 mark 1 mark	3 marks	3 marks
II. 7	**Formal Parameter** / **Actual parameter** Formal parameters are always variables / Actual parameters do not have to be variables. Parameter will be in the called function. / Parameter Written in Function Call is Called "Actual Parameter". The data type must be included. / Data types will not be mentioned.	½ mark each	1½+1½= 3 Marks	3 marks
II. 8	Input and output Logic • Divide the input number by 8 and obtain its remainder and quotient. Store the remainder in the array. • Print the array in reverse order	1 mark 2 marks	1+2= 3 Marks	3 Marks
II.9	**One dimensional Array** / **Two dimensional Array** Stores a single list of the elements of same data type. / Stores a 'list of lists' of the elements of same data type. Represent multiple data items as a list. / Represent multiple data items as a table consisting of rows and columns. Eg. int a[10]; / Eg. int a[2][2]	½ mark each	1½+1½= 3 Marks	3 marks
II.10	x[0]=3 x[1]=5 x[2]=1 x[3]=2 x[4]=-1	3 marks	3 marks	3 marks

	PART C			42
III. 1	a)	2Marks		

Flowchart



3 marks | 3Marks

b) Formatted I/O functions

i) They are used to take various inputs from the user and display multiple outputs to the user.

ii) These types of I/O functions can help to display the output to the user in different formats using the format specifiers.

7 Marks

2 marks

printf():

- printf() function is used in a C program to display any value like float, integer, character, string, etc on the console screen.
- It is a pre-defined function that is already declared in the stdio.h.

2+2= 4 marks

Syntax :

printf("Format Specifier", var1, var2, …., varn);

2 marks

scanf():

- scanf() function is used in the C program for reading or taking any value from the keyboard by the user.
- These values can be of any data type like integer, float, character, string, and many more.
- This function is declared in stdio.h.

Syntax:

scanf("Format Specifier", &var1, &var2, …., &varn);

OR

III. 2 a)

1. **Operator precedence** determines the grouping of terms in an expression and decides how an expression is evaluated. — 2 marks
2. Certain operators have higher precedence than others within an expression, higher precedence operators will be evaluated first.
3. **Associativity,** defines the order in which the operators of the same precedence will be evaluated in an expression. — 1 marks

2+1+4=7 Marks

	Here, operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. **Category** / **Operator** / **Associativity** Multiplicative: * / % — Left to right Additive: + - — Left to right Relational: < <= > >= — Left to right Logical: |,^,&&,|| — Left to right	4 marks		7 Marks
III. 3	Selection statement 1. Simple if statement 2. if- else statement 3. Nested if - else 4. if –else -if **(any two statement with example)**	1mark 3 Marks each	1+6=7 marks	7 marks
	OR			

III. 4	A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each **switch case.**
	syntax
	switch(expression){
	case value1:
	//code to be executed;
	break; //optional
	case value2:
	//code to be executed;
	break; //optional
	.....
	default:
	code to be executed if all cases are not matched;
	}
	example
		1 marks
		3 Marks
		3 Marks
			1+3=4 marks
			4+3=7 marks
				7marks

III. 5	**Looping statements** A loop statement allows us to execute a statement or group of statements multiple times. **Types of loops** 1. <u>while loop</u> syntax & example 2. <u>for loop</u>	1 Marks 3 Marks each	1+6= 7 marks	7 Marks

	syntax & example 3. do… while loop syntax & example **(any two statement with example)**			
	OR			
III. 6	Palindrome or not • Input • Logic • Output	1 mark 5 marks 1 mark	1+5+1= 7 marks	7 marks
III. 7	**a)Fibonacci series –Program** • input • Logic (function declaration, main(),function call, function definition) • Output	1 mark 5 marks 1 mark	1+5+1=7 marks	7 marks
	OR			
III. 8	a) Modular programming – Explanation b) 1. sum() 2. formal parameter –a,b, actual parameter – 5,4 3. int sum (int a,int b); 4. 2	3 marks 1 mark each	3+4 =7 marks	7 marks
III. 9	1. input count and array elements 2. logic(array processing) 3. output	3 marks 3 marks 1 marks	3+3+1= 7 marks	7 marks
	OR			
III. 10	1. input count, array elements ,searching number 2. logic (array processing) 3. output	2 mark 3 marks 2 marks	2+3+2=7 marks	7 marks
III. 11	a) display the diagonal elements in the matrix 1. input order of the matrix 2. input elements of the matrix 3. processing array (logic) 4. display the array which stores diagonal elements	1 mark 2 marks 2 marks 2 marks	1+2+2+2= 7 marks	7 marks
	OR			
III. 12	a) matrix addition 1. input order of matrices 2. input elements of the matrices using for loop 3. processing of the matrices (addition) 4. output resultant matrix using for loop statement	1 mark 2 marks 2 marks 2 marks	1+2+2+2= 7 marks	7 marks