

MAHARASHTRASTATE BOARD OF TECHNICAL EDUCATION

(Autonomous)

(ISO/IEC - 27001 - 2005 Certified)

22218

# MODEL ANSWER

## **SUMMER-19 EXAMINATION**

## Subject Title: C' Programming Language

Subject Code: 22218

## **Important Instructions to examiners:**

- 1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
- 2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
- 3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
- 4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for anyequivalent figure drawn.
- 5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
- 6) In case of some questions credit may be given by judgment on part of examiner of relevant answer based on candidate's understanding.
- 7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q.N.	Answer		Marking Scheme
Q.1		Attempt any Five :		10 M
	a)	List any four relational operators with	their use.	2M
	Ans:	Relational operators are use in conditi	Each	
			Use	with its use
		Operator		
		==	equal to	72 NI
		!=	Not equal to	



		<	less than	(Any 4)
		>	Greater than	
		<=	Less than equal to	
		>=	Greater than equal to	
b)	Give synta	ax of switch-case statement.		2M
Ans:	Syntax:			Correct
	switch(var	iable)		syntax 2M
	{			
	case value	1:		
	statements			
	break;			
	case value	2:		
	statements	• •		
	break;			
	default:			
	statements	• •		
	break;			
	}			
c)	Give svnta	ax of for loop.		2M
•				C
Ans:	Syntax:			Correct
	for(initializ	zation; condition; increment/decre	ment)	syntax 21vi
	{			
	Statements	;		
	}			
d)	State any	two differences between call by	value and call by reference.	2M
				Any two
	Sr.	Call by value	Call by reference	differences
	No.			1M each
		A copy of actual arguments (value) is passed to respective	Address of actual arguments is	
		formal arguments	passed to formal arguments.	
Ans:	2	Actual arguments will remain	Alteration to actual arguments is	
		safe, they cannot be modified	possible within from called	



	3	accidentally. Address of the actual and formal arguments are different Changes made inside the function are not reflected in other functions	function; therefore the code must handle arguments carefully else you get unexpected results. Address of the actual and formal arguments are the same Changes made in the function are reflected outside also.	
<b>e</b> )	Define poi	nter and state any two uses of p	ointer.	2M
Ans:	Definition which is of Uses of po 1. Pointers 2. Pointers function 3. They ca argume	A pointer is a variable that stores similar data type. inter:- s are used for dynamic memory m permit references to functions an as as arguments to other functions an be used to return multiple value onts.	s memory address of another variable anagement. d thereby facilitating passing of es from a function via function	1M any two correct uses 1/2M each)
<b>f</b> )	State the u	use of 🖈, *, & symbols used in j	pointers.	2M
Ans:	* operator	:-		1M
	It is used to <b>Example:</b>	o declare a pointer variable.		
	The	e above statement declares ptr as a	n integer pointer variable.	
			OR	
	It is also u pointer var <b>Example:</b> The variable	sed as value at operator i.e. it re iable. printf("%d", *ptr); e above statement displays valu	ads the value from the address stored in he present at the address stored in ptr	
	e oporato			1M
	It is used to <b>Example:</b>	p retrieve address of a variable fro int *ptr,a; ptr=&a	om memory.	



		The above statement stores the address of variable a in the pointer variable	
		ptr.	
	<b>g</b> )	Define structure.	2M
	Ans:	Definition:	2M
		A structure is a collection of one or more variables of same or different data types	
		grouped together under a single name.	
Q 2		Attempt any Three :	12M
	a)	Explain any four data types used in C with example.	<b>4</b> M
	Ans:	Data type in C are:	Explanation
		short int/cigned short int/int/signed int/unsigned int/long int/signed long int/unsigned	four data
		short musighed short mumusighed mutuhsighed musighed musighed	type with
		long int/char/float/double/long double/void	example
		Integer data type:	••••••••••••••••••••••••••••••••••••••
		• <b>short int/signed short int:</b> It is used to declare integer type variable. It occupy 8 bits(1 byte) memory size to store data such as 10.20.etc.	IM each
		Example: short int number;	
		• <b>int/signed int/unsigned int</b> : It is used to declare integer type variable. It occupy 16 bits (2 bytes) memory size to store data such as 100,200,etc.	
		<ul> <li>long int/signed long int/unsigned long int: It is used to declare integer type</li> </ul>	
		variable. It occupy 32 bits (4 bytes) memory size to store data such as mobile	
		number.	
		Example: long int contactno;	
		Character data type:	
		• <b>char/signed char/unsigned char:</b> It is used to declare character type variable. It occupy 8 bits(1 byte) memory size to store data such as 'a' b' '%' etc.	
		Example: char ch:	
		Floating point data type:	
		• <b>float:</b> It is used to declare floating point type variable. It occupy 32 bits	
		(4 bytes) memory size to store data such as 1.1,2.2,etc.	
		Example: Itoat percentage;	
		• <b>double:</b> It is used to declare floating point type variable. It occupy 64 bits (8 bytes), memory size to store data such as 10,1,22,2 etc.	
		Example: double percentage:	



<pre>long double: It is used to declare floating point type variable. It occupy 80 bits (10 bytes) memory size to store data such as 11.11,21.2,etc. Example: long double percentage; Hata type: 1 data type has no values. When a function does not return any value then the n type of function is specified with void data type. example: void add() { Statements; } in nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use 1 ifelse statement.</pre>	4M 2M
<pre>(10 bytes) memory size to store data such as 11.11,21.2,etc. Example: long double percentage; data type: data type has no values. When a function does not return any value then the n type of function is specified with void data type. example: void add() { Statements; } din nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.</pre>	4M 2M
Example: long double percentage; data type: d data type has no values. When a function does not return any value then the n type of function is specified with void data type. example: void add() { Statements; } in nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.	4M 2M
Hata type:         I data type has no values. When a function does not return any value then the         In type of function is specified with void data type.         example:         void add()         {         Statements;         }         in nested if-else statement with syntax and example.         ition:         se statement used inside if statement used in a program is called as nested         se statement. When series of decisions are involved in a program we can use         I ifelse statement.	4M 2M
<pre>d data type has no values. When a function does not return any value then the n type of function is specified with void data type. example: void add() { Statements; } din nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.</pre>	4M 2M
<pre>n type of function is specified with void data type. example: void add() { Statements; } in nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.</pre>	4M 2M
example: void add() { Statements; } in nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.	4M 2M
<pre>void add() { Statements; } ain nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.</pre>	4M 2M
<pre>{ Statements; } in nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.</pre>	4M 2M
Statements; in nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use i felse statement.	4M 2M
ain nested if-else statement with syntax and example. ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.	4M 2M
in nested if-else statement with syntax and example.         ition:         se statement used inside if statement used in a program is called as nested         se statement. When series of decisions are involved in a program we can use         ifelse statement.	4M 2M
<b>ain nested if-else statement with syntax and example. ition:</b> se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.	4M 2M
ition: se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.	2M
se statement used inside if statement used in a program is called as nested se statement. When series of decisions are involved in a program we can use d ifelse statement.	
se statement. When series of decisions are involved in a program we can use I ifelse statement.	
l ifelse statement.	
X :	2M
condition1)	
if(test condition2)	
statement-1;	
} else	
{	
statement-2;	
}	
nent-3;	
nent-x;	
nent-x; t condition-1 is true, then condition-2 is checked. If condition-2 is true, then ment-1 is evaluated. If condition-2 is false then statement-2 is evaluated and	
nent-x; t condition-1 is true, then condition-2 is checked. If condition-2 is true, then nent-1 is evaluated. If condition-2 is false then statement-2 is evaluated and control is transferred to the statement-x. If condition-1 is false then control	
	hent-3; hent-x; t condition-1 is true, then condition-2 is checked. If condition-2 is true, then



1	example:	1M
	int num=75; if(num<100)Condition 1	
	{ if(num<50)Condition 2 {	
	<pre>printf("Number is less than 50");Statement 1 }</pre>	
	else {	
	printf("Number is greater than 50 but less than 100"); statement 2 }	
	} else {	
	<pre>printf("Number is greater than 100");statement 3 }</pre>	
	In the above example, variable num is initialized to value 75.In condition 1 num is Compared with 100 and the condition evaluates to true. So control passes to condition 2.In condition 2 num is greater than 50 so condition is false. Control passes to statement 2 and output is "Number is greater than 50 but less than 100". Then control comes out of pected if _ also statement	
,	Then control comes out of nested heise statement.	
<b>c</b> )	Explain Array. State two advantages of array.	4M
c) Ans:	Explain Array. State two advantages of array. Explanation:	4M 3M
c) Ans:	Explain Array. State two advantages of array.         Explanation:         An array is a sequenced collection of similar type of data. Values in an array are	4M 3M
c) Ans:	Explain Array. State two advantages of array.         Explanation:         An array is a sequenced collection of similar type of data. Values in an array are stored in Continuous memory locations. All data values stored in an array share a common name. To access individual value from an array, index variable is used. An index variable starts with 0th position. First value in an array is stored at 0th position and last value is stored in size-1 index position. For example, in an array of 5 elements, first value is stored at 0th position and last value is stored at 0th position.	4M 3M
c) Ans:	<ul> <li>Explain Array. State two advantages of array.</li> <li>Explanation: <ul> <li>An array is a sequenced collection of similar type of data. Values in an array are stored in Continuous memory locations. All data values stored in an array share a common name. To access individual value from an array, index variable is used. An index variable starts with 0th position. First value in an array is stored at 0th position and last value is stored in size-1 index position. For example, in an array of 5 elements, first value is stored at 0th position and last value is stored at 0th position.</li> </ul> </li> <li>Syntax to declare an array: <ul> <li>data type arr. name[size]:</li> </ul> </li> </ul>	4M 3M
c) Ans:	Explain Array. State two advantages of array.         Explanation:         An array is a sequenced collection of similar type of data. Values in an array are stored in Continuous memory locations. All data values stored in an array share a common name. To access individual value from an array, index variable is used. An index variable starts with 0th position. First value in an array is stored at 0th position and last value is stored in size-1 index position. For example, in an array of 5 elements, first value is stored at 0th position and last value is stored at 4th index position.         Syntax to declare an array:       data type arr_name[size];         In the above syntax,       In the above syntax,	4M 3M
c) Ans:	Explain Array. State two advantages of array.         Explanation:         An array is a sequenced collection of similar type of data. Values in an array are stored in Continuous memory locations. All data values stored in an array share a common name. To access individual value from an array, index variable is used. An index variable starts with 0th position. First value in an array is stored at 0th position and last value is stored in size-1 index position. For example, in an array of 5 elements, first value is stored at 0th position and last value is stored at 0th position.         Syntax to declare an array:       data type arr_name[size];         In the above syntax,       • data type specify type of data that can be stored in an array.         • arr_name specify name of array.	4M 3M



	<b>Example:</b> int arr[5] = {10, 20, 5, 3, 55};	
	In the above example, an array variable arr is declared and initialized with integer values with the size 5.	
	arr[0]arr[1]arr[2]arr[3]arr[4]10205355	
		<sup>1</sup> ⁄ <sub>2</sub> Each
	Advantages of an array:	(Any 2)
	<ul> <li>Array represents multiple data items of similar type using single name.</li> <li>Adding and removing data element at any index position is possible.</li> </ul>	
d)	List any 2 string functions. Give syntax and use of each function.	4M
Ans:	String functions:	List 1M
	1. strcat ( ) 2. strcmp( ) 3. strcpy ( ) 4. strlen ( ) 5. strlwr( ) 6. strupr( )	Syntox 1/2
	<b>1. strcat</b> ():- This string function is used to join two strings together.	Each
	<b>Syntax:</b> streat (string1, string2); string1 and string2 are character arrays. When the function streat () is executed, string2 is appended to string1 i.e. contents of string2 are added at the end of string1.	Use 1M
	<b>2. strcmp</b> ():- This string function is used to compare the contents of two strings. It returns 0 if both string are equal. Otherwise it returns the numerical difference between the ascii values of the first non matching pair of characters.	Each
	Syntax: strcmp(string1,string2); string1 and string2 are character arrays.	
	<b>3.</b> strcpy():- This string function is used to copy the content of one string to the other string.	
	Syntax: strcpy(string1,string2);	



		The following example will explain it. 1010 1100 	example of any four bitwise operator
		Bitwise OR –   It takes 2 bit patterns, and performs OR operations on each pair of corresponding bits	
		Description:	
		>> – right shift	
		= bitwise AON $<< -$ left shift	
		~ – One''s complement	
		& – Bitwise AND	
		-Bitwise OR	
	Ans:	Bitwise operators:	List 2M
	a)	Enlist any four bitwise operators used in C and give example of each.	<b>4</b> M
Q.3		Attempt any Three:	12M
		Syntax: strupr(string); string is a character array.	
		<b>6. strupr( ):-</b> This string function is used to convert a given string into upper case letters.	
		Syntax: strlwr(string); string is a character array.	
		<b>5. strlwr</b> ():- This string function is used to convert a given string into lower case letters.	
		<b>Syntax:</b> variable_name=strlen(string); string is a character array.variable_name is an integer variable that stores the value of the length of the string return by strlen() function.	
		<b>4.</b> strlen():- This string function is used to count and return number of characters stored in a string.	
		string1 and string2 are character arrays. When strcpy() function executes the contents of string2 are copied into string1.	



Bitwise AND - &	
It takes 2 bit patterns, and perform AND operations with it.	
1010	
1100	
AND 1000	
The Bitwise AND will take pair of bits from each position, and if only both the	bit is
1, the result on that position will be 1.	
Bitwise AND is used to Turn-Off bits.	
Bitwise NOT:	
One"s complement operator (Bitwise NOT) is used to convert each "1-bit to "	0-bit"
and "0-bit to 1-bit", in the given binary pattern. It is a unary operator i.e. it takes	s only
one operand.	
1001	
NOT 0110	
Bitwise XOR ^	
Bitwise XOR ^, takes 2 bit patterns and perform XOR operation with it.	
0101	
0110	
XOR 0011	
Left shift Operator – <<	
The left shift operator will shift the bits towards left for the given number of time	nes.
int a=2<<1;	
Right shift Operator – >>	
The right shift operator will shift the bits towards right for the given numb	per of
times.	
int a=8>>1;	



b)	Explain Pointer Arithmetic.	<b>4M</b>
Ans:	The pointer arithmetic is done as per the data type of the pointer. The basic operations on pointers are	Arithmetic operation
	<b>Increment:</b> It is used to increment the pointer. Each time a pointer is incremented, it points to the	1M Each
	next location with respect to memory size .	
	Example, If ptr is an integer pointer stored at address 1000,then ptr++ shows 1002 as incremented location for an int. It increments by two locations as it requires two bytes storage.	
	Decrement	
	It is used to decrement the pointer. Each time a pointer is decremented, it points to the previous location with respect to memory size.	
	If the current position of pointer is 1002, then decrement operation ptr results in the pointer pointing to the location 1000 in case of integer pointer as it require two bytes storage.	
	Addition: When addition operation is performed on pointer, it gives the location incremented by the added value according to data type.	
	If ptr is an integer pointer stored at address 1000, Then ptr+2 shows $1000+(2*2) = 1004$ as incremented location for an int.	
	Subtraction:	
	When subtraction operation is performed on the pointer variable, it gives the location decremented by the subtracted value according to data type.	
	Eg: If ptr is an integer pointer stored at address 1004,	
	Then ptr-2 shows $1004-(2*2) = 1000$ as decremented location for an int.	
c)	Explain meaning of following statement with reference to Pointer:	<b>4</b> M
	int var = 50;	
	int *p1, *p2;	



	P1= & var;	
	P2= p1;	
Ans:	int var = 50;	Correct
	It is declaration and initialization of integer variable var with value 50.	meaning of each
	int *p1, *p2;	statement 1M
	It is declaration of integer pointer p1 and integer pointer p2.	
	P1= & var;	
	Address of var is assigned to variable P1.	
	P2= p1;	
	Value of p1 is assigned to P2.	
<b>d</b> )	Explain declaration of structure with example.	4M
	<ul> <li>Structure: A structure is a collection of one or more variables of same or different data types grouped together under a single name.</li> <li>struct structure_name </li> <li>Data_type variable 1; Data_type variable 2; <ul> <li>.</li> <li>Data_type variable n;</li> </ul> </li> <li>Variable_name;</li> </ul> <li>Structure variable is used to access members of structure inside main function with dot operator.</li> <li>Variables of structure can be declared as: <ul> <li>Variable of structure can be declared as:</li> <li>Variable of structure can be declared at the end of structure declaration before semi colon or inside the main function.</li> <li>struct book b; //for a single book</li> </ul></li>	
	struct book b[5]; //to store data of 5 books Example: struct book {	Any Example 2M



		char tit[20]; char auth[20]; int price;	
0.4	A)	<pre>}b1; Attempt any THREE :</pre>	12 M
	a)	Write a C program to accept two integer numbers from user and print the result of addition and subtraction.	4M
	Ans:	<pre>#include<stdio.h> #include<conio.h> void main() {</conio.h></stdio.h></pre>	Correct Logic 2M
		<pre>int a,b,add,sub; clrscr(); printf("Enter value for a and b:"); scanf("%d%d",&amp;a,&amp;b); add=a + b; sub=a - b; printf("\nAddition of a and b=%d\n",add); printf("\nSubtraction of a and b=%d",sub); getch(); }</pre>	Correct syntax 2M
	b)	Write a C program to check whether given number is positive or negative and display message accordingly.	<b>4M</b>
	Ans:	<pre>#include <stdio.h> #include <conio.h> void main() {     int num;     clrscr();     printf("\n Enter number: ");</conio.h></stdio.h></pre>	Correct Logic 2M Correct syntax 2M
		scanf("%d",#); if(num > = 0)	



	printf("\n %d is positive number", num);	
	else	
	printf("\n %d is negative number", num);	
	getch();	
	}	
	Output:	
	Enter number: 4	
	4 is positive number	
	Enter number: -6	
	-6 is negative number	
c)	What is an output of following C code:	4M
	#include <stdio.h></stdio.h>	
	Void main()	
	{	
	int a[5]= {10,20,30,40,50};	
	printf("output");	
	for(i=0; i<3; i++)	
	{	
	Print("%d", a[i]);	
	}	
	}	
Ans:	output	Correct
	10 20 30	output 4M



<b>d</b> )	Declare a structure book having elements as book_number, book_title, book_price and also declare array of structure taking input of 10 books using C programming language.	4M
Ans:	#include <stdio.h></stdio.h>	Correct
	#include <conio.h></conio.h>	Logic 2M
	void main()	
	{	
	int i;	~
	struct book {	Correct syntax 2M
	char book_title[50];	
	int book_number;	
	int book_price;	
	}b[10];	
	clrscr();	
	for(i = 0; i < 10;i++)	
	{	
	printf("Enter book number, title and price of a book:");	
	scanf("%d%s%d",&b[i].book_number,b[i].book_title,&b[i].book_price);	
	}	
	printf("The details of book are:\nBook_Number \tTitle \tPrice\n");	
	for(i = 0; i < 10;i++)	
	{	
	printf("\tbook number=%d \tbook title=%s \tbook price =%d\n", b[i].book_number,	
	b[i].book_title,b[i].book_price);	



		}	
		getch();	
		}	
Q.5		Attempt any TWO :	12 M
	a)	Explain if-else statement using syntax and example.	6M
	Ans:	Syntax of if-else statement :	2M
		if (test expression)	
		True-block statement (s)	
		}	
		else {	
		False-block statement (s)	
		} Statement-x;	
		<ul> <li>Explanation : <ol> <li>If-else statement is a decision making statement and is used to control the flow of execution of statements.</li> <li>It allows the computer to evaluate the expression first and then depending on whether the value of the expression is true or false, it transfers the control to the particular statement block.</li> <li>If the test expression is true, then true block statement(s) are executed, immediately following the if statement are executed otherwise false block statement(s) are executed.</li> </ol> </li> </ul>	1M
		Fyample.	3M
		<pre>Example: #include<stdio.h> #include<stdio.h></stdio.h></stdio.h></pre>	NOTE.
		void main()	Any other
		{ int num; printf("Enter the number"); scanf("%d",#)	example shall be considered
		if(num>0) {	



	printf("Number is positive");	
	}	
	else	
	{	
	<pre>printf("Number is negative");</pre>	
	}	
	getch();	
	}	
<b>b</b> )	Write a C program to read string from keyboard and find whether it is palindrome or not.	6M
Ans:	#include <stdio.h></stdio.h>	Correct
	#include <conio.h></conio.h>	logic 3M,
	#include <string.h></string.h>	
	void main()	Correct
	{	syntax 3M
	char str1[20],str2[20];	
	clrscr();	
	printf("Enter String to check if it is palindrome : ");	NOTE:
	scanf("%s",str1);	Any other
	<pre>strcpy(str2,str1);</pre>	example
	strrev(str2);	shall be
	if(strcmp(str1,str2)==0)	considered
	printf("String is a palindrome");	
	else	
	printf("String is not a paliindrome");	
	getch();	
	}	
	(OR)	
	<pre>#include <stdio.h></stdio.h></pre>	
	#include <conio.h></conio.h>	
	#include <string.h></string.h>	
	void main(){	
	char string1[20];	
	int i, length;	
	int flag = 0;	
	clrscr();	
	printf("Enter a string:");	
	<pre>scanf("%s", string1);</pre>	



	length = strlen(string1);	
	for(i=0;i < length ;i++)	
	{	
	if(string1[i] != string1[length-i-1])	
	{	
	flag = 1;	
	break;	
	}	
	}	
	if (flag)	
	printf("%s is not a palindrome", string1);	
	else	
	printf("%s is a palindrome", string1);	
	getch();	
	}	
c)	Write a program to find length of a string.	6M
Ans:	#include <stdio.h></stdio.h>	Correct
	#include <conio.h></conio.h>	logic 3M
	#include <string.h></string.h>	0
	void main()	Correct
	{	syntax 3M
	char str[100];	
	int length;	
	clrscr();	NOTE:
	printf("Enter a string to calculate it's length\n");	Any other
	scanf("%s",str);	example
	length = strlen(str);	shall be
	printf("Length of the string = $%d n$ ", length);	considered
	getch();	
	}	
	(OR)	
	#include <stdio.h></stdio.h>	
	#include <conio.h></conio.h>	
	void main()	
	char s[30];	
	int i;	



	printf("Enter a string: "):	
	scanf("%s" s):	
	for $(i - 0; s[i] ! - '(0'; ++i))$	
	$\mathbf{printf("I ength of string: %d" i)}$	
	gatab():	
	getch();	
06	Attempt on TWO:	12M
Q.0	Attempt any 1 wo:	12111
a)	Write a program to add two 3 * 3 matrices.	6M
Ans:	#include <stdio.h></stdio.h>	Correct
	#include <conio.h></conio.h>	logic 3M
	void main()	
	{	
	int a[3][3], b[3][3], add[3][3], i, j;	Correct
	clrscr();	correct syntax 3M
	printf("Enter values for first matrix: \n");	Syntax Sivi
	for(i=0:i<3:i++)	
	{	
	for(i=0:i<3:i++)	NOTE:
		Any other
	printf("Enter matrix 1 entry(%d %d); " i i);	example
	scanf("%d" &a[i][i]).	shall be
		considered
	}	
	for (i. 0 vi (2 vi ( ))	
	10f(1=0;1<3;1++)	
	for(j=0;j<3;j++)	
	printf("Enter matrix 2 entry(%d,%d): ",1,J);	
	scanf("%d",&b[i][j]);	
	}	
	}	
	//Performing addition	
	for(i=0;i<3;i++)	
	{	
	for(j=0;j<3;j++)	
	{	



	add[i][j] = a[i][j] + b[i][j];	
	}	
	}	
	printf("Addition matrix is: $n$ ");	
	for(i=0;i<3;i++)	
	{	
	for(j=0;j<3;j++)	
	{	
	<pre>printf("%d\t",add[i][j]);</pre>	
	}	
	<pre>printf("\n");</pre>	
	}	
	getch();	
	}	
b)	Write a program to add two numbers using function.	6M
Ans:	#include <stdio.h></stdio.h>	Correct
	#include <conio.h></conio.h>	program
	void add(int, int);	3M
	void main()	Connect
	{	Correct logia 3M
	int a, b;	logic SM
	clrscr();	
	printf("Enter two number: ");	
	scanf("%d%d",&a,&b);	
	add(a,b);	NOTE:
	getch();	Any other
	}	example
	void add(int a, int b)	shall be
	{	considered
	printf("Addition of %d and %d is %d",a,b,a+b);	
	}	
	}	
	Write a C program to create structure for student having data members like	
c)	roll_no. name and marks in 3 subjects and display % of marks as output of	6M
<i>c</i> ,	program.	
Ans:	#include <stdio.h></stdio.h>	Structure
1	#include <conio.h></conio.h>	declaration
	struct student	2 <b>M</b>



<pre>{     int roll_no;     char name[20];     int sm1,sm2,sm3; }s; void main() {     float percent;     clrscr();     printf("Enter student roll no, name, subject1 marks,subject2 marks ,subject3     marks ;"); }</pre>	Accept elements 2M Display answer 2M
<pre>scanf("%d%s%d%d%d",&amp;s.roll_no,s.name,&amp;s.sm1,&amp;s.sm2,&amp;s.sm3); percent=(s.sm1+s.sm2+s.sm3)/3; printf("\nRollNo\tName\tPercentage"); printf("\n%d\t%s\t%.2f",s.roll_no,s.name,percent); getch(); }</pre>	NOTE: Any other example shall be considered