



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

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 any equivalent 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 judgement 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
1.	(a) Ans.	Attempt any TEN of the following: Define following terms: (i) Keyword (ii) Token (i) Keyword: Keyword is pre-defined or reserved word in a programming language. These cannot be used as identifiers in the program. (ii) Token: A token is the smallest element of a program that is meaningful to the compiler.	20 2M <i>1M each for correct definition</i>
	(b) Ans.	Give syntax for switch case statement. switch(variable) { case value1: statements break; case value2:	2M <i>Correct syntax</i> 2M



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		statements; break; default: statements; break; }	
(c) Ans.	Define array. State its types. Definition: An array is a data structure which can hold a number of values of the same data type. The different types of array are: <ul style="list-style-type: none">• one dimensional• two dimensional	2M <i>Definition 1M</i> <i>Types 1M</i>	
(d) Ans.	Define recursive function. When a function calls itself, it is called recursive function.	2M <i>Definition 2M</i>	
(e) Ans.	Give output for following code: <pre># include <stdio.h> main() { int a = 5, b = 6, *ptr1, *ptr2; ptr1 = &b; ptr2 = &a; printf("%d%d", *ptr1, *ptr2); }</pre> Output: 65	2M <i>Correct output 2M</i>	
(f) Ans.	List any four relational operators in 'C'. Relational operators: == equal to != Non equal to < less than > Greater than <= Less than equal to >= Greater than equal to	2M <i>List Any four relational operators 1/2M each</i>	
(g) Ans.	State one difference between the terms variable and constant. Variable: It is a data name that is used to store a data value. The values of variables can be changed in the program.	2M	



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in ‘C’

Subject Code: 17212

		<p>Constant: It is referred to as a fixed value that does not change while the program is under execution.</p>	<p><i>2M for correct difference</i></p>															
	<p>(h) Ans.</p>	<p>State the use of continue statement. Continue is used to force the next iteration of the loop to take place, skipping any code in between.</p>	<p>2M <i>2M for use</i></p>															
	<p>(i) Ans.</p>	<p>Write syntax and example for strcpy (). <i>(Note: Code snippet may be considered as example).</i> Syntax: strcpy(char[] dest, char[] source) Example: char source[]="mystring"; char dest[10]; strcpy(dest,source);</p>	<p>2M <i>Correct syntax 1M</i> <i>Example 1M</i></p>															
	<p>(j) Ans.</p>	<p>Define structure. A structure is a user defined data type in C. A structure creates a data type that can be used to group items of possibly different types into a single type. ‘struct’ keyword is used to create a structure</p>	<p>2M <i>2M for definition</i></p>															
	<p>(k) Ans.</p>	<p>Give anyone difference between while loop and do-while loop.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Sr. No.</th> <th style="width: 40%;">While loop</th> <th style="width: 50%;">do-while loop</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>In 'while' loop the controlling condition appears at the start of the loop.</td> <td>In 'do-while' loop the controlling condition appears at the end of the loop.</td> </tr> <tr> <td style="text-align: center;">2</td> <td>The iterations do not occur if, the condition at the first iteration, appears false.</td> <td>The iteration occurs at least once even if the condition is false at the first iteration.</td> </tr> <tr> <td style="text-align: center;">3</td> <td>It is an entry controlled loop</td> <td>It is an exit controlled loop</td> </tr> <tr> <td style="text-align: center;">4</td> <td>while(condition) { body }</td> <td>do { body }while(condition);</td> </tr> </tbody> </table>	Sr. No.	While loop	do-while loop	1	In 'while' loop the controlling condition appears at the start of the loop.	In 'do-while' loop the controlling condition appears at the end of the loop.	2	The iterations do not occur if, the condition at the first iteration, appears false.	The iteration occurs at least once even if the condition is false at the first iteration.	3	It is an entry controlled loop	It is an exit controlled loop	4	while(condition) { body }	do { body }while(condition);	<p>2M <i>Any one difference 2M</i></p>
Sr. No.	While loop	do-while loop																
1	In 'while' loop the controlling condition appears at the start of the loop.	In 'do-while' loop the controlling condition appears at the end of the loop.																
2	The iterations do not occur if, the condition at the first iteration, appears false.	The iteration occurs at least once even if the condition is false at the first iteration.																
3	It is an entry controlled loop	It is an exit controlled loop																
4	while(condition) { body }	do { body }while(condition);																
	<p>(l)</p>	<p>Give output for following code: # include <stdio.h> main ()</p>	<p>2M</p>															



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		<pre>{ int i=4; if(i= =5) printf("\n I am in Fy"); else printf("\n I am in Sy"); }</pre>	<i>2M for correct output</i>
2.	(a) Ans.	<p>Attempt any FOUR of the following:</p> <p>List various logical operators and describe use of each with suitable example.</p> <p>The logical operators are: &&-logical AND - logical OR ! – logical NOT</p> <p>&&-logical AND. Used when we want more than one condition to be checked and also the statements should be executed only if all the conditions are true.</p> <p> - logical OR - Used when we want more than one condition to be checked and also the statements should be executed if either of the conditions are true</p> <p>! – logical NOT – used we want to negate a condition.</p> <p>Eg : for &&(and) #include<stdio.h> #include<conio.h> void main() { int i,j; clrscr(); printf("Enter two values"); scanf("%d%d",&i,&j); if(i<=5 && j<=7) printf("i and j are less than 5"); getch(); }</p>	<p>16 4M</p> <p><i>1M for listing</i></p> <p><i>Use and any example of each logical operator 1M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<pre>Eg: for (or) #include<stdio.h> #include<conio.h> void main() { int i,j; clrscr(); printf("Enter two values"); scanf("%d%d",&i,&j); if(i<5 j<7) { printf("The values are%d%d",j,j); } getch(); } Eg: for !(Not) #include<stdio.h> #include<conio.h> void main() { int i,j; clrscr(); printf("Enter two values"); scanf("%d%d",&i,&j); if(!(i<=5 && j<=7)) { printf("the values are %d%d",i,j); } getch(); }</pre>	
(b)	<p>Describe importance of break statement in switch case statement. <i>(Note: Code snippet may also be considered).</i></p> <p>Ans. When break is encountered, the loop is terminated and the control goes to the next statement after the loop. Break is used in switch case to exit the switch-case after a specific match for the case is met with. If no case is matched, then the default statement is executed. If no break is used then once the case is matched, all the rest of the cases thereafter will get executed.</p> <p>Eg: #include<stdio.h> #include<conio.h></p>	4M <i>Relevant descripti on 4M</i>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

<pre>void main() { int var; clrscr(); printf("Enter a value"); scanf("%d", &var); switch(var) { case 1: printf("Number is less than 5"); case 2: printf("Number is less than 15"); default: printf("Number is more than 25"); } getch(); }</pre> <p>If the user inputs 1, the case matching the value 1 for var, is executed. However since there is no break, the statements corresponding to all the cases there after including the default will be displayed. If break is used, only the statement matching the particular case will get executed.</p> <p>Eg:</p> <pre>#include<stdio.h> #include<conio.h> void main() { int var; clrscr(); printf("Enter a value"); scanf("%d",&var); switch(var) { case 1: printf("\nNumber is less than 5"); break; case 2: printf("\nNumber is less than15"); break; default: printf("\nNumber is more than 25");</pre>	
--	--



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		<pre>break; } getch(); }</pre> <p>If we input the value 1 for var, for the above program, only Number less than 5 for the case 1 will get executed.</p>	
(c)	Write a 'C' program to find whether the entered number is prime or not. <i>(Note: Any other correct logic shall be considered).</i>		4M
Ans.	<pre>#include<stdio.h> #include<conio.h> void main() { int num, flag = 0, i; clrscr(); printf("Enter the number"); scanf("%d",&num); for(i = 2;i<num;i++) { flag = 0; if(num%i==0) { flag=0; break; } else { flag =1; } } if(flag==1) { printf("The number is prime"); } else { printf("The number is not prime"); } getch(); }</pre>	<p><i>Correct logic 2M</i></p> <p><i>Correct syntax 2M</i></p>	
(d)	With suitable example describe how to declare and initialize two dimensional array.		4M
Ans.	<p>A two dimensional array is an array of arrays. It consists of rows and columns. It is also called a matrix.</p> <p>To declare and initialize two dimensional array with 2 rows and 2 columns:</p>		



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<p>(e)</p> <p>Write a 'C' program to accept radius and calculate area of circle using function. <i>(Note: Any other correct logic shall be considered).</i></p> <p>Ans.</p>	<pre>#include<stdio.h> #include<conio.h> float cal_area(int r) { return 3.14*r*r; } void main() { int r; float ar; clrscr(); printf("Enter radius"); scanf("%d",&r); ar=cal_area(r); printf("%f",ar); getch(); }</pre>	<p>4M</p> <p><i>Correct syntax</i> 2M</p> <p><i>Correct logic</i> 2M</p>
	<p>(f)</p> <p>Write a 'C' program to count length of the string using pointer. <i>(Note: Any other correct logic shall be considered).</i></p> <p>Ans.</p>	<pre>#include<stdio.h> #include<conio.h> void main() { char ar[50]; char *s; int l=0; clrscr(); printf("enter a string"); scanf("%s",ar); s=ar; while(*s!='\0'){ l++; s++; } printf("Length of th string is %d",l); getch(); }</pre>	<p>4M</p> <p><i>Correct syntax</i> 2M</p> <p><i>Correct logic</i> 2M</p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

3.	(a) Ans.	<p>Attempt any FOUR of the following: Write on algorithm and draw a flow chart to find whether the entered number is even or odd.</p> <p>Algorithm: 1. start the program 2. Input a number 3. check whether number is divisible by 2 or not. if yes goto step 4 otherwise goto step 5 4. display that the numebr is even and go to step 6 5. display that the number is odd. 6. stop the program</p> <p>Flowchart:</p> <pre>graph TD; Start([Start]) --> Input[/Input Number/]; Input --> Decision{If Number % 2 == 0}; Decision -- Yes --> DisplayEven[/Display "Even Number"/]; Decision -- No --> DisplayOdd[/Display "Odd Number"/]; DisplayEven --> End([End]); DisplayOdd --> End;</pre>	16 4M <i>2M algorithm m</i> <i>2M flowchar t</i>
	(b) Ans.	<p>Describe with example in which case do-while loop is most suitable than while loop.</p> <p>Do while example is best suitable when at least one iteration is required, because it is an exit controlled loop, in the sense, the condition is checked at the end.</p> <p>Example: A menu driven program which shows options for result of any one arithmetic operation as per the selection:</p>	4M <i>Descript ion 2M</i>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<pre>#include<stdio.h> #include<conio.h> void main () { int a,b,ch=0;; clrscr(); printf("enter number 1"); scanf("%d",&a); printf("enter number 2"); scanf("%d",&b); do { printf("1. Add\n"); printf("2. Subtract\n"); printf("3. Exit\n"); printf("Enter your choice:"); scanf("%d",&ch); switch(ch) { case 1: { printf("Addition : %d\n",(a+b)); break; } case 2: { printf("Subtraction : %d\n",(a-b)); break; } default: { printf("Bye"); break; } }}while(ch<3); getch(); }</pre> <p>In the above example, menu will be displayed without checking any condition. Depending upon user's choice a case from switch will execute. If user wish to continue then while loop takes the control back to do statement.</p>	<p><i>Example 2M</i></p>
<p>(c)</p> <p>Ans.</p>	<p>Write a 'C' program to accept 10 numbers in an array and display the smallest number from them. <i>(Note: Any other correct logic shall be considered).</i></p>	<p>4M</p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<pre>#include<stdio.h> main() { int a[10]; int i,min=0; clrscr(); printf("Enter 10 integers"); for(i=0;i<10;i++) { scanf("%d",&a[i]); } min=a[0]; for(i=0;i<10;i++) { if(min>a[i]) min=a[i]; } printf("Minimum number: %d",min); getch(); }</pre>	<p><i>Correct logic 2M</i></p> <p><i>Correct syntax 2M</i></p>
<p>(d) Ans.</p>	<p>State categories of function. Describe any one with example.</p> <p>The different categories of functions are:</p> <ol style="list-style-type: none">1. Function without arguments without return type2. Function without arguments with return type3. Function with arguments without return type4. Function with arguments with return type <p>Function without arguments without return type: Here the function will not return any value and it will not have any argument.</p> <p>Example:</p> <pre>#include<stdio.h> #include<conio.h> void printNum(); void main() { printNum(); getch(); } void printNum()</pre>	<p>4M</p> <p><i>List of categories 2M</i></p> <p><i>Description of any one with example 2M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<pre>{ int i = 10; printf("%d",i); }</pre> <p>Function without arguments with return type: Here the function will return a value from the function but it will not have any arguments.</p> <p>Example:</p> <pre>#include<stdio.h> #include<conio.h> int printNum(); void main() { int i = printNum(); printf("%d",i); getch(); } int printNum() { int i = 10; clrscr(); return i; }</pre> <p>Function with argument without return type: Here the function takes values as arguments but it does not return any value</p> <p>Example:</p> <pre>#include<stdio.h> #include<conio.h> void printNum(int); void main() { int i = 10; clrscr(); printNum(i); getch(); } void printNum(int i)</pre>	
--	---	--



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<pre>{ printf("%d",i); }</pre> <p>Function with arguments with return type: Here the function takes values as arguments and returns value. Example:</p> <pre>#include<stdio.h> #include<conio.h> int printNum(int); void main() { int i = 10; int sq=0; clrscr(); sq = printNum(i); printf("%d",sq); getch(); } int printNum(int i) { int s = i*i; return s; }</pre>	
(e) Ans.	<p>Write a 'C' program to display fibonacci series using recursion. <i>(Note: Any other correct logic shall be considered).</i></p> <pre>#include<stdio.h> main() { int fibonaci(int); int n, i = 0, c; scanf("%d", &n); printf("Fibonacci series terms are:\n"); for (c = 1; c <= n; c++) { printf("%d\n", fibonaci(i)); i++; } getch();</pre>	4M <i>Correct logic 2M</i> <i>Correct syntax 2M</i>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		<pre>} int fibonaci(int n) { if (n == 0 n == 1) return n; else return (fibonaci(n-1) + fibonaci(n-2)); }</pre>	
	(f) Ans.	<p>Define the terms pointer and pointer to array. Also write two advantages of pointer.</p> <p>Definition :</p> <p>Pointer: Pointer is a variable that stores the address of another variable which is of similar data type. <i>Eg:</i> int i=3; int *ptr = &i; Here the address of i is stored in the pointer variable ptr.</p> <p>Pointer to array: An array name is a constant pointer to the first element of the array. <i>Eg</i> int arr[5], *p; then p=arr; Here p acts as pointer to array 'arr'.</p> <p>Advantages of using pointer:</p> <p>(i) It allows passing of arrays and strings to functions more efficiently.</p> <p>(ii) It makes possible to pass address of structure instead of entire structure to the functions.</p> <p>(iii) It makes possible to return more than one value from the function.</p> <p>(iv) It supports dynamic memory management</p>	4M <i>Definitio n pointer 1M</i> <i>Definitio n pointer to array 1M</i> <i>Any two advanta ges 2M</i>
4.	(a) Ans.	<p>Attempt any FOUR of the following:</p> <p>Describe with suitable example how to use increment and decrement operators.</p> <p>Increment operator: Increment operator (++) is a unary operator. It operates on one operand. It is used to add one to an existing value of variable.</p>	16 4M



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<p>Syntax: variable_name++ or ++variable_name</p> <p>Example: int num=6; printf("%d",num); num++; printf("\n%d",num); In above example initially value of num is 6. Due to increment operator (++) value of variable num will become 7.</p> <p>Decrement operator: Decrement operator(--) is an unary operator. It operates on one operand. It is used to subtract one from an existing value of variable.</p> <p>Syntax: variable_name-- or --variable_name</p> <p>Example: int num=5; printf("%d",num); num--; printf("\n%d",num); In above example initially value of num is 5. Due to decrement operator (--) value of num will become 4.</p>	<p><i>Explanation of operators 1M each</i></p> <p><i>Example of operator 1M each</i></p>
<p>(b)</p> <p>Ans.</p>	<p>Write a 'C' program to enter any number between 1 to 7 and display its corresponding week day using switch case statement. (Note: Any other correct logic shall be considered).</p> <pre>#include<stdio.h> main() { int n; clrscr(); printf("enter any number between 1 to 7 :"); scanf("%d",&n); switch(n) { case 1: printf("Monday"); break; case 2: printf("Tuesday"); break; case 3:</pre>	<p>4M</p> <p><i>Correct logic 2M</i></p> <p><i>Correct syntax 2M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<pre>printf("Wednesday"); break; case 4: printf("Thursday"); break; case 5: printf("Friday"); break; case 6: printf("Saturday"); break; case 7: printf("Sunday"); break; default: printf("Invalid number"); } getch(); }</pre>	
(c) Ans.	<p>Describe following functions with its syntax and example: (i) strcmp() (ii) strlen() (i) strcmp() : It is a string function, which 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. Syntax: strcmp(string1,string2); Eg: if s1="there" and s2="their" the output of strcmp(s1,s2) will be 9 as the difference between ascii values of 'r' and 'i' is 9.</p> <p>(ii) strlen() : strlen() is a string function which is used to find length of the string. Syntax : strlen(string) Eg:</p>	4M <i>Description, Syntax, example of each 2M</i>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		If str contains "abcd" as its contents, strlen(str) will return 4 as length of str.	
(d)	Write a program in 'C' to define a structure 'Person' with structure members as name and age. Accept this data for one person and display the same. (Note: Any other correct logic shall be considered).		4M
Ans.	<pre>#include<stdio.h> struct Person { char name[20]; int age; }; main() { struct Person p; clrscr(); //accept data printf("Enter name of the person:"); scanf("%s",p.name); printf("Enter age of the person:"); scanf("%d",&p.age); //Display data printf("Name of the person : %s\n",p.name); printf("Age of the person : %d",p.age); getch(); }</pre>	<i>Correct logic 2M</i> <i>Correct syntax 2M</i>	
(e)	Describe the concept of command line argument with example.		4M
Ans.	<p>Command line arguments are given after the name of the program in command-line shell of Operating System.</p> <p>The command line arguments are handled using main() function arguments where argc refers to the number of arguments passed, and argv[] is a pointer array which points to each argument passed to the program.</p> <p>Example:</p> <pre>#include <stdio.h> void main(int argc, char *argv []) { printf(" \n Name of my Program %s \t", argv[0]);</pre>	<i>Description 2M</i> <i>Example 2M</i>	



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		<pre>if(argc == 2) { printf("\n Value given by user is: %s \t", argv[1]); } else if(argc > 2) { printf("\n Many values given by users.\n"); } else { printf(" \n Single value expected.\n"); } }</pre>	
	(f) Ans.	<p>Write a program in 'c' to exchange values of two numbers using pointer. <i>(Note: Any other correct logic shall be considered).</i></p> <pre>#include<stdio.h> void main() { void swap(int*, int*); int a,b; clrscr(); a=8;b=10; printf("before swap : a=%d, b=%d\n",a,b); swap(&a,&b); printf("After swap: a=%d, b=%d",a,b); getch(); } void swap(int *p1,int *p2) { int temp; temp=*p1; *p1=*p2; *p2=temp; }</pre>	4M <i>Correct logic 2M</i> <i>Correct syntax 2M</i>
5.	(a) Ans.	<p>Attempt any FOUR of the following: Describe conditional operator with syntax and example. Conditional operator is represented with ternary operator pair"?. It is uses to evaluate conditional expression.</p>	16 4M <i>Description 2M</i>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<p>Syntax: exp1? exp2 : exp 3 ; In the syntax , exp1, exp2 and exp3 are expressions. exp1 is evaluated first. If it is true, then expression exp2 is evaluated and resultant value of this expression is the output of conditional statement. If exp1 is false then exp3 is evaluated and its resultant value is the output of conditional statement.</p> <p>Example: int a=10,b=5,x; x=(a>b) ? a : b; In the above example x will take value 10 because given condition a>b is true.</p>	<p>Syntax 1M</p> <p>Example 1M</p>
(b) Ans.	<p>Write a 'C' program to display cube of 1 to 10 numbers using loop. (Note: Any other correct logic shall be considered).</p> <pre>#include<stdio.h> #include<conio.h> void main() { int i; clrscr(); for(i=1;i<=10;i++) printf("\n Cube of %d=%d",i,i*i*i); getch(); }</pre>	<p>4M</p> <p>Correct syntax 2M</p> <p>Correct logic 2M</p>
(c) Ans.	<p>Write a program to find whether the year is leap or not. (Note: Any other correct logic shall be considered).</p> <pre>#include<stdio.h> #include<conio.h> void main() { int year; clrscr(); printf("Enter year:"); scanf("%d",&year); if(year%4==0) printf("\n Year is leap year"); else printf("\n Year is not a leap year"); getch(); }</pre>	<p>4M</p> <p>Correct syntax 2M</p> <p>Correct logic 2M</p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<p>(d)</p> <p>Ans.</p>	<p>Write a 'C' program to find whether the character entered is a alphabet, digit or special character. <i>(Note: Any other correct logic shall be considered).</i></p> <pre>#include<stdio.h> #include<conio.h> int main() { char ch; printf("Enter any character: "); scanf("%c", &ch); if((ch >= 'a' && ch <= 'z') (ch >= 'A' && ch <= 'Z')) { printf("%c' is alphabet.", ch); } else if(ch >= '0' && ch <= '9') { printf("%c' is digit.", ch); } else { printf("%c' is special character.", ch); } return 0; }</pre>	<p>4M</p> <p><i>Correct syntax 2M</i></p> <p><i>Correct logic 2M</i></p>									
	<p>(e)</p> <p>Ans.</p>	<p>Differentiate between call by value and call by reference methods. (any four points)</p> <table border="1" data-bbox="391 1434 1284 1873"><thead><tr><th data-bbox="391 1434 475 1507">Sr. No.</th><th data-bbox="475 1434 834 1507">Call by value</th><th data-bbox="834 1434 1284 1507">Call by reference</th></tr></thead><tbody><tr><td data-bbox="391 1507 475 1654">1</td><td data-bbox="475 1507 834 1654">A copy of actual arguments(value) is passed to respective formal arguments.</td><td data-bbox="834 1507 1284 1654">Address of actual arguments is passed to formal arguments</td></tr><tr><td data-bbox="391 1654 475 1873">2</td><td data-bbox="475 1654 834 1873">Actual arguments will remain safe, they cannot be modified accidentally.</td><td data-bbox="834 1654 1284 1873">Alteration to actual arguments is possible within from called function; therefore the code must handle arguments carefully else you get unexpected results.</td></tr></tbody></table>	Sr. No.	Call by value	Call by reference	1	A copy of actual arguments(value) is passed to respective formal arguments.	Address of actual arguments is passed to formal arguments	2	Actual arguments will remain safe, they cannot be modified accidentally.	Alteration to actual arguments is possible within from called function; therefore the code must handle arguments carefully else you get unexpected results.	<p>4M</p> <p><i>Any four points 1M each</i></p>
Sr. No.	Call by value	Call by reference										
1	A copy of actual arguments(value) is passed to respective formal arguments.	Address of actual arguments is passed to formal arguments										
2	Actual arguments will remain safe, they cannot be modified accidentally.	Alteration to actual arguments is possible within from called function; therefore the code must handle arguments carefully else you get unexpected results.										



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		3	Address of the actual and formal arguments are different	Address of the actual and formal arguments are the same	
		4	Changes made inside the function are not reflected in other functions	Changes made in the function are reflected outside also.	
	(f)	Describe addition and subtraction operations on pointer. Give suitable example for each.			4M
	Ans.	<p>In C, pointer holds address of a value, so there can be arithmetic operations on the pointer variable. Following arithmetic operations are possible on pointer in C language:</p> <ol style="list-style-type: none">1. Increment/addition2. Decrement/subtraction <p>1. Increment: It is used to increment the pointer. Each time a pointer is incremented, it points to the 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.</p> <p style="text-align: center;">OR</p> <p>1. Addition When addition operation is performed on pointer, it gives the location incremented by the added value according to data type. Eg: If ptr is an integer pointer stored at address 1000, Then ptr+2 shows $1000+(2*2) = 1004$ as incremented location for an int.</p> <p>2. 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. Example,</p>			<p><i>Descript ion with example of addition 2M</i></p> <p><i>Subtract ion 2M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		<p>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.</p> <p style="text-align: center;">OR</p> <p>2. 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.</p>	
6.	(a) Ans.	<p>Attempt any FOUR of the following: Write a 'C' program to accept the marks of three subjects and display total marks and average marks. <i>(Note: Any other correct logic shall be considered).</i></p> <pre>#include<stdio.h> #include<conio.h> void main() { float m1,m2,m3,total,avg; clrscr(); printf("Enter marks of three subjects:"); scanf("%f%f%f",&m1,&m2,&m3); total=m1+m2+m3; avg=total/3; printf("\n Total=%f",total); printf("\n Average=%f",avg); getch(); }</pre>	<p>16 4M</p> <p><i>Acceptin g marks</i> 1M</p> <p><i>Calculat e total, average</i> 2M</p> <p><i>Display total, average</i> 1M</p>
	(b) Ans.	<p>Describe use of else-if ladder with suitable example. else if ladder is used to take a multipath decision. It is used in a program when there are more than one conditions are involved. Conditions are evaluated from the top to the bottom. As soon as the true condition is found, the statement associated with it is executed and the control is transferred to the statement-x. When all the conditions become false, then the final else containing the default statement will be executed.</p>	<p>4M</p> <p><i>Descript ion of use</i> 2M</p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<p>Example: int per=65; if(per>=75) printf("Distinction Class"); else if(per<75 && per>=60) printf("First Class"); else if(per<60 && per>=45) printf("Second Class"); else if(per<45 && per>=35) printf("Pass Class"); else printf("Fail");</p> <p>In the above example, per variable is initialize to 65. In else if ladder first condition is checked with value of per. The condition is false so control moves to second condition. Second condition is true so its associated statement is executed and control passes out of ladder.</p>	<p><i>Example 2M</i></p>
<p>(c)</p> <p>Write a 'C' program to enter a string and a character. Count number of times that character appears in entered string and display the count. <i>(Note: Any other correct logic shall be considered).</i></p> <p>Ans.</p>	<pre>#include<stdio.h> #include<conio.h> void main() { char str[10],ch; int i=0,count=0; clrscr(); printf("Enter string:"); gets(str); printf("Enter character:"); scanf("%c",&ch); while(str[i]!='\0') { if(str[i]==ch) { count++; } }</pre>	<p>4M</p> <p><i>Acceptin g string and characte r 1M</i></p> <p><i>Countin g logic 2M</i></p> <p><i>Display count 1M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		<pre>i++; } printf("\n Count=%d",count); getch(); }</pre>	
(d)	Write a 'C' program to define a structure 'cricket' having structure members as playername and battingaverage. Accept and display data for three players. (Note: Any other correct logic shall be considered).		4M
Ans.	<pre>#include<stdio.h> #include<conio.h> struct cricket { char playername[10]; int battingaverage; }c[3]; void main() { int i; clrscr(); for(i=0;i<3;i++) { printf("\n Enter playername:"); scanf("%s",c[i].playername); printf("\n Enter batting average:"); scanf("%d",&c[i].battingaverage); } for(i=0;i<3;i++) { printf("\n Player Name=%s",c[i].playername); printf("\n Batting Average=%d",c[i].battingaverage); } getch(); }</pre>	<p><i>Declaring structure 2M</i></p> <p><i>Accepting player data 1M</i></p> <p><i>Displaying player data 1M</i></p>	
(e)	Describe auto and extern storage classes with example.		4M
Ans.	Automatic variables: These are declared inside a function in which they are to be used. They are created when a function is called and destroyed when the function completes its execution. They are private		



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

	<p>to the function. Therefore these variables are also known as local or internal variables. To declare automatic variables explicitly the keyword auto can be used. The values of automatic variables defined in a function cannot be changed by some other function.</p> <p>Eg: void main() { auto int a; a=10; printf(“%d”,a); }</p> <p>External variables: these variables are active and alive throughout the entire program. These are also known as global variables. These variables can be accessed by any function in the program. External variables are declared outside a function. In case a local variable and global variable has the same name, the local variable will have preference over the global variable. The value of a global variable can be changed by any function, the subsequent functions will refer to the new value.</p> <p>Eg: int number; void main() { number=10; printf(“%d”,number); } void function1() { number=20; printf(“%d”,number); }\</p>	<p><i>Description with example of each class 2M</i></p>
<p>(f) Ans.</p>	<p>Describe use of for loop with its syntax and example. for loop is used to execute statement or group of statements repeatedly. It is an entry controlled loop that combines three steps such as initialization, condition and increment/decrement.</p> <p>Syntax: for(initialization; condition; increment/decrement) { Statements; }</p>	<p>4M</p> <p><i>Description 2M</i></p> <p><i>Syntax 1M</i></p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Programming in 'C'

Subject Code: 17212

		<p>In for loop, initialization of index variable is done first. In condition checking, value of index variable is checked. If condition is true then control enters into the loop and executes loop statements. After every iteration, value of index variable increments or decrements by one. Then control passes to condition again. Loop executes till the condition is true. Once the condition becomes false, control comes out of for loop.</p> <p>Example: int i; for(i=1;i<=5;i++) { printf("%d",i); }</p> <p>In the above example, variable i is a index variable initialized to 1.printf () statement inside loop executes 5 times i.e. till the condition is true. After each iteration value of index variable i increments by one. When value of i becomes 6, condition in for loop becomes false and control comes out of loop.</p>	<p><i>Example 1M</i></p>
--	--	--	------------------------------