

289

51

AK. 1 Page: 12
April: 2024

Scoring Indicators

Course Name: Problem Solving and Programming
Course code: 2131

QID: 2103230002

Q NO	Scoring Indicators	Split score	Subtotal	Total score
PART-A				9
1	Flowchart	1	1	
2	printf()	1	1	
3	(), *, <=, &&	1	1	
4	sizeof()	1	1	
5	Break	1	1	
6	Functions	1	1	
7	True	1	1	
8	19	1	1	
9	Array of arrays to store homogeneous data in tabular form	1	1	

PART-B			Split score	Subtotal	Total score
					24
1	Flowchart to calculate area($l*b$) Flowchart to calculate perimeter $2*(l+b)$	1.5 each			
2	Logical AND, OR, NOT with exact truth table	1 each			
3	a) $y=10$ b) $y=11$	1.5 each			

4	<p>Syntax Condition ? Statement1:Statement 2 If condition is true statement 1 will be executed else statement2 will be executed</p> <p>Example</p>	1.5each		
5	<p>Break is used to exit from a loop And next instruction in the program will be executed Continue is used to skip one single iteration of a loop And continues with the next iteration of the loop</p> <p>Difference-2marks Example-1/2 mark each</p>			
6	<pre>i=1; while(i<=10) { printf(“%d”,i); i++; }</pre>	3marks		
7	<p>sqrt()-Math library to find square root of a number in math.h isdigit()-to check whether number isalpha()-to check whether alphabet islower()-to check whether lowercase pow()-Math function to compute power</p> <p>(Any 3 1 mark each)</p>	1		
8	<ul style="list-style-type: none"> In the Call by value method original value is not modified, whereas in the Call by reference method, the original value is modified. In Call by value, a copy of the variable is passed, whereas in Call by reference, a variable itself is passed. In Call by value, actual and formal arguments will be created in different memory locations, whereas in Call by reference, actual and formal arguments will be created in the same memory location. 	1each		
9	<p>Array-1mark</p> <p><u>Program Code</u></p> <pre>#include <stdio.h> int main() { int n,max; int arr[50]; printf("Enter the number of elements (1 to 100): "); scanf("%d", &n); for (int i = 0; i < 6; ++i) { printf("Enter number%d: ", i + 1);</pre>			

	<pre>scanf("%d", &arr[i]); } max=arr[0]; // storing the largest number to arr[0] for (int i = 0; i < 6; ++i) { if (max < arr[i]) { max = arr[i]; } } printf("Highest mark = %d", max); return 0; }</pre>			
10	<pre>int arr[3][2]={{1,2},{2,3},{3,4},};</pre> <p>Declaration - 1.5mark Initialization- 1.5mark</p>			

		PART C		56
III	It is a variable that stores data type value in a program.	It is similar to a variable and cannot be changed during program execution.		
	It is a variable that can be changed after defining the variable in a program	It is a fixed variable that cannot be changed after defining the variable in a program.		
	The value of a variable can change depending on the conditions.	In constants, the value cannot be changed.		
	Typically, it uses int, float, char, string, double, etc. data types in a program.	It can be expressed in two ways: #define pre-processor and the const keyword.		
	Example: int a = 5; float radius = 5.2; char 'A';	Example: const int Len = 5; #define PI 3.14		
	Keyword is a pre-defined word.	The identifier is a user-defined word		
	It must be written in a lowercase letter.	It can be written in both lowercase and uppercase letters.		
Its meaning is pre-defined in	Its meaning is not defined in			

	the c compiler. It is a combination of alphabetical characters. It does not contain the underscore character.	the c compiler. It is a combination of alphanumeric characters. It can contain the underscore character.			
	Any 3 difference from each 1.5mak each				
	<i>OR</i>				
IV	Unary –Operator with single operand eg-+,++,-- Binary-Operator with 2 operands +,-,*,/		1.5 each		
V	STRUCTURE(2marks) Include header file Global declaration Function main Declaration Execution Function call(optional) Comments(optional) Example(1mark)				
	<i>OR</i>				
VI	<pre>#include<stdio.h> int main() { int a=10, b=20; printf("Before swap a=%d b=%d",a,b); a=a+b; b=a-b;// a=a-b;// printf("\nAfter swap a=%d b=%d",a,b); return 0; }</pre>				
VII	Usage of scanf in exact format-2marks Condition if (num%2==0)even Else odd-2marks Overall correctness -3marks				

VIII	<p>Switch statement is decision making statement multiple conditions With explanation-2marks switch(case variable) { case 1://statements break; case 2://statements break; case 3://statements break; . . . default://statements }(2marks) Example-3marks</p>			
IX	<p><u>While loop</u> initialization expression; while(condition) { //body of loop } with example and explanation(3.5 marks)</p> <p><u>do while</u> do { //body of loop }while(condition); with example and explanation(3.5 marks)</p>			
OR				
X	<p>Using while loop or for loop Compute fact=fact*i until i is less than than1</p> <p>Logic-2marks Exact Selection and syntax of loop-3marks Overall correctness-2marks</p>			

XI	<p>a) Function prototype includes</p> <ul style="list-style-type: none"> • Return type • Function name • Parameter list • Body of function • Return value <p style="text-align: right;">(2marks)</p> <p>Return-type function_name (parameter list)</p> <pre>{ //body return value; }</pre> <p style="text-align: right;">(2marks)</p> <p>b)</p> <pre>int sum_dig(int n) { int x=0; while(n>0) { y=n%10; x=y+x; n=n/10; } return x; }</pre> <p style="text-align: right;">(2marks)</p> <p>Calling the function and displaying function in main program (1mark)</p>	N=	
OR			
XII	<p>a)</p> <ul style="list-style-type: none"> • In call by value method, the value of the actual parameters is copied into the formal parameters. In other words, we can say that the value of the variable is used in the function call in the call by value method. • In call by value method, we can not modify the value of the actual parameter by the formal parameter. • In call by value, different memory is allocated for actual and formal parameters since the value of the actual parameter is copied into the formal parameter. • The actual parameter is the argument which is used in the function call whereas formal parameter is the argument which is used in the function definition. <ul style="list-style-type: none"> • In call by reference, the address of the variable is passed into the function call as the actual parameter. • The value of the actual parameters can be modified by changing the formal parameters since the address of the actual parameters is passed. • In call by reference, the memory allocation is similar for both formal parameters and actual parameters. 		

	<ul style="list-style-type: none"> • All the operations in the function are performed on the value stored at the address of the actual parameters, and the modified value gets stored at the same address. <p>ANY 3 differences 3marks</p> <p>b) Create a function to check prime number with exact logic-2 marks Calling function and displaying result-1mark</p>		
<p>XII</p> <p>I</p>	<pre>#include <stdio.h> void main() { int i, j, a, n, number[30]; printf("Enter the value of N \n"); scanf("%d", &n); printf("Enter the numbers \n"); for (i = 0; i < n; ++i) scanf("%d", &number[i]); for (i = 0; i < n; ++i) { for (j = i + 1; j < n; ++j) { if (number[i] > number[j]) { a = number[i]; number[i] = number[j]; number[j] = a; } } } printf("The numbers arranged in ascending order are given below \n"); for (i = 0; i < n; ++i) printf("%d\n", number[i]); } Exact program-7marks</pre>		
	OR		
<p>XIV</p>	<pre>#include <stdio.h> void main () {</pre>		

```

int array[10][10];
int i, j, m, n, a = 0, sum = 0;

printf("Enetr the order of the matix \n");
scanf("%d %d", &m, &n);

if (m == n )
{
printf("Enter the co-efficients of the matrix\n");
for (i = 0; i < m; ++i)
{
for (j = 0; j < n; ++j)
{
scanf("%d", &array[i][j]);
}
}

printf("The given matrix is \n");
for (i = 0; i < m; ++i)
{
for (j = 0; j < n; ++j)
{
printf(" %d", array[i][j]);
}
printf("\n");
}

for (i = 0; i < m; ++i)
{
sum = sum + array[i][i];
}

printf("\nThe sum of the main diagonal elements is = %d\n", sum);
}

else
printf("The given order is not square matrix\n");
}

```

Exact program-7 marks

```

int array[10][10];
int i, j, m, n, a = 0, sum = 0;

printf("Enetr the order of the matix \n");
scanf("%d %d", &m, &n);

if (m == n )
{

printf("Enter the co-efficients of the matrix\n");
for (i = 0; i < m; ++i)
{
for (j = 0; j < n; ++j)
{
scanf("%d", &array[i][j]);
}
}

printf("The given matrix is \n");
for (i = 0; i < m; ++i)
{
for (j = 0; j < n; ++j)
{
printf(" %d", array[i][j]);
}
printf("\n");
}

for (i = 0; i < m; ++i)
{
sum = sum + array[i][i];
}

printf("\n\nThe sum of the main diagonal elements is = %d\n", sum);
}

else
printf("The given order is not square matrix\n");

}

```

Exact program-7 marks

Blue Print

Mark Distribution

Module	Hours/Module (hi)	Marks/Module ($h_i/\sum H_i$) * 123 (±5%)	Type of Questions							
			Part A		Part B		Part C		Total	
			No. of questions	Marks	No. of questions	Marks	No. of questions	Marks	No. of questions	Marks
1	11	31	2	2	2	6	2	14	6	22
2	11	31	3	3	3	9	4	28	10	40
3	10	30	2	2	3	9	4	28	9	39
4	11	31	2	2	2	6	2	14	6	22
Total	43	123	9	9	10	30	12	84	31	123

Cognitive Level Distribution

Cognitive Level	Marks	% of Marks
Remembering	50	41
Understanding	38	31
Applying	35	28
Total	123	100

BLUE PRINT

Mark Distribution

Module	Hr / Module	$(h_i / \sum H_i) * 123$	TYPE OF QUESTIONS							
			PART A		PART B		PART C		TOTAL	
			No of Questions	Marks	No of Questions	Marks	No of Questions	Marks	No of Questions	Marks
I	8	22.8	3	3	3	9	4	28	10	40
II	15	43.05	2	2	3	9	4	28	10	39
III	8	22.8	2	2	2	6	2	14	6	22
IV	12	33.21	2	2	2	6	2	14	6	22
Total	43	123								
			9	9	10	30	12	84	32	123

Cognitive Level Wise Question Analysis

Mark Distribution

Cognitive Level	% Marks	Marks	TYPE OF QUESTIONS							
			PART A		PART B		PART C		TOTAL	
			No of Questions	Marks	No of Questions	Marks	No of Questions	Marks	No of Questions	Marks
R			3	3	5	15	1	7	9	25
U			4	4	3	9	5	35	12	48
A			2	2	2	6	6	42	10	50
Total	100									
			9	9	10	30	12	84	31	123

STATE PRINT

and

LIST OF STATES

STATE	POPULATION	AREA	POPULATION	AREA	POPULATION	AREA	POPULATION	AREA
Alabama	2,049,000	52,400	Alabama	2,049,000	52,400	Alabama	2,049,000	52,400
Alaska	277,000	588,000	Alaska	277,000	588,000	Alaska	277,000	588,000
Arizona	1,512,000	113,900	Arizona	1,512,000	113,900	Arizona	1,512,000	113,900
Arkansas	1,188,000	53,100	Arkansas	1,188,000	53,100	Arkansas	1,188,000	53,100
California	15,981,000	155,900	California	15,981,000	155,900	California	15,981,000	155,900
Colorado	2,766,000	104,000	Colorado	2,766,000	104,000	Colorado	2,766,000	104,000
Connecticut	3,287,000	5,500	Connecticut	3,287,000	5,500	Connecticut	3,287,000	5,500
Delaware	787,000	2,400	Delaware	787,000	2,400	Delaware	787,000	2,400
Florida	17,642,000	65,700	Florida	17,642,000	65,700	Florida	17,642,000	65,700
Georgia	4,779,000	59,700	Georgia	4,779,000	59,700	Georgia	4,779,000	59,700
Hawaii	1,059,000	10,900	Hawaii	1,059,000	10,900	Hawaii	1,059,000	10,900
Idaho	1,208,000	141,600	Idaho	1,208,000	141,600	Idaho	1,208,000	141,600
Illinois	12,121,000	149,900	Illinois	12,121,000	149,900	Illinois	12,121,000	149,900
Indiana	6,081,000	36,400	Indiana	6,081,000	36,400	Indiana	6,081,000	36,400
Iowa	2,998,000	145,700	Iowa	2,998,000	145,700	Iowa	2,998,000	145,700
Kansas	1,781,000	175,000	Kansas	1,781,000	175,000	Kansas	1,781,000	175,000
Kentucky	3,996,000	40,400	Kentucky	3,996,000	40,400	Kentucky	3,996,000	40,400
Louisiana	4,465,000	52,400	Louisiana	4,465,000	52,400	Louisiana	4,465,000	52,400
Maine	1,275,000	33,000	Maine	1,275,000	33,000	Maine	1,275,000	33,000
Maryland	5,773,000	11,300	Maryland	5,773,000	11,300	Maryland	5,773,000	11,300
Massachusetts	6,547,000	8,000	Massachusetts	6,547,000	8,000	Massachusetts	6,547,000	8,000
Michigan	9,856,000	96,600	Michigan	9,856,000	96,600	Michigan	9,856,000	96,600
Minnesota	3,463,000	225,300	Minnesota	3,463,000	225,300	Minnesota	3,463,000	225,300
Mississippi	2,865,000	47,800	Mississippi	2,865,000	47,800	Mississippi	2,865,000	47,800
Missouri	5,937,000	168,000	Missouri	5,937,000	168,000	Missouri	5,937,000	168,000
Montana	989,000	147,000	Montana	989,000	147,000	Montana	989,000	147,000
Nebraska	1,901,000	167,800	Nebraska	1,901,000	167,800	Nebraska	1,901,000	167,800
Nevada	2,050,000	110,600	Nevada	2,050,000	110,600	Nevada	2,050,000	110,600
New Hampshire	1,198,000	9,300	New Hampshire	1,198,000	9,300	New Hampshire	1,198,000	9,300
New Jersey	8,791,000	14,300	New Jersey	8,791,000	14,300	New Jersey	8,791,000	14,300
New Mexico	1,964,000	121,900	New Mexico	1,964,000	121,900	New Mexico	1,964,000	121,900
New York	19,378,000	47,200	New York	19,378,000	47,200	New York	19,378,000	47,200
North Carolina	9,535,000	51,900	North Carolina	9,535,000	51,900	North Carolina	9,535,000	51,900
North Dakota	714,000	77,000	North Dakota	714,000	77,000	North Dakota	714,000	77,000
Ohio	10,539,000	44,800	Ohio	10,539,000	44,800	Ohio	10,539,000	44,800
Oklahoma	3,756,000	175,000	Oklahoma	3,756,000	175,000	Oklahoma	3,756,000	175,000
Oregon	3,438,000	95,300	Oregon	3,438,000	95,300	Oregon	3,438,000	95,300
Pennsylvania	12,281,000	46,000	Pennsylvania	12,281,000	46,000	Pennsylvania	12,281,000	46,000
Rhode Island	1,059,000	1,500	Rhode Island	1,059,000	1,500	Rhode Island	1,059,000	1,500
South Carolina	4,012,000	29,800	South Carolina	4,012,000	29,800	South Carolina	4,012,000	29,800
South Dakota	779,000	77,000	South Dakota	779,000	77,000	South Dakota	779,000	77,000
Tennessee	5,687,000	42,000	Tennessee	5,687,000	42,000	Tennessee	5,687,000	42,000
Texas	20,505,000	695,600	Texas	20,505,000	695,600	Texas	20,505,000	695,600
Utah	2,763,000	199,600	Utah	2,763,000	199,600	Utah	2,763,000	199,600
Vermont	609,000	9,400	Vermont	609,000	9,400	Vermont	609,000	9,400
Virginia	7,642,000	111,800	Virginia	7,642,000	111,800	Virginia	7,642,000	111,800
Washington	6,081,000	71,300	Washington	6,081,000	71,300	Washington	6,081,000	71,300
West Virginia	1,781,000	62,000	West Virginia	1,781,000	62,000	West Virginia	1,781,000	62,000
Wisconsin	5,421,000	145,500	Wisconsin	5,421,000	145,500	Wisconsin	5,421,000	145,500
Wyoming	563,000	97,800	Wyoming	563,000	97,800	Wyoming	563,000	97,800

(continued on next page)

and

LIST OF STATES

STATE	POPULATION	AREA	POPULATION	AREA	POPULATION	AREA	POPULATION	AREA
Alabama	2,049,000	52,400	Alabama	2,049,000	52,400	Alabama	2,049,000	52,400
Alaska	277,000	588,000	Alaska	277,000	588,000	Alaska	277,000	588,000
Arizona	1,512,000	113,900	Arizona	1,512,000	113,900	Arizona	1,512,000	113,900
Arkansas	1,188,000	53,100	Arkansas	1,188,000	53,100	Arkansas	1,188,000	53,100
California	15,981,000	155,900	California	15,981,000	155,900	California	15,981,000	155,900
Colorado	2,766,000	104,000	Colorado	2,766,000	104,000	Colorado	2,766,000	104,000
Connecticut	3,287,000	5,500	Connecticut	3,287,000	5,500	Connecticut	3,287,000	5,500
Delaware	787,000	2,400	Delaware	787,000	2,400	Delaware	787,000	2,400
Florida	17,642,000	65,700	Florida	17,642,000	65,700	Florida	17,642,000	65,700
Georgia	4,779,000	59,700	Georgia	4,779,000	59,700	Georgia	4,779,000	59,700
Hawaii	1,059,000	10,900	Hawaii	1,059,000	10,900	Hawaii	1,059,000	10,900
Idaho	1,208,000	141,600	Idaho	1,208,000	141,600	Idaho	1,208,000	141,600
Illinois	12,121,000	149,900	Illinois	12,121,000	149,900	Illinois	12,121,000	149,900
Indiana	6,081,000	36,400	Indiana	6,081,000	36,400	Indiana	6,081,000	36,400
Iowa	2,998,000	145,700	Iowa	2,998,000	145,700	Iowa	2,998,000	145,700
Kansas	1,781,000	175,000	Kansas	1,781,000	175,000	Kansas	1,781,000	175,000
Kentucky	3,996,000	40,400	Kentucky	3,996,000	40,400	Kentucky	3,996,000	40,400
Louisiana	4,465,000	52,400	Louisiana	4,465,000	52,400	Louisiana	4,465,000	52,400
Maine	1,275,000	33,000	Maine	1,275,000	33,000	Maine	1,275,000	33,000
Maryland	5,773,000	11,300	Maryland	5,773,000	11,300	Maryland	5,773,000	11,300
Massachusetts	6,547,000	8,000	Massachusetts	6,547,000	8,000	Massachusetts	6,547,000	8,000
Michigan	9,856,000	96,600	Michigan	9,856,000	96,600	Michigan	9,856,000	96,600
Minnesota	3,463,000	225,300	Minnesota	3,463,000	225,300	Minnesota	3,463,000	225,300
Mississippi	2,865,000	47,800	Mississippi	2,865,000	47,800	Mississippi	2,865,000	47,800
Missouri	5,937,000	168,000	Missouri	5,937,000	168,000	Missouri	5,937,000	168,000
Montana	989,000	147,000	Montana	989,000	147,000	Montana	989,000	147,000
Nebraska	1,901,000	167,800	Nebraska	1,901,000	167,800	Nebraska	1,901,000	167,800
Nevada	2,050,000	110,600	Nevada	2,050,000	110,600	Nevada	2,050,000	110,600
New Hampshire	1,198,000	9,300	New Hampshire	1,198,000	9,300	New Hampshire	1,198,000	9,300
New Jersey	8,791,000	14,300	New Jersey	8,791,000	14,300	New Jersey	8,791,000	14,300
New Mexico	1,964,000	121,900	New Mexico	1,964,000	121,900	New Mexico	1,964,000	121,900
New York	19,378,000	47,200	New York	19,378,000	47,200	New York	19,378,000	47,200
North Carolina	9,535,000	51,900	North Carolina	9,535,000	51,900	North Carolina	9,535,000	51,900
North Dakota	714,000	77,000	North Dakota	714,000	77,000	North Dakota	714,000	77,000
Ohio	10,539,000	44,800	Ohio	10,539,000	44,800	Ohio	10,539,000	44,800
Oklahoma	3,756,000	175,000	Oklahoma	3,756,000	175,000	Oklahoma	3,756,000	175,000
Oregon	3,438,000	95,300	Oregon	3,438,000	95,300	Oregon	3,438,000	95,300
Pennsylvania	12,281,000	46,000	Pennsylvania	12,281,000	46,000	Pennsylvania	12,281,000	46,000
Rhode Island	1,059,000	1,500	Rhode Island	1,059,000	1,500	Rhode Island	1,059,000	1,500
South Carolina	4,012,000	29,800	South Carolina	4,012,000	29,800	South Carolina	4,012,000	29,800
South Dakota	779,000	77,000	South Dakota	779,000	77,000	South Dakota	779,000	77,000
Tennessee	5,687,000	42,000	Tennessee	5,687,000	42,000	Tennessee	5,687,000	42,000
Texas	20,505,000	695,600	Texas	20,505,000	695,600	Texas	20,505,000	695,600
Utah	2,763,000	199,600	Utah	2,763,000	199,600	Utah	2,763,000	199,600
Vermont	609,000	9,400	Vermont	609,000	9,400	Vermont	609,000	9,400
Virginia	7,642,000	111,800	Virginia	7,642,000	111,800	Virginia	7,642,000	111,800
Washington	6,081,000	71,300	Washington	6,081,000	71,300	Washington	6,081,000	71,300
West Virginia	1,781,000	62,000	West Virginia	1,781,000	62,000	West Virginia	1,781,000	62,000
Wisconsin	5,421,000	145,500	Wisconsin	5,421,000	145,500	Wisconsin	5,421,000	145,500
Wyoming	563,000	97,800	Wyoming	563,000	97,800	Wyoming	563,000	97,800