INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)

Dundigal, Hyderabad - 500 043

COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

Course Title	PROGRAMMING FOR PROBLEM SOLVING USING C				
Course Code	ACSC04				
Program	B.Tech				
Semester	II				
Course Type	FOUNDATION				
Regulation	UG-20				
Course Structure	Theory			Practical	
	Lecture	Tutorials	Credits	Laboratory	Credits
	3	-	3	-	-
Course Coordinator	Dr. Shaik Jakeer Hussain, Associate Professor				

COURSE OBJECTIVES:

The students will try to learn:

I	Problem-solving through programming.
II	Programming language, programming, reading a set of Data, stepwise refinement, concepts of Loops, Functions, Control structure, Arrays, Structure, Pointer and File concept.
III	To build efficient programs in C language essential for future programming and software engineering courses.
IV	Acquire programming skills in C Programming.

COURSE OUTCOMES:

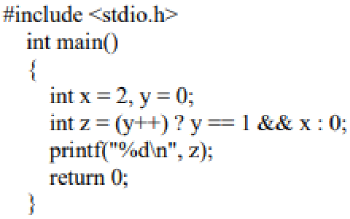
After successful completion of the course, students should be able to:

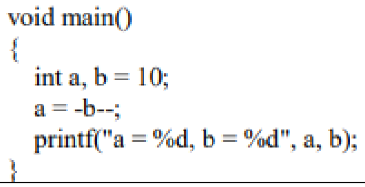
CO 1	**Define** the algorithms and draw flowcharts for solving Mathematical and Engineering problems.	Remember
CO 2	**Construct** programs for decision structures and loops.	Apply
CO 3	**Interpret** various types of functions, arrays, and strings for complex problem solving.	Understand
CO 4	**Illustrate** he dynamic memory allocation, structures, unions and enumerations to solve problems.	Understand

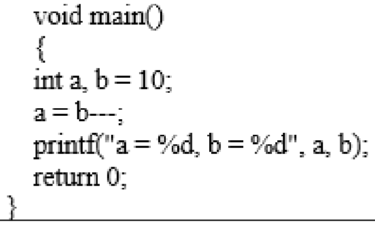
CO 5	**Interpret** file input and output functions to do integrated programming.	Understand
CO 6	**Utilize** the algorithms in C language to real-life computational problems.	Apply

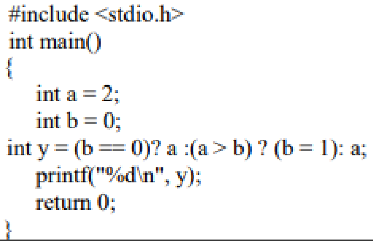
QUESTION BANK:

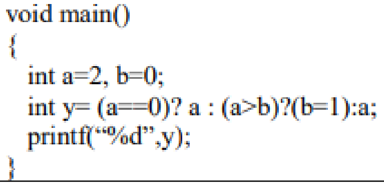
Q.No	QUESTION	Taxonomy	How does this subsume the level	CO's
MODULE I				
INTRODUCTION				
PART A-PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS				

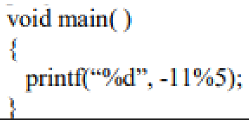
1	Find the output of the following program.
	
	#include <stdio.h>
	int main()
	{
	int x = 2, y = 0;
	int z = (y++) ? y == 1 && x : 0;
	printf("%d\n", z);
	return 0;
	}
		Understand	The learner toRecall types of operators in C and demonstratethe use of logical, conditional operators.	CO 1

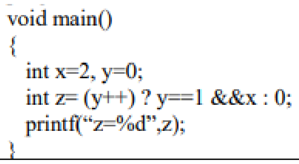
2	Find the output of the following program.
	
	void main()
	{
	int a, b = 10;
	a = -b--;
	printf("a = %d, b = %d", a, b);
	}
		Understand	The learner to Recall types of operators in C and demonstrate the use of unary operators.	CO 1

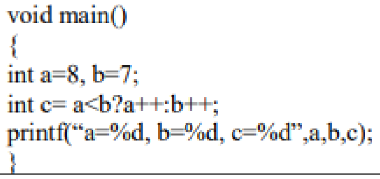
3	Find the output of the following program.
	
	void main()
	{
	int a, b = 10;
	a = b---;
	printf("a = %d, b = %d", a, b);
	return 0;
	}
		Understand	The learner to Recall types of operators in C and demonstrate the use of unary operators.	CO 1

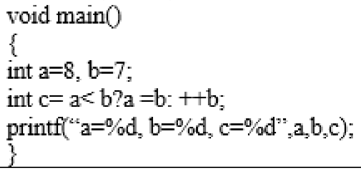
4	Find the output of the following program.
	
	#include <stdio.h>
	int main()
	{
	int a = 2;
	int b = 0;
	int y = (b == 0)? a :(a > b) ? (b = 1): a;
	printf("%d\n", y);
	return 0;
	}
		Unserstand	The learner to Recall types of operators in C and demonstrate the use of conditional, comparison operators.	CO 1

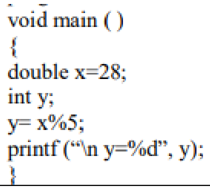
5	Find the output of the following program.
	
	void main()
	{
	  int a=2, b=0;
	  int y= (a==0)? a : (a>b)?(b=1):a;
	  printf(“%d”,y);
	}
		Understand	The learner to Recall types of operators in C and demonstrate the use of conditional operators.	CO 1

6	Find the output of the following program.
	
	void main( )
	{
	  printf(“%d”, -11%5);
	}
		Understand	The learner to Recall types of operators in C and demonstrate the use of arithmetic operators.	CO 1

7	Find the output of the following program.
	
	void main()
	{
	  int x=2, y=0;
	  int z= (y++) ? y==1 &&x : 0;
	  printf(“z=%d”,z);
	}
		Understand	The learner to Recall types of operators in C and demonstrate the use of unary, logical, conditional operators.	CO 1

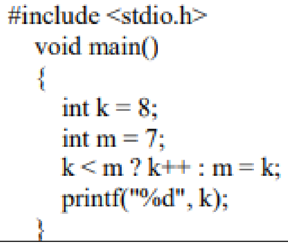
8	Find the output of the following program.
	
	void main()
	{
	int a=8, b=7;
	int c= a<b?a++:b++;
	printf(“a=%d, b=%d, c=%d”,a,b,c);
	}
		Understand	The learner to Recall types of operators in C and demonstrate the use of unary, comparison, conditional operators.	CO 1

9	Find the output of the following program.
	
	void main()
	{
	int a=8, b=7;
	int c= a< b?a =b: ++b;
	printf(“a=%d, b=%d, c=%d”,a,b,c);
	}
		Understand	The learner to Recall types of operators in C and Demonstrate the use of unary, comparison, conditional operators.	CO 1

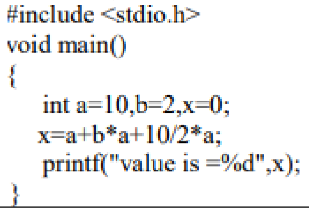
10	Find the output of the following program.
	
	void main ( )
	{
	double x=28;
	int y;
	y= x%5;
	printf (“\n y=%d”, y);
	}
		Understand	The learner to Recall types of operators in C and demonstrate the use of arithmetic operators.	CO 1

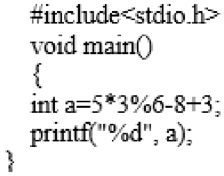
PART-B LONG ANSWER QUESTIONS				
1	Explain in detail about computer hardware and software.	Remember	–	CO 1

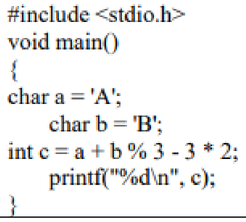
2	Differentiate among high level, low level and middle level language.	Remember	-	CO 1
3	Differentiate among compiler, assembler and interpreter	Remember	-	CO 1
4	Define flowchart and explain different symbols used for constructing flowchart.	Remember	-	CO 1
5	Explain structure of a C program with example.	Remember	-	CO 1
6	Explain all the data types with their ranges, examples.	Remember	-	CO 1
7	Explain Process of compiling and running a C program.	Remember	-	CO 1
8	What is variable? Give the rules for variable declaration.	Remember	-	CO 1
9	Explain syntax with examples of printf() and scanf() statements.	Remember	-	CO 1
10	Explain in detail about the types of operators in C	Remember	-	CO 1
11	Explain in detail about operator precedence and associativity in C.	Remember	-	CO 1
12	Explain Type Conversion and type casting in C.	Remember	-	CO 1

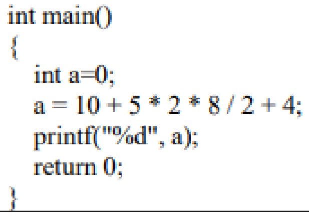
13	Find the output of the following code.
	
	#include <stdio.h>
	void main()
	{
	int k = 8;
	int m = 7;
	k < m ? k++ : m = k;
	printf("%d", k);
	}
		Understand	Learner to recall the types of operators and demonstrate the use of conditional, unary operators.	CO 1

14	Evaluate the following expressions: 1. a+=b*=c-=5 where a=3, b=5, c=8 2. int a,b; float x; a=4; b=5; x=b/a ; 3. int a,b; float x; a=4; b=5; x=(float)b/a;	Understand	Learner to recall the types of operators and demonstrate the use of types of operators in evaluating expressions.	CO 1

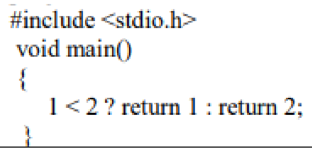
15	Find the output of the following code.
	
	#include <stdio.h>
	void main()
	{
	int a=10,b=2,x=0;
	x=a+b*a+10/2*a;
	printf("value is =%d",x);
	}
		Understand	Learner to recall the types of operators and demonstrate the use of types of operators precedence in evaluating expressions.	CO 1

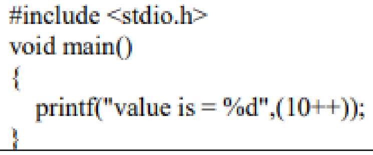
16	Find the output of the following code.
	
	#include<stdio.h>
	void main()
	{
	int a=5*3%6-8+3;
	printf("%d", a);
	}
		Understand	Learner to recall the types of operators and demonstrate the use of types of operator precedence in evaluating expressions.	CO 1

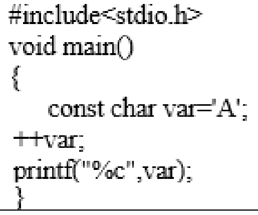
17	Find the output of the following code.
	
	#include <stdio.h>
	void main()
	{
	char a = 'A';
	char b = 'B';
	int c = a + b % 3 - 3 * 2;
	printf("%d\n", c);
	}
		Understand	Learner to recall the types of operators and demonstrate the use of types of operator precedence in evaluating expressions.	CO 1

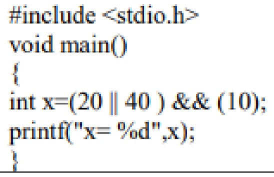
18	Find the output of the following code.
	
	int main()
	{
	int a=0;
	a = 10 + 5 * 2 * 8 / 2 + 4;
	printf("%d", a);
	return 0;
	}
		Understand	Learner to recall the types of operators and demonstrate the use of types of operator precedence in evaluating expressions.	CO 1

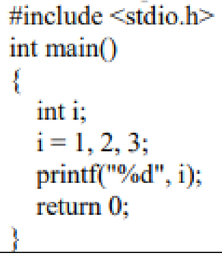
19	Evaluate the following expressions: 1. x = a-b/3 + c*2 - 1 when a = 9, b= 12 & c = 13 2. 10 != 10 —— 5 ¡ 4 && 8 3. Evaluate the z=5%3/8*3+4	Understand	Learner to recall the types of operators and demonstrate the use of types of operator precedence in evaluating expressions.	CO 1
20	iEvaluate the following expression 6*2/ (2+1 * 2/3 + 6) + 8 * (8/4)	Understand	Learner to recall the types of operators and demonstrate the use of types of operator precedence in evaluating expressions.	CO 1
PART-C SHORT ANSWER QUESTIONS				
1	List the major components of computer.	Remember	-	CO 1
2	Define the term operating system.	Remember	-	CO 1
3	Define the term algorithm?	Remember	-	CO 1
4	Define the term flowchart?	Remembe	-	CO 1
5	Write the properties of an algorithm.	Remembe	-	CO 1
6	Write how a compiler works.	Remember	-	CO 1
7	Compare the differences between compiler and an interpreter.	Remember	-	CO 1
8	Write about datatypes in C.	Remember	-	CO 1
9	List the rules for naming identifiers in C.	Remember	-	CO 1
10	List the types of operators in C.	Remember	-	CO 1
11	Explain operator precedence in C.	Remember	-	CO 1
12	Compare & and * operators in C.	Remember	-	CO 1

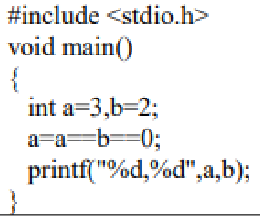
13	Find the output of the following program.
	
	#include <stdio.h>
	void main()
	{
	1 < 2 ? return 1 : return 2;
	}
		Understand	Learner to recall the types of operators and demonstrate the use of comparison operators.	CO 1

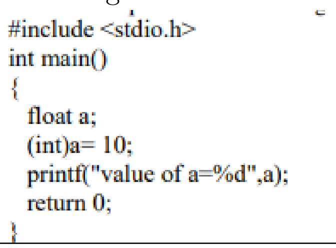
14	Find the output of the following program.
	
	#include <stdio.h>
	void main()
	{
	printf("value is = %d",(10++));
	}
		Understand	Learner to recall the types of operators and demonstrate the use of unary operators	CO 1

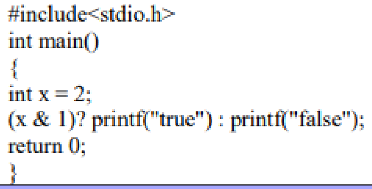
15	Find the output of the following program.
	
	#include<stdio.h>
	void main()
	{
	const char var='A';
	++var;
	printf("%c",var);
	}
		Understand	Learner to recall the types of operators and demonstrate the use of unary operators	CO 1

16	Find the output of the following code.
	
	#include <stdio.h>
	void main()
	{
	int x=(20 || 40) && (10);
	printf("x= %d",x);
	}
		Understand	Learner to recall the types of operators and demonstrate the use of logical operators	CO 1

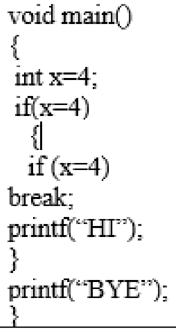
17	Find the output of the following code.
	
	#include <stdio.h>
	int main()
	{
	int i;
	i = 1, 2, 3;
	printf("%d", i);
	return 0;
	}
		Understand	Learner to recall the types of operators and demonstrate the use of assignment operators	CO 1

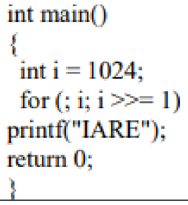
18	Find the output of the following code.
	
	#include <stdio.h>
	void main()
	{
	int a=3,b=2;
	a=a==b==0;
	printf("%d,%d",a,b);
	}
		Understand	Learner to recall the types of operators and demonstrate the use of assignment operators	CO 1

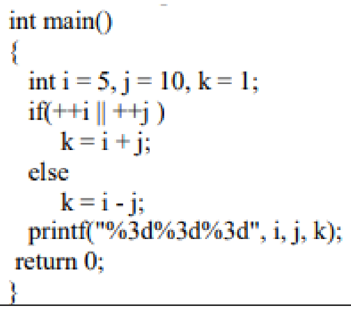
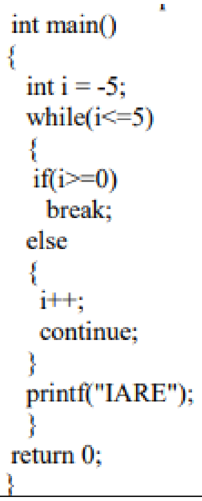
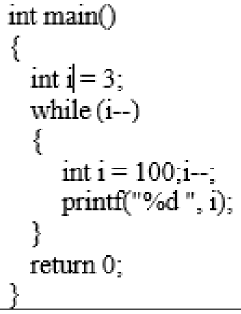
19	Find the output of the following code.
	
	#include <stdio.h>
	int main()
	{
	  float a;
	  (int)a= 10;
	  printf("value of a=%d",a);
	  return 0;
	}
		Understand	Learner to recall the types of operators and demonstrate the use of assignment operators	CO 1

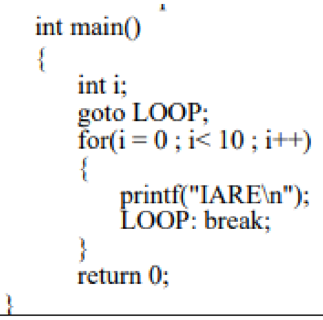
20	Find the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	int x = 2;
	(x & 1)? printf("true") : printf("false");
	return 0;
	}
		Understand	Learner to recall the types of operators and demonstrate the use of conditional operators	CO 1

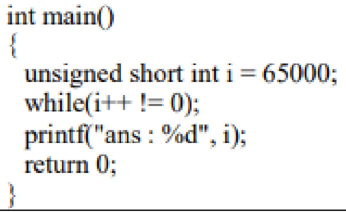
MODULE II				
CONTROL STRUCTURES				
PART-A PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS				

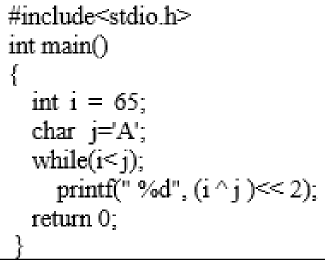
1	Predict the output of the following code
	
	void main()
	{
	 int x=4;
	 if(x=4)
	  {
	  if (x=4)
	break;
	printf(“HI”);
	}
	printf(“BYE”);
	}
		Understand	Learner to recall decision control statements in C and demonstrate these to predict the output.	CO 2

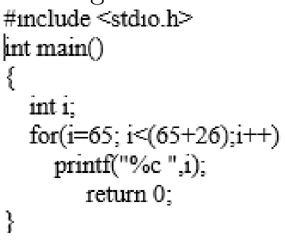
2	Predict the output of the following code.
	
	int main()
	{
	 int i = 1024;
	 for (; i; i >>= 1)
	printf("IARE");
	return 0;
	}
		Understand	Learner to recallloop control, decision control statements in C and demonstrate these to predict the output.	CO 2

<table>
<tr><td>3</td><td>Predict the output of the following code.

int main()
{
int i = 5, j = 10, k = 1;
if(++i || ++j)
k = i + j;
else
k = i - j;
printf("%3d%3d%3d", i, j, k);
return 0;
}</td><td>Understand</td><td>Learner to recalldecision control statements in C and demonstrate these to predict the output.</td><td>CO 2</td></tr>
<tr><td>4</td><td>Predict the output of the following code.

int main()
{
int i = -5;
while(i<=5)
{
if(i>=0)
break;
else
{
i++;
continue;
}
printf("IARE");
}
return 0;
}</td><td>Understand</td><td>Learner to recallloop control, decision control statements in C and demonstrate these to predict the output.</td><td>CO 2</td></tr>
<tr><td>5</td><td>Predict the output of the following code.

int main()
{
int i = 3;
while (i--)
{
int i = 100;i--;
printf("%d ", i);
}
return 0;
}</td><td>Understand</td><td>Learner to recall loop control statements in C and demonstrate the use of loops to predict the output.</td><td>CO 2</td></tr>
</table>

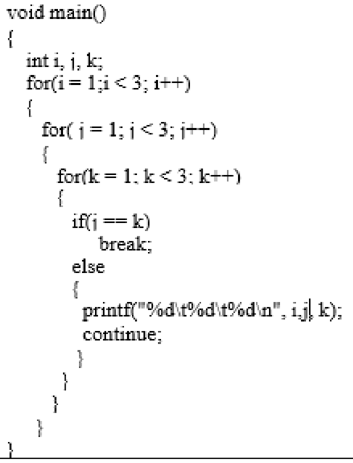
6	Predict the output of the following code.
	
	int main()
	{
	int i;
	goto LOOP;
	for(i = 0 ; i< 10 ; i++)
	{
	printf("IARE\n");
	LOOP: break;
	}
	return 0;
	}
		Understand	Learner to recall loop control statements in C and demonstrate the use of nested loops to predict the output.	CO 2

7	Predict the output of the following code.
	
	int main()
	{
	unsigned short int i = 65000;
	while(i++ != 0);
	printf("ans : %d", i);
	return 0;
	}
		Understand	Learner to recall loop control statements in C and demonstrate the use of loops to predict the output.	CO 2

8	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	int i = 65;
	char j='A';
	while(i<j);
	printf(" %d", (i ^ j)<< 2);
	return 0;
	}
		Understand	This would require the learner to recall micro-operation then explain the one stage logic circuit.	CO 2

9	Predict the output of the following code
	
	#include <stdio.h>
	int main()
	{
	int i;
	for(i=65; i<(65+26);i++)
	printf("%c ",i);
	return 0;
	}
		Understand	Learner to recall loop control statements in C and demonstrate the use of loops to predict the output.	CO 2

<table>
<tr><td>10</td><td>Predict the output of the following code.



```
void main()
{
   int i, j, k;
   for(i = 1;i < 3; i++)
   {
     for( j = 1; j < 3; j++)
     {
       for(k = 1; k < 3; k++)
       {
         if(j == k)
              break;
         else
         {
           printf("%d\t%d\t%d\n", i,j, k);
           continue;
         }
       }
     }
   }
}
```

</td><td>Understand</td><td>Learner to recall loop control statements in C and demonstrate the use of nested loops to predict the output.</td><td>CO 2</td></tr>
<tr><td colspan="5">PART-B LONG ANSWER QUESTIONS</td></tr>
<tr><td>1</td><td>Write a program in C to display the n terms of square natural number and their sum.1 4 9 16 ... n Terms Sample Input/ Output:
Input the number of terms : 5
Expected Output :
The square natural upto 5 terms are :1 4 9 16 25
The Sum of Square Natural Number
upto 5 terms = 55</td><td>Understand</td><td>Learner to recalltypes of loops in C and demonstrate the use of loops.Use these concepts to print terms of square natural number and their sum.</td><td>CO 2</td></tr>
<tr><td>2</td><td>Write a program in C to find the prime numbers within a range of numbers.
Sample Input/ Output:
Input starting number of range: 1
Input ending number of range : 50
Expected Output :
The prime number between 1 and 50 are :2 3 5 7 11 13 17 19 23 29 31 37 41 43 47</td><td>Understand</td><td>Learner to recalltypes of loops in C and demonstrate the use of nested loops. Use these concepts to find the prime numbers within a range of numbers.</td><td>CO 2</td></tr>
</table>

3	Write a C program to display the traffic control signal lights based on the following. i. If user entered character is R or r then print RED Light Please STOP. ii. If user entered character is Y or y then print YELLOW Light Please Check and Go. iii. If user entered character is G or g then print GREEN Light Please GO. iv. If user entered some other character then print THERE IS NOSIGNAL POINT	Understand	Learner to recalldecision making statements in C and demonstrate the use of else if ladder to display the traffic control signal lights.	CO 2
4	Admission to a professional course is subject to the following conditions: i. Marks in Mathematics ¿=60 ii. Marks in Physics ¿= 50 Marks in Chemistry ¿=40 iii. Total in all three subjects ¿=200 iv. Total in Mathematics and Physics ¿=150 Given the marks in the three subjects, Write a C program to process the application to list the eligible candidates.	Understand	Learner to recalldecision making statements in C and demonstrate the use of else if ladder to list the eligible candidates.	CO 2

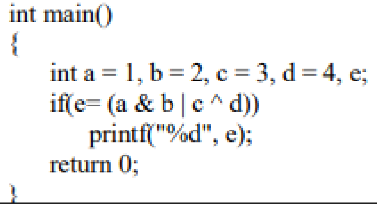
5	Write a C program to compute the real roots of a quadratic equation ax2 + bx+ c = 0. The program should request for the values of the constants a, b and c and print the values of x1 and x2. Use the following rules: i. No solution, if both a and b are zero There is only one root, ifa=0 ii. There are no real roots, if b2 - 4ac is negative Otherwise, there are two real roots Write a C program to test all the above conditions.	Understand	Learner to recalldecision making statements in C and demonstrate the use of if else statements to compute the real roots of a quadratic equation.	CO 2
6	Write a program that counts from one to ten, prints the values on a separate line for each, and includes a message of your choice when the count is 3 and a different message when the count is 7.	Understand	Learner to recalldecision making statements, loop control statements in C and demonstrate the use of if, for statements to print the desired output.	CO 2
7	Write a C program to calculate commission for the input value of sales amount. Commission is calculated as per the following rules: i. Commission is nil for sales amount Rs5000/. ii. Commission is 2% for sales when sales amount is greater than 5000and less than equal to10000. iii. Commission is 5% for sales amount greater than10000.	Understand	Learner to recalldecision making statements in C and demonstrate the use of else if ladder to calculate commission.	CO 2

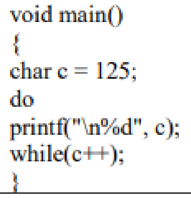
8	A character is entered through keyboard. Write a C program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol using if-else and switch case. The following table shows the range of ASCII values for various characters. Characters ASCII values A–Z 65 –90 a–z 97 –122 0–9 48 – 57 Special symbols 0 – 47, 58 – 64, 91 – 96, 123 -127	Understand	Learner to recalldecision making statements in C and demonstrate the use of if-else and switch caseto determine type character entered.	CO 2
9	If cost price and selling price of an item S input through the keyboard, write a program to determine whether the seller has made profit or incurred loss. Write a C program to determine how much profit or loss incurred in percentage.	Understand	Learner to recalldecision making statements in C and demonstrate the use of if else statements to determine how much profit or loss.	CO 2
10	Write a C program to print the following pattern. 1 3 5 7 9 11 13 15 17 19	Understand	Learner to recallloop control statements in C and demonstrate the use of nested loops to print the pattern.	CO 2
11	Write a C program to print the following pattern. 1 12 123 1234	Understand	Learner to recallloop control statements in C and demonstrate the use of nested loops to print the pattern.	CO 2

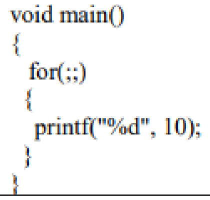
12	Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression1+x+x2+x3+...xn. For example: if n is 3 and x is 5, then the program computes 1+5+25+125. Print x, n, the sum. Perform error checking. For example the formula does not make sense for negative Exponents – if n is less than 0. Have your program print an error message if n¡0,then go back and read in then pair of numbers of without computing the sum. Are any values of x also illegal? If so, test for them too	Understand	Learner to recallloop control statements in C and demonstrate the use of nested loops to find thesum of geometric progression.	CO 2
13	Write a C program to print Armstrong numbers between 1 to n where n value is entered by the user. [Hint: Armstrong number is defined as the sum of cubes of individual digits of a number. e.g. 371 = 33 + 73 + 13]	Understand	Learner to recallloop control statements in C and demonstrate the use of nested loops to print Armstrong numbers between 1 to n.	CO 2
14	Write a C program to generate all prime numbers between 1 and n, where n value is supplied by the user.	Understand	Learner to recallloop control statements in C and demonstrate the use of nested loops to print prime numbers between 1 to n.	CO 2
15	Write a C program to print first n lines of Floyd's Triangle. 1 2 3 4 5 6 7 8 9 10	Understand	Learner to recallloop control statements in C and demonstrate the use of nested loops to print the pattern.	CO 2
16	Write a C program to print the following series 1/1! + 2/2! + 3/3! + ...	Understand	Learner to recallloop control statements in C and demonstrate the use of nested loops to print the series	CO 2

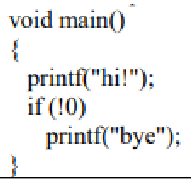
17	Write a C program to compute and display the sum of all integers that are divisible by 6 but not divisible by 4 and lie between 0 and 100. The program should also count and display the number of such values.	Understand	Learner to recallloop control statements in C and demonstrate the use of nested loops to compute and display the sum of all integers.	CO 2
18	Write a C program to find the LCM and GCD of two integers.	Apply	Learner to recallloop control statements in C and demonstrate the use of loops to display the n terms of odd natural number and their sum.	CO 2
19	Write a program in C to display the n terms of odd natural number and their sum. Sample Input/ Output: Input number of terms : 10 Expected Output : The odd numbers are :1 3 5 7 9 11 13 15 17 19 The Sum of odd Natural Number upto 10 terms : 100	Understand	Learner to recallloop control statements in C and demonstrate the use of loops to find the LCM and GCD of two integers.	CO 2
20	Write a C program to print the following pattern. 1 22 333 4444	Understand	Learner to recallloop control statements in C and demonstrate the use of nested loops to print the pattern.	CO 2
PART-C SHORT ANSWER QUESTIONS				
1	List the control structures in C.	Remember	-	CO 2
2	List the decision making statements in C	Remember	-	CO 2
3	List the loop control statements in C.	Remember	-	CO 2
4	Explain continue statement in C?	Remember	-	CO 2
5	Explain goto statement in C	Remember	-	CO 2
6	Explain break statement in C.	Remember	-	CO 2

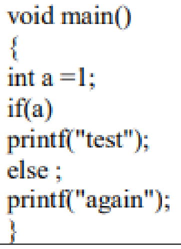
7	Compare the difference between entry controlled and exit controlled statements.	Remember	-	CO 2
8	Explain switch statement in C.	Remember	-	CO 2

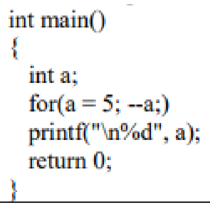
9	Find the output of the following code.
	
	`int main()`
	`{`
	`int a = 1, b = 2, c = 3, d = 4, e;`
	`if(e= (a & b | c ^ d))`
	`printf("%d", e);`
	`return 0;`
	`}`
		Understand	Learner to recall the types of operators and demonstrate the use of operator precedence in evaluating expressions.	CO 2

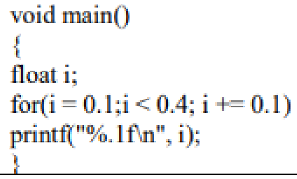
10	Find the output of the following code.
	
	`void main()`
	`{`
	`char c = 125;`
	`do`
	`printf("\n%d", c);`
	`while(c++);`
	`}`
		Understand	Learner to recalltypes of loops in C and demonstrate the use of while loop in finding output.	CO 2

11	Find the output of the following code.
	
	`void main()`
	`{`
	`for(;;)`
	`{`
	`printf("%d", 10);`
	`}`
	`}`
		Understand	Learner to recalltypes of loops in C and demonstrate the use of for loop in finding output.	CO 2

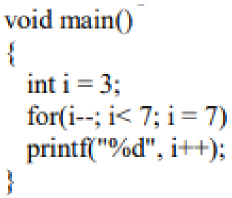
12	Find the output of the following code.
	
	`void main()`
	`{`
	`printf("hi!");`
	`if (!0)`
	`printf("bye");`
	`}`
		Understand	Learner to recalldecision making statements in C and demonstrate the use of if statement in finding output.	CO 2

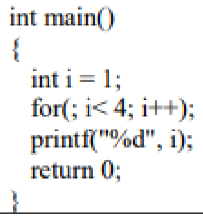
13	Find the output of the following code.
	
	`void main()`
	`{`
	`int a =1;`
	`if(a)`
	`printf("test");`
	`else ;`
	`printf("again");`
	`}`
		Understand	Learner to recalldecision making statements in C and demonstrate the use of if else statement in finding output.	CO 2

14	Find the output of the following code.
	
	int main()
	{
	int a;
	for(a = 5; --a;)
	printf("\n%d", a);
	return 0;
	}
		Understand	Learner to recalltypes of loops in C and demonstrate the use of for loop in finding output.	CO 2

15	Find the output of the following code.
	
	void main()
	{
	float i;
	for(i = 0.1;i < 0.4; i += 0.1)
	printf("%.1f\n", i);
	}
		Understand	Learner to recalltypes of loops in C and demonstrate the use of for loop in finding output.	CO 2

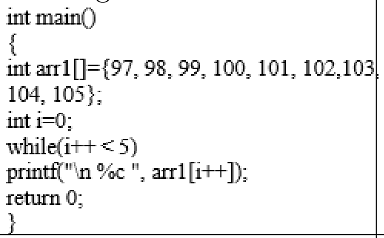
16	Explain switch case execution process with and without break statement.	Remember	-	CO 2

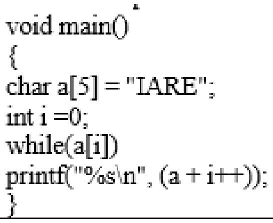
17	Find the output of the following code.
	
	void main()
	{
	int i = 3;
	for(i--; i< 7; i = 7)
	printf("%d", i++);
	}
		Understand	Learner to recalltypes of loops in C and demonstrate the use of for loop in finding output.	CO 2

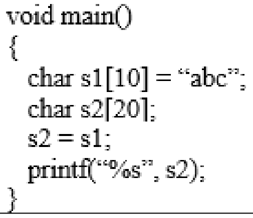
18	Find the output of the following code.
	
	int main()
	{
	int i = 1;
	for(; i< 4; i++);
	printf("%d", i);
	return 0;
	}
		Understand	Learner to recalltypes of loops in C and demonstrate the use of for loop in finding output.	CO 2

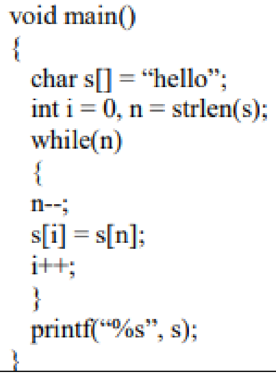
19	Write a program in C to display the n terms of odd natural number and their sum. Sample Input/ Output Input number of terms : 10 Expected Output : The odd numbers are :1 3 5 7 9 11 13 15 17 19 The Sum of odd Natural Number upto 10 terms : 100.	Understand	Learner to recalltypes of loops in C and demonstrate the use of loops.Use these concepts to print n terms of odd natural number and their sum.	CO 2

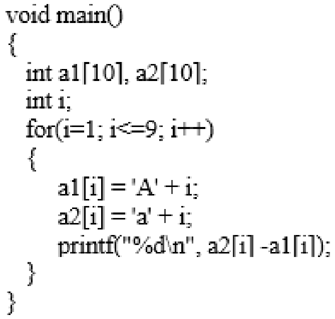
20	Write a program in C to display the multiplication table of a given integer. Sample Input/ Output Input the number (Table to be calculated) : 15 Expected Output : 15 X 1 = 15 ... ... 15 X 10 = 150	Understand	Learner to recalltypes of loops in C and demonstrate the use of loops.Use these concepts to print multiplication table.	CO 2
MODULE III				
ARRAYS AND FUNCTIONS				
PART A-PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS				

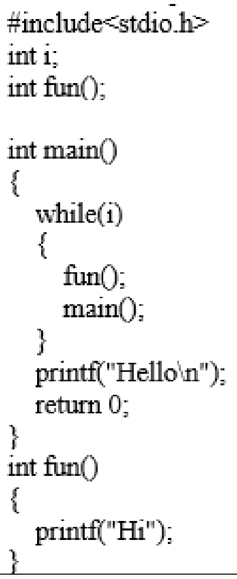
1	Predict the output of the following code?
	
	int main()
	{
	int arr1[]={97, 98, 99, 100, 101, 102,103,
	104, 105};
	int i=0;
	while(i++ < 5)
	printf("\n %c ", arr1[i++]);
	return 0;
	}
		Applyr	Learner to recallarrays and demonstrate the use of arrays, loops in predicting the output.	CO 3

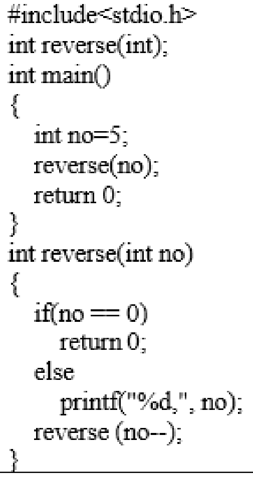
2	Find the output of the following code.
	
	void main()
	{
	char a[5] = "IARE";
	int i =0;
	while(a[i])
	printf("%s\n", (a + i++));
	}
		Apply	Learner to recallstrings and demonstrate the use of strings, loops in predicting the output.	CO 3

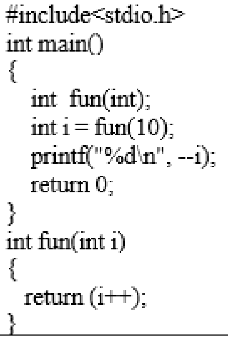
3	Find the output of the following code.
	
	void main()
	{
	char s1[10] = “abc”;
	char s2[20];
	s2 = s1;
	printf(“%s”, s2);
	}
		Understand	Learner to recallstrings and demonstrate the use of strings operations in predicting the output	CO 3

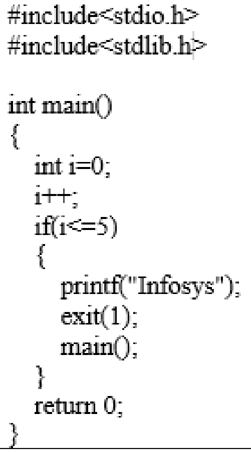
4	EFind the output of the following code.
	
	void main()
	{
	char s[] = “hello”;
	int i = 0, n = strlen(s);
	while(n)
	{
	n--;
	s[i] = s[n];
	i++;
	}
	printf(“%s”, s);
	}
		Apply	Learner to recallstrings and demonstrate the use of strings, loops in predicting the output.	CO 3

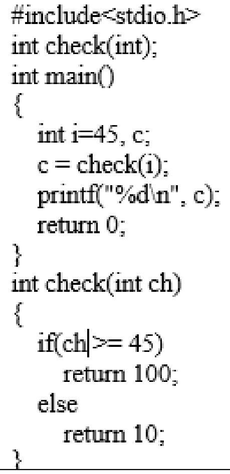
5	Predict the output of the following code.
	
	void main()
	{
	int a1[10], a2[10];
	int i;
	for(i=1; i<=9; i++)
	{
	a1[i] = 'A' + i;
	a2[i] = 'a' + i;
	printf("%d\n", a2[i] -a1[i]);
	}
	}
		Apply	Learner to recallarrays and demonstrate the use of arrays, loops in predicting the output.	CO 3

6	Predict the output of the following code.
	
	#include<stdio.h>
	int i;
	int fun();
	
	int main()
	{
	while(i)
	{
	fun();
	main();
	}
	printf("Hello\n");
	return 0;
	}
	int fun()
	{
	printf("Hi");
	}
		Apply	This would require the learner to recall floating points then solve problem using subtraction	CO 3

7	Predict the output of the following code.
	
	#include<stdio.h>
	int reverse(int);
	int main()
	{
	int no=5;
	reverse(no);
	return 0;
	}
	int reverse(int no)
	{
	if(no == 0)
	return 0;
	else
	printf("%d,", no);
	reverse (no--);
	}
		Apply	This would require the learner to recall arithmetic operations then explain the different arithmetic micro-operations	CO 3

8	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	int fun(int);
	int i = fun(10);
	printf("%d\n", --i);
	return 0;
	}
	int fun(int i)
	{
	return (i++);
	}
		Apply	Learner to recallfunctions and demonstrate the use of function calling methods to predict the output.	CO 3

9	EPredict the output of the following code.
	
	#include<stdio.h>
	#include<stdlib.h>
	
	int main()
	{
	int i=0;
	i++;
	if(i<=5)
	{
	printf("Infosys");
	exit(1);
	main();
	}
	return 0;
	}
		Apply	Learner to recallfunctions and demonstrate the use of function calling methods to predict the output.	CO 3

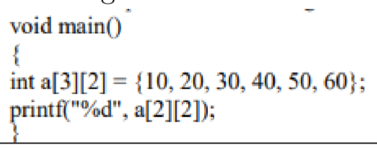
10	Predict the output of the following code.
	
	#include<stdio.h>
	int check(int);
	int main()
	{
	int i=45, c;
	c = check(i);
	printf("%d\n", c);
	return 0;
	}
	int check(int ch)
	{
	if(ch|>= 45)
	return 100;
	else
	return 10;
	}
		Apply	Learner to recallfunctions and demonstrate the use of function calling methods to predict the output.	CO 3

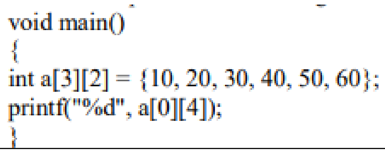
PART-B LONG ANSWER QUESTIONS				
1	Write C program to find the sum of given list of integers.	Understand	Learner to recallarrays and demonstrate the use of arrays, loops to find the sum of given list of integers.	CO 3
2	Write C program to find the largest and smallest number among a list of integers.	Understand	Learner to recallarrays and demonstrate the use of arrays, loops to find the largest and smallest number.	CO 3
3	Write C program to read a list of elements into an array and print the reverse of the list.	Apply	Learner to recallarrays and demonstrate the use of arrays, loops to find the largest and smallest number.	CO 3
4	Write C program to read two matrices and find the addition and multiplication of two matrices.	Apply	Learner to recallarrays and demonstrate the use of two dimensional arrays, nested loops to find the addition and multiplication of two matrices.	CO 3

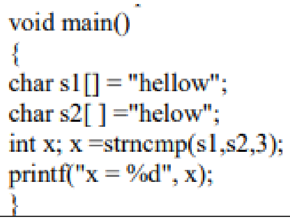
5	Write C programto find the transpose of a matrix. Example Given matrix 1 2 3 4 5 6 Transpose of the matrix: 1 4 2 5 3 6	Apply	Learner to recallarrays and demonstrate the use of two dimensional arrays, nested loops to find the transpose of matrix.	CO 3
6	Write a C program to store numbers into an array and find the frequency of a particular number in array and print it.	Apply	Learner to recallarrays and demonstrate the use of arrays, loops to find the frequency of a particular number in array and print it.	CO 3
7	Write a C program to copy the string str2 into str1 without using strcpy() function.	Apply	Learner to recallstrings and demonstrate the use of strings, loops to copy the string.	CO 3
8	Write a C program to check whether a string is palindrome or not without using string function.	Understand	Learner to recallstrings and demonstrate the use of strings, loops to check whether a string is palindrome or not.	CO 3
9	Write a C program to read your email id and print the number of vowels, consonants and special characters in it.	Understand	Learner to recallstrings and demonstrate the use of strings, loops to count characters in given string.	CO 3
10	Write a C program to insert a sub-string in to given main string at a given position without using string functions.	Understand	Learner to recallstrings and demonstrate the use of strings, loops to insert a sub-string	CO 3
11	Write C program that uses both recursive and non-recursive functions to find the sum of n naturalnumbers.	Understand	Learner to recallfunctions and demonstrate the use of recursive and non-recursive calling methods to find the sum of n naturalnumbers.	CO 3

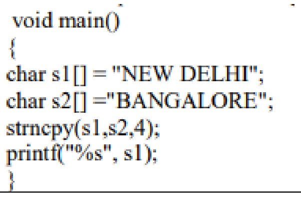
12	Write a C program that uses functions to convert decimal number to binarynumber.	Understand	Learner to recallfunctions and demonstrate the use of function calling methods to convert decimal number to binarynumber.	CO 3
13	Write C program that uses functions tofind the Nth Fibonaccinumber.	Understand	Learner to recallfunctions and demonstrate the use of function calling methods tofind the Nth Fibonaccinumber.	CO 3
14	Explain call by value and call by reference with example.	Remember	–	CO 3
15	Write a user defined function which takes an array of sorted integers and returns the value? [Hint: For odd set of integers there will be a single median and for even set of integers, there will be two middle values and medianis the average of the two middle values]0	Understand	Learner to recallfunctions and demonstrate the use of function, arrays tofind median value.	CO 3
16	Write C program that uses both recursive and non-recursive functions to find the factorial of a givennumber.	Understand	Learner to recallfunctions and demonstrate the use of recursive and non-recursive calling methods to find factorial of a givennumber.	CO 3
17	Write a C program that uses functions to convert binary number to decimalnumber.	Understand	Learner to recallfunctions and demonstrate the use of function calling methods to binary number to decimalnumber	CO 3
18	Write a C program that uses functions to find 2”s complement of a binarynumber.	Understandy	Learner to recallfunctions and demonstrate the use of function calling methods to 2”s complement of a binarynumber.	CO 3

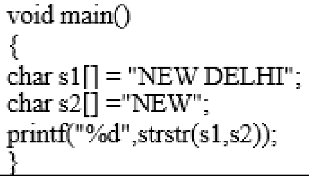
19	Write a program in C to find the square of any number using the function. Example : Input any number for square : 20 Expected Output : The square of 20 is : 400.00	Understand	CO 3	
20	Write a program in C to check whether a number is a prime number or not using the function. Example : Input a positive number : 5 Expected Output : The number 5 is a prime number.	Understand	Learner to recallfunctions and demonstrate the use of recursive and non-recursive calling methods to check a number is a prime number or not.	CO 3
PART-C SHORT ANSWER QUESTIONS				
1	Define what an array is and write the syntax to declare an arraycle.	Remember	-	CO 3

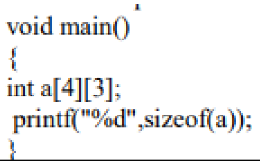
2	Find the output of the following code.
	
	void main()
	{
	int a[3][2] = {10, 20, 30, 40, 50, 60};
	printf("%d", a[2][2]);
	}
		Understand	Learner to recall arrays and demonstrate the use of two dimensional arrays in finding the output.	CO 3

3	Find the output of the following code.
	
	void main()
	{
	int a[3][2] = {10, 20, 30, 40, 50, 60};
	printf("%d", a[0][4]);
	}
		Understand	Learner to recallarrays and demonstrate the use of two dimensional arrays in finding the output.	CO 3

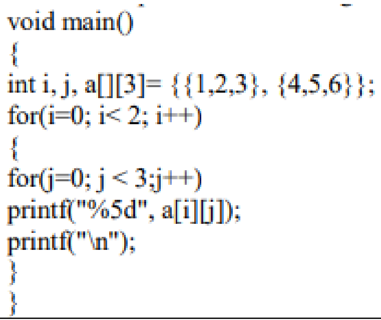
4	Find the output of the following code.
	
	void main()
	{
	char s1[] = "hellow";
	char s2[ ] ="helow";
	int x; x =strncmp(s1,s2,3);
	printf("x = %d", x);
	}
		Understand	Learner to recallstrings and demonstrate the use of string handling functions in finding the output.	CO 3

5	Find the output of the following code.
	
	void main()
	{
	char s1[] = "NEW DELHI";
	char s2[] ="BANGALORE";
	strncpy(s1,s2,4);
	printf("%s", s1);
	}
		Understand	Learner to recallstrings and demonstrate the use of string handling functions in finding the output.	CO 3

6	Find the output of the following code.
	
	void main()
	{
	char s1[] = "NEW DELHI";
	char s2[] ="NEW";
	printf("%d",strstr(s1,s2));
	}
		Understand	Learner to recallstrings and demonstrate the use of string handling functions in finding the output.	CO 3

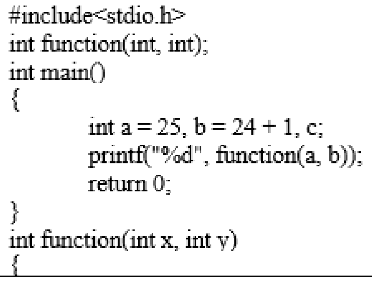
7	Find the output of the following code.
	
	void main()
	{
	int a[4][3];
	printf("%d",sizeof(a));
	}
		Understand	Learner to recallarrays and demonstrate the use of two dimensional arrays, sizeof operator in finding the output.	CO 3

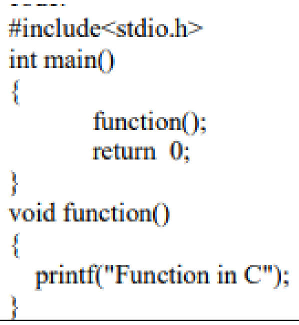
8	Compare the string handling functions strcat() and strncat().	Remember	-	CO 3

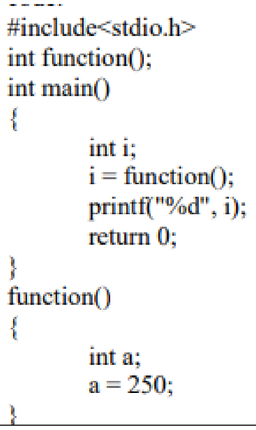
9	FFind the output of the following code.
	
	void main()
	{
	int i, j, a[][3]= {{1,2,3}, {4,5,6}};
	for(i=0; i< 2; i++)
	{
	for(j=0; j < 3;j++)
	printf("%5d", a[i][j]);
	printf("\n");
	}
	}
		Understand	Learner to recallarrays and demonstrate the use of two dimensional arrays in finding the output.	CO 3

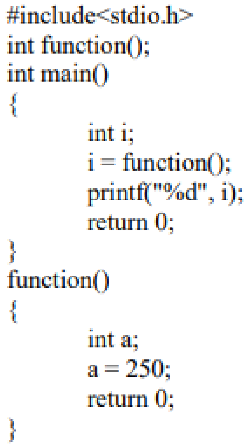
10	Explain the following functions string handling functions. i. strcmp() ii.strrev()	Remember	–	CO 3
11	Define the function with example in C.	Remember	-	CO 3
12	List the types of functions in C.	Remember	-	CO 3

13	Differentiate recursive function and non-recursive function	Remember	-	CO 3
14	Explain various parameter passing methods in C.	Remember	-	CO 3
15	Explain recursion with example.	Remember	-	CO 3
16	List the types of storage classes in C.	Remember	-	CO 3

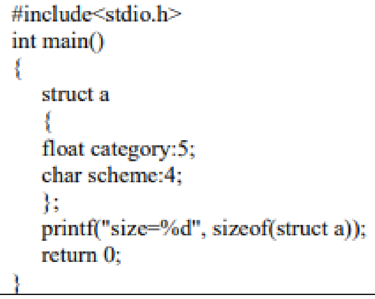
17	Predict the output of the following code.
	
	#include<stdio.h>
	int function(int, int);
	int main()
	{
	int a = 25, b = 24 + 1, c;
	printf("%d", function(a, b));
	return 0;
	}
	int function(int x, int y)
	{
		Understand	Learner to recallfunctions and demonstrate the use of function calling methods predicting the output.	CO 3

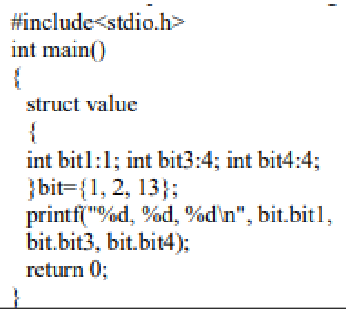
18	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	function();
	return 0;
	}
	void function()
	{
	printf("Function in C");
	}
		Understand	Learner to recallfunctions and demonstrate the use of function calling methods predicting the output.	CO 3

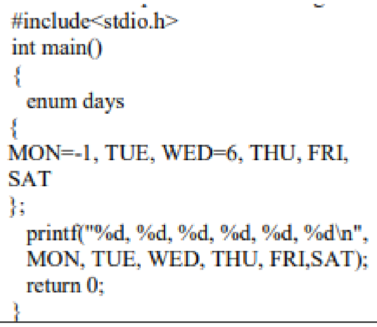
19	Predict the output of the following code.
	
	#include<stdio.h>
	int function();
	int main()
	{
	int i;
	i = function();
	printf("%d", i);
	return 0;
	}
	function()
	{
	int a;
	a = 250;
	}
		Understand	Learner to recallfunctions and demonstrate the use of function calling methods predicting the output.	CO 3

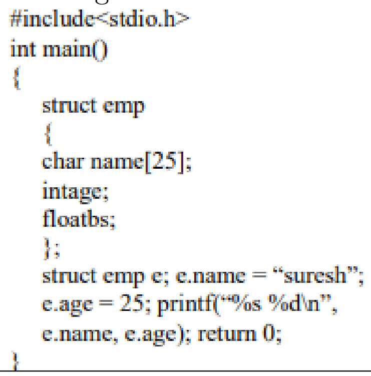
20	Predict the output of the following code.
	
	#include<stdio.h>
	int function();
	int main()
	{
	int i;
	i = function();
	printf("%d", i);
	return 0;
	}
	function()
	{
	int a;
	a = 250;
	return 0;
	}
		Understand	Learner to recallfunctions and demonstrate the use of function calling methods predicting the output.	CO 3

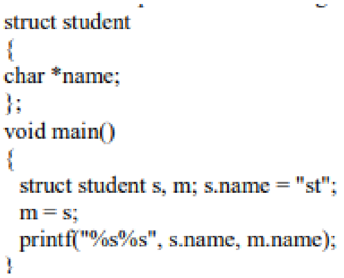
MODULE IV				
POINTER AND STRUCTURES				
PART A- PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS				

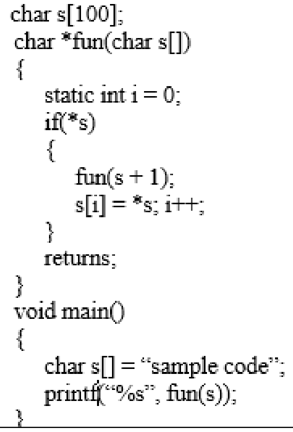
1	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	struct a
	{
	float category:5;
	char scheme:4;
	};
	printf("size=%d", sizeof(struct a));
	return 0;
	}
		Understand	Learner to recallstructures and demonstrate the use of structures to predict the output.	CO 4

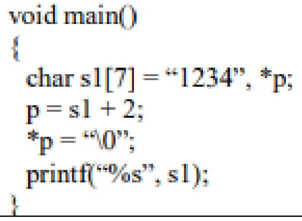
2	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	struct value
	{
	int bit1:1; int bit3:4; int bit4:4;
	}bit={1, 2, 13};
	printf("%d, %d, %d\n", bit.bit1, bit.bit3, bit.bit4);
	return 0;
	}
		Understand	Learner to recallstructures and demonstrate the use of structures, bit fields to predict the output.	CO 4

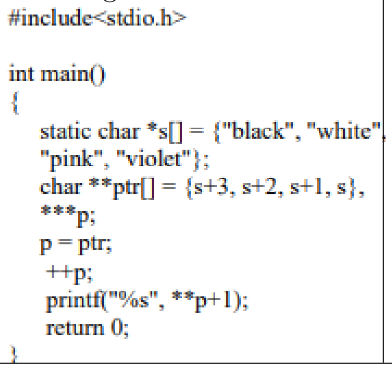
3	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	enum days
	{
	MON=-1, TUE, WED=6, THU, FRI,
	SAT
	};
	printf("%d, %d, %d, %d, %d, %d\n",
	MON, TUE, WED, THU, FRI,SAT);
	return 0;
	}
		Understand	Learners to recall enumerated data type and demonstrate the use of enum type to predict the output.	CO 4

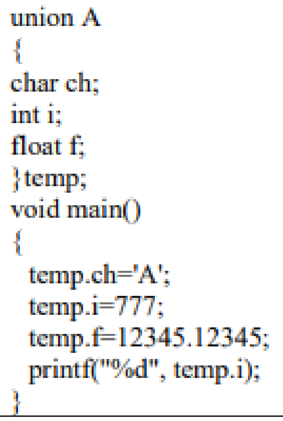
4	Identify the error in the following.
	
	#include<stdio.h>
	int main()
	{
	struct emp
	{
	char name[25];
	intage;
	floatbs;
	};
	struct emp e; e.name = “suresh”;
	e.age = 25; printf(“%s %d\n”,
	e.name, e.age); return 0;
	}
		Understand	This would require the learner to recall the priority then explain the parallel priority	CO 4

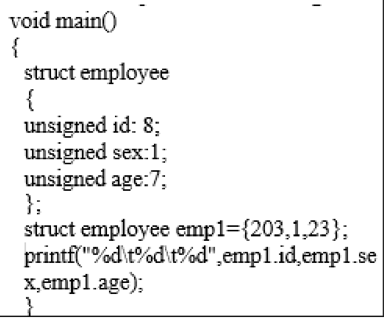
5	Predict the output of the following code.
	
	struct student
	{
	char *name;
	};
	void main()
	{
	struct student s, m; s.name = "st";
	m = s;
	printf("%s%s", s.name, m.name);
	}
		Understand	This would require the learner to recall the control memory then explain the data transfer with handshaking	CO 4

6	Predict the output of the following code.
	
	char s[100];
	char *fun(char s[])
	{
	    static int i = 0;
	    if(*s)
	    {
	        fun(s + 1);
	        s[i] = *s; i++;
	    }
	    returns;
	}
	void main()
	{
	    char s[] = “sample code”;
	    printf(“%s”, fun(s));
	}
		Understand	This would require the learner to recall the memory then explain the characteristics of memory	CO 4

7	Predict the output of the following code.
	
	void main()
	{
	  char s1[7] = “1234”, *p;
	  p = s1 + 2;
	  *p = “\0”;
	  printf(“%s”, s1);
	}
		Understand	This would require the learner to recall the instruction then explain the privileged instruction	CO 4

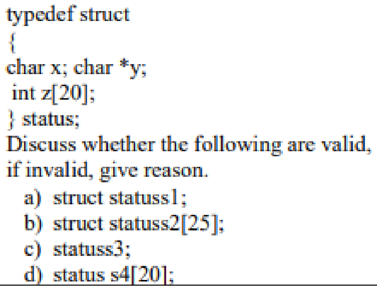
8	Predict the output of the following code.
	
	#include<stdio.h>
	
	int main()
	{
	    static char *s[] = {"black", "white",
	    "pink", "violet"};
	    char **ptr[] = {s+3, s+2, s+1, s},
	    ***p;
	    p = ptr;
	     ++p;
	     printf("%s", **p+1);
	     return 0;
	}
		Understand	This would require the learner to recall the bus then explain the asynchronous bus with read and write cycles	CO 4

9	Predict the output of the following code.
	
	union A
	{
	char ch;
	int i;
	float f;
	}temp;
	void main()
	{
	  temp.ch='A';
	  temp.i=777;
	  temp.f=12345.12345;
	  printf("%d", temp.i);
	}
		Apply	The learner to define memory then explain about data transfer assigning to memory organization	CO 4

10	Predict the output of the following code.
	
	void main()
	{
	  struct employee
	  {
	  unsigned id: 8;
	  unsigned sex:1;
	  unsigned age:7;
	  };
	  struct employee emp1={203,1,23};
	  printf("%d\t%d\t%d",emp1.id,emp1.se
	  x,emp1.age);
	  }
		Understand	This would require the learner to recall the memory then explain the interface in memory organization	CO 4

PART-B LONG ANSWER QUESTIONS				
1	Write a C program to read your full name, Date of birth and display the same using the concept of nested structure.	Understand	Learner to recallstructures and demonstrate the use of structures to get the desired output.	CO 4
2	Write a C program to maintain a book structure containing name,author and pages as structure members. Pass the address of structure variable to a user defined function and display the contents.	Understand	Learner to recallstructures and demonstrate the use of structures to create book structure with specified requirements.	CO 4

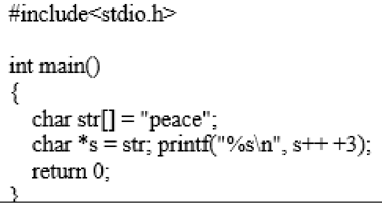
3	A marketing company is having 50 employees and it maintains employee records in terms of their empid, empname, desg, salary, quantity, sales amount. The company gives 10% hike in salary to the employees if their sales amount is more than 50000/-. Write a C program that displays the employee records who got hike in salary.	Understand	Learner to recallstructures and demonstrate the use of structures to displays the employee records who got hike in salary.	CO 4
4	IARE College is maintaining student attendance records by storing rollno, stdname, attendance percentage in 5 different subjects. Write a C program using structures to find the average attendance percentage and print the following a. If attendance percentage ¿=75 then print student is eligible for writing final exam. b. If attendance percentage ¿= 65 and ¡75 then print studentisincondonationlist. c. Otherwise not eligible for writingexams.	Understand	Learner to recallstructures and demonstrate the use of structures to find the average attendance percentage.	CO 4

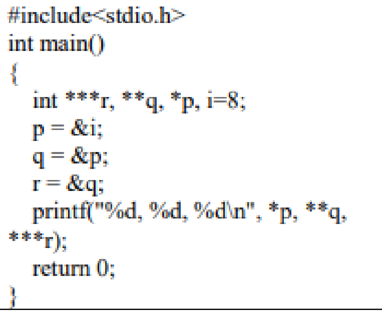
5	Consider the declaration of the structure
	
	typedef struct
	{
	char x; char *y;
	int z[20];
	} status;
	Discuss whether the following are valid, if invalid, give reason.
	a) struct statuss1;
	b) struct statuss2[25];
	c) statuss3;
	d) status s4[20];
		Understand	Learner to recall structures and demonstrate the use of structures to validate the expressions.	CO 4

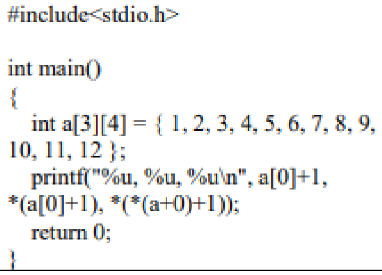
6	Compare and explain the following with suitable examples a. NestedStructures b. Arrayofstructures	Understand	Learner to recallstructures and demonstrate the use of nestedstructures, array of structures	CO 4
7	Explain the following with suitable example a. self referential structures b. enumerated types	Understand	Learner to recallstructures and demonstrate the use of selfreferential structures, enumeratedtypes.	CO 4
8	Write a C program to pass a copy of the entire structure namedstores containing members product name, price and quantity to a function.	Understand	Learner to recallstructures and demonstrate the use of structures to pass a copy of the entire structure.	CO 4
9	Explain the meaning and purpose of the following: a. struct keyword b. typedef keyword c. sizeof operator	Understand	Learner to recallstructures and demonstrate the use of struct , typedef, sizeof operators.	CO 4
10	Define slack byteand explain how it affects the implementation ofstructures through sample code.	Remember	–	CO 4
11	Write a C program to maintain a record of n student details using an array of structures with four fields (rollno, name, marks and grade). Assume appropriate data type for each field. Print the marks of the student name as input.	Understand	Learner to recallstructures and demonstrate the use of structures to maintain a record of n student details.	CO 4
12	Define a structure called complex consisting of two floating point numbers x and y and declare a variable p of type complex. Assign initial values 0.0 and 1.1 to the members.	Apply	Learner to recallstructures and demonstrate the use of structures to define complex numbers.	CO 4

13	Define a structure data type called time struct containing 3 members integer hour, integer minute and integer second. Develop a program that would assign values to the individual members and display the time in the following format:16 : 40 : 51 Learner to recallstructures and demonstrate the use of structures to access the elements of structure.	Apply	Learner to recallstructures and demonstrate the use of structures to access the elements of structure	CO 4
14	Define a structure named census with the following 3 members a. A character array city[ ] to storenames. b. A long integer to store population of thecity. c. A float member to store the literacylevel. Write a program to do thefollowing: d. To read details for 5 cities randomly using an arrayvariable. e. To sort the listalphabetically. f. To sort the list based on literacylevel. g. To sort the list based onpopulation. h. To display sortedlists.	Apply	Learner to recallstructures and demonstrate the use of structures in implementing the desired output.	CO 4

15	Define a structure that can describe a hotel. It should have members that include the name, address, grade, average room charge, and number of rooms. Write functions to perform the following operations: a. To print out hotels of a given grade in order ofcharges. b. To print out hotels with room charges less than a givenvalue	Apply	Learner to recallstructures and demonstrate the use of structures in implementing the desired output	CO 4
16	Define a structure called cricket that will describe the following information: Player name ,Team name ,Batting average using cricket, declare an array play program to read the information about all the 50players and print a teamwise with their batting average.	Apply	Learner to recallstructures and demonstrate the use of structures in implementing the desired output	CO 4
17	IARE maintains salary details of every employee by storing their name, department, basic pay, da, hra and cca. Store this information in a nested structure and display the salary of an employee.	Apply	Learner to recallstructures and demonstrate the use of structures to display the salary of an employee.	CO 4

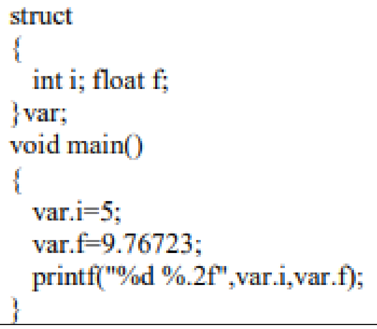
18	Predict the output of the following code.
	
	#include<stdio.h>
	
	int main()
	{
	char str[] = "peace";
	char *s = str; printf("%s\n", s++ +3);
	return 0;
	}
		Understand	Learner to recallpointers and demonstrate the use of pointers to predict the output.	CO 4

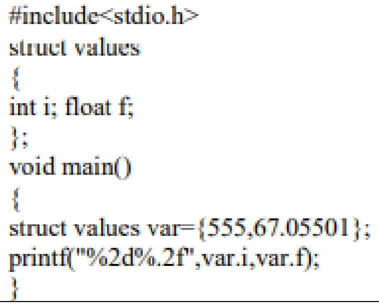
19	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	   int ***r, **q, *p, i=8;
	   p = &i;
	   q = &p;
	   r = &q;
	   printf("%d, %d, %d\n", *p, **q, ***r);
	   return 0;
	}
		Understand	Learner to recallpointers and demonstrate the use of pointers to predict the output	CO 4

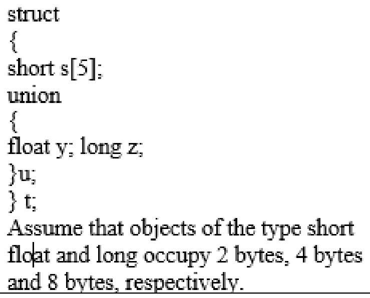
20	Predict the output of the following code.
	
	#include<stdio.h>
	
	int main()
	{
	   int a[3][4] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
	   printf("%u, %u, %u\n", a[0]+1, *(a[0]+1), *(*(a+0)+1));
	   return 0;
	}
		Understand	Learner to recallpointers, arrays and demonstrate the use of these to predict the output	CO 4

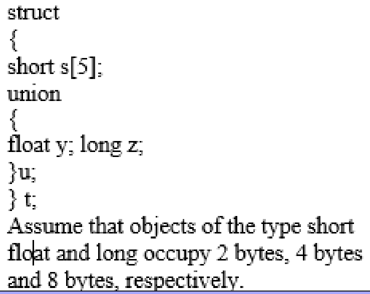
PART-C SHORT ANSWER QUESTIONS				
1	Define the term pointer and state the applications.	Remember	-	CO 4
2	Define generic pointers and Null pointers in C.	Remember	-	CO 4
3	List the functions used for dynamic memory allocation in C.	Remember	-	CO 4
4	Explain pointer to pointer in C.	Remember	-	CO 4
5	Explain bit fields in C.	Remember	-	CO 4
6	Explain Preprocessor directives with examples.	Remember	-	CO 4
7	Define the term structure and state how the members of a structure are accessed.	Remember	-	CO 4
8	Compare the differences between arrays and structures.	Remember	-	CO 4
9	Explain nested structure in C.	Remember	-	CO 4
10	Compare the differences between structure and union.	Remember	-	CO 4

11	Explain of array of structures in C.	Remember	-	CO 4
12	Write about enumerated data type.	Remember	-	CO 4
13	State the default starting values of enumerated set.	Remember	-	CO 4
14	Explain the usage of typedef with example	Remember	-	CO 4
15	State how to access the members of structure in C.	Remember	-	CO 4
16	Explain Pointers as functions arguments with example.	Remember	-	CO 4

17	Predict the output of the following code.
	
	struct
	{
	   int i; float f;
	}var;
	void main()
	{
	   var.i=5;
	   var.f=9.76723;
	   printf("%d %.2f",var.i,var.f);
	}
		Understand	Learner to recallstructures and demonstrate the use of structures to predict the output	CO 4

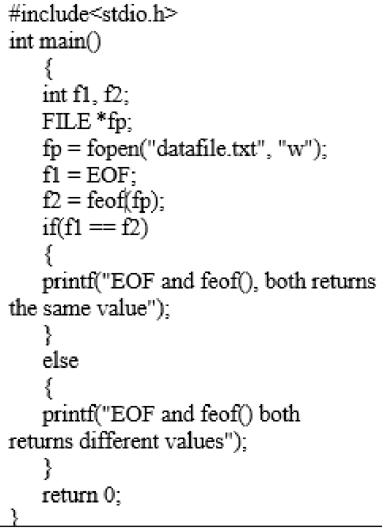
18	Predict the output of the following code.
	
	#include<stdio.h>
	struct values
	{
	int i; float f;
	};
	void main()
	{
	struct values var={555,67.05501};
	printf("%2d%.2f",var.i,var.f);
	}
		Understand	Learner to recallstructures and demonstrate the use of structures to predict the output	CO 4

19	Consider the following C declaration and find the size required by the structure.
	
	struct
	{
	short s[5];
	union
	{
	float y; long z;
	}u;
	} t;
	Assume that objects of the type short float and long occupy 2 bytes, 4 bytes and 8 bytes, respectively.
		Understand	Learner to recallstructures and demonstrate the use of structures to find size of structure	CO 4

20	Predict the output of following C
	
	struct
	{
	short s[5];
	union
	{
	float y; long z;
	}u;
	} t;
	Assume that objects of the type short float and long occupy 2 bytes, 4 bytes and 8 bytes, respectively.
		Understand	Learner to recallstructures and demonstrate the use of structures to predict the output	CO 4

MODULE V				
FILE HANDLING AND APPLICATIONS IN C				
PART A-PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS				
1	C program to read name and marks of n number of students and store them in a file.	Understand	TLearner to recall files and demonstrate the use of files to read name and marks of n number of students and store them in a file	CO 5

2	Write a C program that request for a file name and an integer known as offset value. The program then reads the file starting from the location specified by the offset value and prints the contents on the screen. If the offset value is a positive integer then printing skips that many lines. If it is negative number it prints that many lines from the end of the file. An appropriate error message should be printed ifanything goes wrong.	Understand	Learner to recall files and demonstrate the use of files to implement the program	CO 5
3	Write a menu driven C program to add, display, search, update and delete the student record. Every student record contains name, roll no, age and marks in individual subjects.	Understand	Learner to recall files and demonstrate the use of files to implement the program.	CO 5
4	Write a function that, given a binary file, copies the odd items (items 1,3,5, . . . ,n) to a second binary file and the even items (items 2,4,6, . . . ,n) to a third binary file. After all items have been copied, print the contents of both output files	Understand	Learner to recall files and demonstrate the use of files to implement the program.	CO 5
5	Write a C Program to Reverse the Contents of a File and Print it.	Understand	Learner to recall files and demonstrate the use of files to reverse the Contents of a File and Print it.	CO 5
6	Write a C Program to Count No of Lines, Blank Lines, and Comments in a given file.	Understand	Learner to recall files and demonstrate the use of files to reverse the contents of a File and Print it.	CO 5

7	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	    {
	    int f1, f2;
	    FILE *fp;
	    fp = fopen("datafile.txt", "w");
	    f1 = EOF;
	    f2 = feof(fp);
	    if(f1 == f2)
	    {
	    printf("EOF and feof(), both returns the same value");
	    }
	    else
	    {
	    printf("EOF and feof() both returns different values");
	    }
	    return 0;
	}
		Understand	Learner to recallfiles and demonstrate the use of files to predict the output.	CO 5

8	C program to read name and marks of n number of students from and store them in a file. If the file previously exits, add the information to the file.	Understand	Learner to recall files and demonstrate the use of files to implement the program.	CO 5
9	Write a C program to copy number of bytes from specific offset to another file	Understand	Learner to recall files and demonstrate the use of files to copy number of bytes from specific offset to another file.	CO 5
10	Write a program in C to read the file and store the lines into an array.	Understand	Learner to recall files and demonstrate the use of files to read the file and store the lines into an arra	CO 5
PART-B LONG ANSWER QUESTIONS				
1	Write a C program to read a text file containing some paragraph. Use fseek()functionandreadthetextafter skipping ncharactersfrombeginningofthe file.	Understand	Learner to recallfiles and demonstrate the use of file functions to implement program.	CO 5

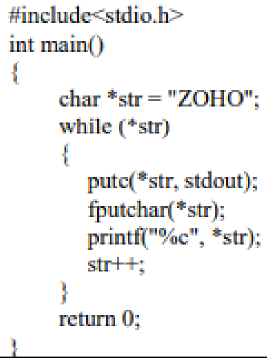
2	Explain the following functions through a sample program which reads a file "test.txt" . a) ftell() b) fseek() c) rewind()	Remember	–	CO 5
3	WWrite a C program to read a text file "sample.txt" and print the following. a) Substring of N characters from theposition I. b) Reverse order of substring of N characters producedin a.	Understand	Learner to recallfiles and demonstrate the use of file functions to implement the program.	CO 5
4	Write the syntax of the following file I/O functions and Explain every option in each function with suitable example : a. fopen() b. fclose() c. fread() d. fwrite()	Remember	—	CO 5
5	Write a program in C to create and store information in a text file. Example : Input a sentence for the file : This is the content of the file test.txt. Expected Output : The file test.txt created successfully...!!	Understand	Learner to recallfiles and demonstrate the use of file functions to create and store information in a text file.	CO 5

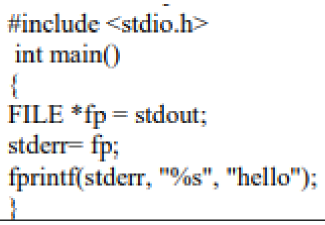
6	Write a C program to open a file names INVENTORY and store in it the following. data Item number price quantityPrinter P1 Scanner S200 5500 5 Hard disk H300 4500 8 Read the data from the INVENTORY file and display the inventory table with the value of each item. [Hint: value = price * quantity and use fprintf() and fscanf() functions]	Understand	Learner to recallfiles and demonstrate the use of file functions to implement the program	CO 5
7	Write a C program to read a given file, convert first letter of each word into uppercase and copy the contents of converted file into a new file.	Understand	Learner to recallfiles and demonstrate the use of file functions to implement the program.	CO 5
8	Write a Cprogram to read name and marks of n number of students from user and store them in a file. If the file previously exists, then add the information of n students to the end of existing content.	Understand	Learner to recallfiles and demonstrate the use of file functions to implement the program.	CO 5
9	Write a C program to print the following from a given file: a) Number of characters b) Number ofspaces c) Number of tabs d) Number ofnewlines.	Understand	Learner to recallfiles and demonstrate the use of file functions to implement the program.	CO 5
10	Create a structure named employee containing name, age and basic pay. Write a C program to create 5 employee records andwrite to a file. Thenread the records from file and display it.	Understand	Learner to recallfiles and demonstrate the use of files to implement the program.	CO 5

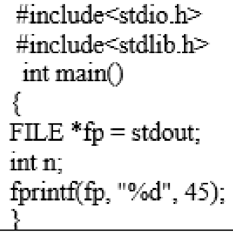
11	Write a C program to maintain a record of “n” student details using an array of structures with four fields (Roll number, Name, Marks, and Grade). Each field is of an appropriate data type. Print the marks of the student given student name as input.	Understand	Learner to recallfiles and demonstrate the use of files to implement the program.	CO 5
12	Write a program in C to read an existing file. Example : Input the file name to be opened : test.txt Expected Output : The content of the file test.txt is : This is the content of the file test.txt.	Understand	TThis would require the learner to recall the pipeline then explain the segment pipeline.	CO 5
13	Write a program in C to write multiple lines in a text file. Test Data : Input : The number of lines to be written : 4 test line 1 test line 2 test line 3 test line 4 Expected Output : The content of the file test.txt is : test line 1 test line 2 test line 3 test line 4	Understand	Learner to recallfiles and demonstrate the use of files to write multiple lines in a text file.	CO 5
14	Write a program in C to find the Number of lines in a Text File.	Understand	Learner to recall files and demonstrate the use of files to find the Number of lines in a Text File.	CO 5

15	Write a program in C to count a number of words and characters in a file.	Understand	Learner to recallfiles and demonstrate the use of files to count a number of words and characters in a file.	CO 5
16	Write a program in C to find the content of the file and number of lines in a Text File. Test Data : Input : The filename to be opened : test.txt Expected Output : The content of the file test.txt are : test line 1 test line 2 test line 3 test line 4	Understand	Learner to recallfiles and demonstrate the use of files to find the content of the file and number of lines in a text file.	CO 5
17	Write a program in C to delete a specific line from a file.	Apply	Learner to recallfiles and demonstrate the use of files to delete a specific line from a file.	CO 5
18	Write a program in C to replace a specific line with another text in a file.	Apply	Learner to recallfiles and demonstrate the use of files to replace a specific line with another text in a file.	CO 5
19	Write a program in C to copy a file in another name.	Apply	Learner to recall files and demonstrate the use of files to copy a file in another name.	CO 5
20	Write a program in C to merge two files and write it in a new file.	Apply	Learner to recall files and demonstrate the use of files to merge two files and write it in a new file.	CO 3
PART-C SHORT ANSWER QUESTIONS				
1	Define file and list basic operations of a file.	Remember	—	CO 5
2	Explain various text file opening modes.	Remember	—	CO 5
3	State the various types of status enquiry library functions in C.	Remember	—	CO 5

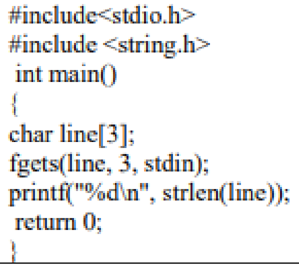
4	Explain ftell() function with example.	Remember	—	CO 5
5	Write the purpose of fseek() with example.	Remember	—	CO 5
6	Write the syntax and usage of rewind( ).	Remember	—	CO 5
7	Explain a file opening modes with example.	Remember	—	CO 5
8	List the different types of files.	Remember	—	CO 5
9	List the application of files.	Remember	—	CO 5

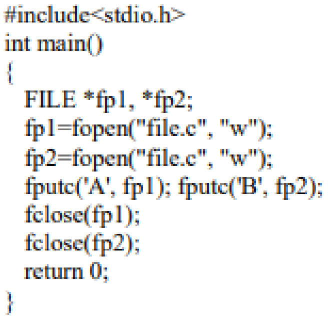
10	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	char *str = "ZOHO";
	while (*str)
	{
	putc(*str, stdout);
	fputchar(*str);
	printf("%c", *str);
	str++;
	}
	return 0;
	}
		Understand	Learner to recallfiles and demonstrate the use of file pointers to predict the output.	CO 5

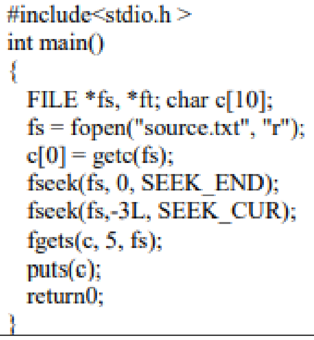
11	Predict the output of the following code.
	
	#include <stdio.h>
	int main()
	{
	FILE *fp = stdout;
	stderr= fp;
	fprintf(stderr, "%s", "hello");
	}
		Understand	Learner to recallfiles and demonstrate the use of file pointers to predict the output.	CO 5

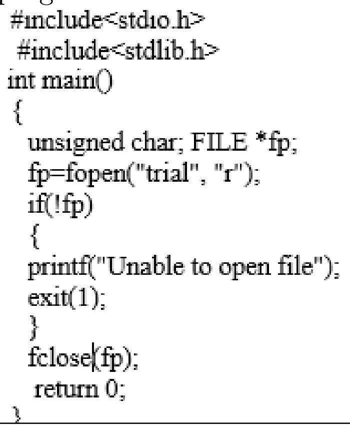
12	Find the output of this code?
	
	#include<stdio.h>
	#include<stdlib.h>
	int main()
	{
	FILE *fp = stdout;
	int n;
	fprintf(fp, "%d", 45);
	}
		Understand	Learner to recallfiles and demonstrate the use of file pointers to predict the output	CO 5

13	Explain the error handling function for files in C.	Remember	—	CO 5

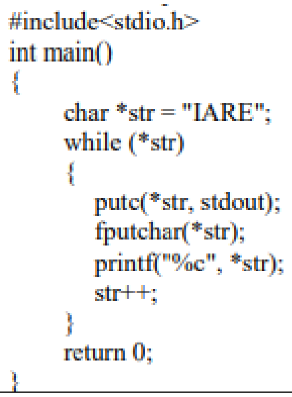
14	Predict the output of this code?
	
	#include<stdio.h>
	#include <string.h>
	int main()
	{
	char line[3];
	fgets(line, 3, stdin);
	printf("%d\n", strlen(line));
	return 0;
	}
		Understand	Learner to recallfiles and demonstrate the use of file permissions to predict the output.	CO 5

15	Predict the content of 'file.c' after executing the following program?
	
	#include<stdio.h>
	int main()
	{
	FILE *fp1, *fp2;
	fp1=fopen("file.c", "w");
	fp2=fopen("file.c", "w");
	fputc('A', fp1); fputc('B', fp2);
	fclose(fp1);
	fclose(fp2);
	return 0;
	}
		Understand	Learner to recallfiles and demonstrate the use of files to predict the output.	CO 5

16	If the file 'source.txt' contains a line "Be my friend", predict the output of below program?
	
	#include<stdio.h >
	int main()
	{
	FILE *fs, *ft; char c[10];
	fs = fopen("source.txt", "r");
	c[0] = getc(fs);
	fseek(fs, 0, SEEK_END);
	fseek(fs,-3L, SEEK_CUR);
	fgets(c, 5, fs);
	puts(c);
	return0;
	}
		Understand	Learner to recallfiles and demonstrate the use of files to predict the output.	CO 5

17	Identify the error in the program?
	
	#include<stdio.h>
	#include<stdlib.h>
	int main()
	{
	unsigned char; FILE *fp;
	fp=fopen("trial", "r");
	if(!fp)
	{
	printf("Unable to open file");
	exit(1);
	}
	fclose(fp);
	return 0;
	}
		Understand	Learner to recallfiles and demonstrate the use of files to predict the output.	CO 5

18	Explain why fseek() should be preferred over rewind().	Remember	—	CO 5
19	Differentiate between file opening mode r+ and w+.	Remember	—	CO 5

20	Predict the output of the following code.
	
	#include<stdio.h>
	int main()
	{
	char *str = "IARE";
	while (*str)
	{
	putc(*str, stdout);
	fputchar(*str);
	printf("%c", *str);
	str++;
	}
	return 0;
	}
		Understand	Learner to recallfiles and demonstrate the use of files to predict the output.	CO 5

Course Coordinator: **HOD CSE**
Dr. Shaik Jakeer Hussain, Associate Professor