



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

WINTER – 2018 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	Solve any TEN: Define algorithm and flowchart. Algorithm: - Algorithm is a stepwise set of instructions written to perform a specific task. Flowchart: - Flowchart is a diagrammatic representation of steps/procedure to perform a specific task.	20 2M Each definition 1M
	(b) Ans	State the use of scanf statement and write its syntax. scanf() is used to accept input from keyboard by the user. Syntax: scanf(“control string/format specifier”, arg1, arg2..);	2M Use 1M Syntax 1M
	(c) Ans	State the use of break statement with its syntax The break statement is used to exit from loop/block. When break statement is executed inside any loop or with switch case, control automatically passes to the first statement outside the loop. Syntax: break;	2M Use 1M Syntax 1M



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	(d) Ans	Write syntax of do-while loop Syntax: do { body of the loop }while(test-condition);	2M <i>Correct Syntax 2M</i>
	(e) Ans	What is an array? Definition: An array is a collection of similar type of elements. Syntax: data_type array_variable_name[size];	2M <i>Correct Definitio n 2M</i>
	(f) Ans	Give syntax of strcat () string function. Syntax: strcat (string1, string2);	2M <i>Correct syntax 2M</i>
	(g) Ans	State four relational operators with their meaning. Operator Meaning > Greater than < Less than >= Greater than or equal to <= Less than or equal to == Equals to != Not equal to	2M <i>Any four operator s ½M each</i>
	(h) Ans	What do you mean by recursion? Recursion means a function calls itself repetitively. A recursive function contains a function call to itself inside its body.	2M <i>Correct definitio n 2M</i>
	(i) Ans	Explain the use of ampersand (&) operator in pointers. An ampersand (&) operator is used as address of operator. It is used to return the memory location (address) of a variable. Consider, var is a variable. Use of ampersand operator with var variable (&var) returns its address from the memory.	2M <i>Explana tion of Use 2M</i>
	(j) Ans	Give the syntax of nested if-else statement. Syntax for Nested if else statement: if(test condition1) { if(test condition2) { statement-1;	2M <i>Correct syntax 2M</i>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

		<pre> } else { statement-2; } } else { statement-3; } statement-x; </pre>																
	<p>(k) Ans</p>	<p>Write any two advantages of pointers. Advantages of pointers:- 1. Reduces the storage space and complexity of the program. 2. Reduces the execution time of the program. 3. Provides an alternate way to access array elements. 4. Pointers allow us to resize the dynamically allocated memory block. 5. Addresses of objects can be extracted using pointers. 6. Increase access speed. 7. Pointers permit references to functions and thereby facilitate passing of functions as arguments to other functions.</p>	<p style="text-align: center;">2M</p> <p style="text-align: center;"><i>Any two advantages 1M each</i></p>															
	<p>(l) Ans</p>	<p>State any two differences between if and switch statement.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">S. No</th> <th style="width: 40%;">switch statement</th> <th style="width: 50%;">if statement</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>Switch case can be used when there are multiple options and only one is to be selected.</td> <td>If statement checks any number of conditions and takes the action for true case or false case.</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Switch statement operates only one quality of values.</td> <td>If works even with the other comparison operators like >, <, >=, <=</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Switch has a default case.</td> <td>If does not have any provision to handle default case.</td> </tr> <tr> <td style="text-align: center;">4</td> <td>Every case needs a break statement to come out of switch block once a condition is matched.</td> <td>No such requirement in case of “if” statement.</td> </tr> </tbody> </table>	S. No	switch statement	if statement	1	Switch case can be used when there are multiple options and only one is to be selected.	If statement checks any number of conditions and takes the action for true case or false case.	2	Switch statement operates only one quality of values.	If works even with the other comparison operators like >, <, >=, <=	3	Switch has a default case.	If does not have any provision to handle default case.	4	Every case needs a break statement to come out of switch block once a condition is matched.	No such requirement in case of “if” statement.	<p style="text-align: center;">2M</p> <p style="text-align: center;"><i>Any 2 correct differences 1M each</i></p>
S. No	switch statement	if statement																
1	Switch case can be used when there are multiple options and only one is to be selected.	If statement checks any number of conditions and takes the action for true case or false case.																
2	Switch statement operates only one quality of values.	If works even with the other comparison operators like >, <, >=, <=																
3	Switch has a default case.	If does not have any provision to handle default case.																
4	Every case needs a break statement to come out of switch block once a condition is matched.	No such requirement in case of “if” statement.																



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

2	(a) Ans	Solve any FOUR: Distinguish between variables and constants. <table border="1"><thead><tr><th data-bbox="402 527 475 600">Sr. No</th><th data-bbox="475 527 857 600">Variables</th><th data-bbox="857 527 1276 600">Constants</th></tr></thead><tbody><tr><td data-bbox="402 600 475 674">1</td><td data-bbox="475 600 857 674">Variables can change their value</td><td data-bbox="857 600 1276 674">constants can never change their value</td></tr><tr><td data-bbox="402 674 475 747">2</td><td data-bbox="475 674 857 747">Variables can be initialized after declaration.</td><td data-bbox="857 674 1276 747">Constant have to be initialized at the time of declaration.</td></tr><tr><td data-bbox="402 747 475 894">3</td><td data-bbox="475 747 857 894">Variables on the other hand represent unknown values.</td><td data-bbox="857 747 1276 894">Constants usually represent known values in an equation, expression or in line of programming.</td></tr><tr><td data-bbox="402 894 475 968">4</td><td data-bbox="475 894 857 968">Syntax: Data_type variable_name;</td><td data-bbox="857 894 1276 968">Syntax: Using #define Constant_name value</td></tr><tr><td data-bbox="402 968 475 1115">5</td><td data-bbox="475 968 857 1115">Example: int Add; float radius; char name[10];</td><td data-bbox="857 968 1276 1115">Example: # define Pi=3.14</td></tr><tr><td data-bbox="402 1115 475 1339">6</td><td data-bbox="475 1115 857 1339">Types are 1. local variable 2. global variable 3. static variable 4. external variable 5. Automatic variable</td><td data-bbox="857 1115 1276 1339">Types are 1. Primary Constants 2. Secondary Constants</td></tr></tbody></table>	Sr. No	Variables	Constants	1	Variables can change their value	constants can never change their value	2	Variables can be initialized after declaration.	Constant have to be initialized at the time of declaration.	3	Variables on the other hand represent unknown values.	Constants usually represent known values in an equation, expression or in line of programming.	4	Syntax: Data_type variable_name;	Syntax: Using #define Constant_name value	5	Example: int Add; float radius; char name[10];	Example: # define Pi=3.14	6	Types are 1. local variable 2. global variable 3. static variable 4. external variable 5. Automatic variable	Types are 1. Primary Constants 2. Secondary Constants	16 4M Any four differences 1M each
Sr. No	Variables	Constants																						
1	Variables can change their value	constants can never change their value																						
2	Variables can be initialized after declaration.	Constant have to be initialized at the time of declaration.																						
3	Variables on the other hand represent unknown values.	Constants usually represent known values in an equation, expression or in line of programming.																						
4	Syntax: Data_type variable_name;	Syntax: Using #define Constant_name value																						
5	Example: int Add; float radius; char name[10];	Example: # define Pi=3.14																						
6	Types are 1. local variable 2. global variable 3. static variable 4. external variable 5. Automatic variable	Types are 1. Primary Constants 2. Secondary Constants																						
	(b) Ans	Write a program to display following pattern: 1 1 2 1 2 3 1 2 3 4 #include <stdio.h> #include <conio.h> void main() { int i,j; clrscr(); for(i=1 ; i<=4 ; i++) { for(j=1 ; j<=i ; j++) {	4M Correct logic 2M Correct Syntax 2M																					



WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<pre>printf("%d \t", j); } printf("\n"); } getch(); }</pre>	
(c) Ans	<p>What is array of structure? Explain with example.</p> <p>Array of structure:- An array of structure in C is a collection of multiple structure variables where each variable contains information about different entities. An array of structure is used to store information about multiple entities of different data types. The array of structures is also known as the collection of structures.</p> <pre>#include<stdio.h> #include <string.h> struct student { int rollno; char name[10]; }; int main() { int i; struct student st[5]; printf("Enter Records of 5 students"); for(i=0;i<5;i++){ printf("\nEnterRollno:"); scanf("%d",&st[i].rollno); printf("\nEnter Name:"); scanf("%s",&st[i].name); } printf("\nStudent Information List:"); for(i=0;i<5;i++){ printf("\nRollno:%d, Name:%s",st[i].rollno,st[i].name); } return 0; }</pre> <p>In above example array of structure student is st having 5 members in it. Five student's data is kept in structure student.</p>	4M <i>Description of array of structure</i> 2M <i>Example</i> 2M



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<p>(d) Ans</p>	<p>Explain strlen() and strcmp() string functions strlen():-This library function is used to count the length of the string i.e. number of characters in a string including blank spaces. Syntax :strlen(string1); strcmp():- This function compares two strings identified by arguments and returns zero if both strings are equal, otherwise it returns the difference between ASCII values of first non-matching character pair from the strings. Syntax: strcmp (string1, string2); strcmp (“there”, “their”) returns value as 9 which is the difference between “r” & “i”.</p>	<p>4M <i>Explanation of strlen</i> 2M & <i>strcmp</i> 2M</p>
	<p>(e) Ans</p>	<p>Write a program to calculate factorial of a number using recursion. #include<stdio.h> #include<conio.h> int factorial(int n); void main() { int n,fact; clrscr(); printf("enter the number"); scanf("%d",&n); fact=factorial(n); printf("factorial of %d=%d",n,fact); getch(); } int factorial(int n) { if(n==1) { return(1); } else { return(n * factorial(n-1)); } }</p>	<p>4M <i>Main () definitio</i> n 2M <i>Recursive function definitio</i> n 2M</p>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<p>(f)</p> <p>Define pointer. State the syntax to declare pointer variable with example.</p> <p>Ans Pointer is a variable used to store the memory address of another variable of similar data type. Variables store the values and pointers stores their addresses. The values stored in the pointers are integer values.</p> <p>Pointer declaration: In declaration statement of pointer, name of pointer variable is preceded by * (indirection operator) operator.</p> <p>Syntax:- data_type * name_of_variable;</p> <p>Example: int *ptr ; Here ptr variable of data type integer is declared as a pointer. Name of variable “ptr” is preceded by * (indirection operator) means that variable ptr is a pointer variable.</p>	<p>4M</p> <p><i>Definition</i> n 2M,</p> <p><i>Syntax of pointer variable</i> 1M,</p> <p><i>Example</i> 1M</p>
3	<p>(a)</p> <p>Write a program to display Fibonacci series upto given number using function.</p> <p>Ans <i>Note: Any other correct logic shall be considered.</i></p> <pre>#include<stdio.h> #include<conio.h> void findfib() { int n,t1=1,t2=1; clrscr(); printf("Enter the last number"); scanf("%d",&n); printf("%d",t1); while(t2<=n) { printf("\n%d",t2); t2=t1+t2; t1=t2-t1; } } void main() { findfib(); getch(); }</pre>	<p>16 4M</p> <p><i>Correct logic</i> 2M</p> <p><i>Correct syntax</i> 2M</p>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<p>(b) Ans</p>	<p>State the use of increment and decrement operator with example.</p> <p>Increment operator: Increment operator is used to increment or increase the value of a variable by one. It is equivalent to adding one to the value of the variable. The symbol used is ++.</p> <p>Syntax: ++var or var++</p> <p>Example:</p> <pre>int m=5; int n = ++m; printf("%d%d",m,n);</pre> <p>Decrement operator: Decrement operator is used to decrement or decrease the value of variable by one. It is equivalent to subtracting one from the value of the variable. The symbol used is --.</p> <p>Syntax: --var or var--</p> <p>Example for decrement operator</p> <pre>int m=5; int n=--m; printf("%d%d",m,n);</pre>	<p>4M</p> <p><i>Explanation of use of each operator with example</i> 2M</p>
	<p>(c) Ans</p>	<p>Write a program to find reverse of a string <i>Note: Any other correct logic shall be considered</i></p> <pre>#include<stdio.h> #include<conio.h> #include<string.h> void main() { char str[100],temp; int i,j=0; clrscr(); printf("Enter a string"); scanf("%s",&str); printf("Entered string is %s",str); i=0; j=strlen(str)-1; while(i<j) { temp=str[i]; str[i]=str[j];</pre>	<p>4M</p> <p><i>Correct logic</i> 2M</p> <p><i>Correct syntax</i> 2M</p>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

		<pre>str[j]=temp; i++; j--; } printf("\nReverse of string is %s",str); getch(); }</pre>	
(d) Ans	<p>Explain use of if-else statement with example.</p> <p>If-else is used for decision making. When if-else is used, the program branches and no longer go with the sequential execution. The statements that are executed depends upon whether the condition used in the program is true or false.</p> <p>If condition returns true then the statements inside the body of “if” are executed and the statements inside body of “else” are skipped. If condition returns false then the statements inside the body of “if” are skipped and the statements in “else” are executed.</p> <p>General syntax:</p> <pre>if(condition) { //body of if } else { //body of else }</pre> <p>Example:</p> <pre>#include<stdio.h> #include<conio.h> void main() { int no; clrscr(); printf("Enter a number"); scanf("%d",&no); if(no<5) { printf("The number is less than 5"); }else { printf("the number is equal to or more than 5"); } getch(); }</pre>	4M <i>Explanation</i> 2M <i>Example</i> 2M	



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<p>(e)</p> <p>Write a program to search an element in an array of 10 numbers using function.</p> <p><i>Note: Any other correct logic shall be considered</i></p> <p>Ans</p>	<pre>#include<stdio.h> #include<conio.h> void findNum() { int a[10]; int i,no, flag=0; clrscr(); for(i=0;i<10;i++) { printf("Enter no"); scanf("%d",&a[i]); } printf("Enter a number to search"); scanf("%d",&no); for(i=0;i<10;i++) { if(a[i]==no) { printf("Number %d is present in the position %d in the array",no,i); flag=0; break; } else { flag=1; } } if(flag==1){ printf("Number is not present"); } } void main() { findNum(); getch(); }</pre>	<p>4M</p> <p><i>Correct Logic</i> 2M</p> <p><i>Correct Syntax</i> 2M</p>
--	---	--	---



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

		<pre>printf("Enter a valid option"); break; } getch(); }</pre>	
4	(a) Ans	Solve any FOUR: State rules for choosing variable name. <ul style="list-style-type: none">• A variable name is any combination of 1 to 31 alphabets, digits or underscores.• Variables names must start with alphabet or underscore.• No other special character, except underscore, is allowed in the variable name.• Blank spaces or white spaces are not allowed in the variable name• Variable name should not be a reserved keyword.	16 4M <i>Any valid four Rules 1M each</i>
	(b) Ans	Write a program to declare a structure book having data members, title, author and price. Accept data and display information for one book. <pre>#include<stdio.h> #include<conio.h> void main() { struct book { char title[50]; char author[50]; float price; }b; clrscr(); printf("Enter title, author and price of one book"); scanf("%s%s%f",&b.title,&b.author,&b.price); printf("The details of book are:\nTitle-%s\nauthor-%s\nprice-%f",b.title,b.author,b.price); getch(); }</pre>	4M <i>Structure declaration-2M</i> <i>Accept data 1M</i> <i>Display data 1M</i>
	(c) Ans	Write a program to accept 10 numbers in an array and print average of it. <pre>#include<stdio.h> #include<conio.h> void main() { int arr[10];</pre>	4M <i>Correct Logic 2M</i>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

		<pre>int i; int sum=0,avg=0; clrscr(); for(i = 0;i<10;i++) { printf("Enter number"); scanf("%d",&arr[i]); } for(i=0;i<10;i++) { sum=sum+arr[i]; } printf("sum is %d",sum); avg = sum/10; printf("Average of 10 numbers is %d",avg); getch(); }</pre>	<i>Correct Syntax 2M</i>
(d) Ans	<p>Explain pointer arithmetic with example.</p> <p>The pointer arithmetic is done as per the data type of the pointer. The basic operations on pointers are:</p> <p>Increment</p> <p>It is used to increment the pointer. Each time a pointer is incremented, it points to the next location. Eg, for an int pointer variable, if the current position of pointer is 1000, when incremented it points to 1002 because for storing an int value it takes 2 bytes of memory.</p> <p>Decrement</p> <p>It is used to decrement the pointer. Each time a pointer is decremented, it points to the previous location. Eg, if the current position of pointer is 1002, then decrement operation results in the pointer pointing to the location 1000.</p> <p>Addition and subtraction</p> <p>When addition or subtraction operation is performed on the pointer variable, it shows that particular location in the memory. Eg: int *ptr; -say address is 1000. If ->ptr+n- then ptr+n*2 . If ->ptr-n then ptr-n*2.</p>	<i>4M Any two arithmetic operator s with pointer Explan ation and example of each 2M</i>	



WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

		<p>Comparison operators Pointers may be compared by using relational operators, such as ==, <, and >. If p1 and p2 point to variables that are related to each other, such as elements of the same array, then p1 and p2 can be compared using the comparison operators.</p> <p>Example:</p> <pre>#include<stdio.h> #include<conio.h> void main() { int i = 10; int *ptr=&i; clrscr(); printf("%x%d",ptr,i); ptr++; printf("\n%x",ptr); printf("\n%x",ptr+2); printf("\n%x",ptr-2); getch(); }</pre>											
(e) Ans	<p>Differentiate between while and do-while loop</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">While</th> <th style="width: 50%; text-align: center;">Do..while</th> </tr> </thead> <tbody> <tr> <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>The iterations do not occur if, the condition is false.</td> <td>The iteration occurs at least once even if the condition is false at the first iteration.</td> </tr> <tr> <td>It is an entry controlled loop</td> <td>It is an exit controlled loop</td> </tr> <tr> <td>Syntax : while(condition) { Code; }</td> <td>Syntax: do { Code; } while(condition);</td> </tr> </tbody> </table>		While	Do..while	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.	The iterations do not occur if, the condition is false.	The iteration occurs at least once even if the condition is false at the first iteration.	It is an entry controlled loop	It is an exit controlled loop	Syntax : while(condition) { Code; }	Syntax: do { Code; } while(condition);	<p>4M</p> <p><i>Any four relevant difference 1M each</i></p>
While	Do..while												
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.												
The iterations do not occur if, the condition is false.	The iteration occurs at least once even if the condition is false at the first iteration.												
It is an entry controlled loop	It is an exit controlled loop												
Syntax : while(condition) { Code; }	Syntax: do { Code; } while(condition);												



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<p>(f)</p> <p>Write a program to print sum of all odd numbers between 1-100 <i>Note: Any other correct logic shall be considered</i></p> <p>Ans</p> <pre>#include<stdio.h> #include<conio.h> void main() { int i; int sum=0; clrscr(); for(i=1;i<=100;i++) { if(i%2!=0) { sum=sum+i; } } printf("Sum is %d",sum); getch(); }</pre>	<p>4M</p> <p><i>Correct Logic</i> 2M</p> <p><i>Correct Syntax</i> 2M</p>
5	<p>(a)</p> <p>Solve any FOUR: Write a program to find sum of digits of an integer e.g. (input = 1234, sum= 1+2+3+4 = 10)</p> <p>Ans</p> <pre>#include<stdio.h> void main() { int num,q,r,sum=0; clrscr(); printf("Enter number :"); scanf("%d",&num); while(num!=0) { r=num%10; q=num/10; sum=sum+r; num=q; } printf("Sum=%d",sum); getch(); }</pre>	<p>16</p> <p>4M</p> <p><i>Input number</i> 1M</p> <p><i>Digit separation and summation logic</i> 2M</p> <p><i>Display of sum</i> 1M</p>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<p>(b) Ans</p>	<p>Write a program to print largest of three numbers. <i>Note: Any other correct logic shall be considered.</i></p> <pre>#include<stdio.h> void main() { int a,b,c; clrscr(); printf("Enter 3 numbers:"); scanf("%d %d %d",&a,&b,&c); if(a>b && a>c) printf("%d is largest",a); if(b>c && b>a) printf("%d is largest",b); if(c>b && c>a) printf("%d is largest",c); getch(); }</pre>	<p>4M</p> <p><i>Input 2 numbers 1M</i></p> <p><i>Conditions to check greater number 2M</i></p> <p><i>Display output 1M</i></p>
	<p>(c) Ans</p>	<p>Explain any two storage classes.</p> <p>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 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><i>Eg:</i></p> <pre>void main() { auto int a; a=10; printf("%d",a); }</pre> <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</p>	<p>4M</p> <p><i>Explanation of any two storage classes 2M each</i></p>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<p>new value.</p> <p>Example: int number; void main() { number=10; printf(“%d”,number); } void function1() { number=20; printf(“%d”,number); }</p> <p>Static variables: the value of the static variable persists until the end of the program execution .A variable can be declared as a static using the keyword static. Static variable can be an external or an internal variable. Internal static variable are those who are declared inside a function, but they remain alive throughout the execution of the program. The static variable is initialized only once when the program is compiled.</p> <p>Example: void func1() { static int x=0; x= x+1; printf(“x=%d”,x); } void main() { int i; for(i=0;i<3;i++) { func1(); } }</p> <p>Register variables: these variables are stored in the registers instead of memory. Since the register access is much faster compared to the memory, frequently used variables can be stored this way.</p>	
--	---	--



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

		Example: void main() { register int count=0; count++; printf(“%d”,count); }	
(d) Ans	Explain formatted input and output Formatted input: When the input data is arranged in a specific format, it is called formatted input. scanf function is used for this purpose in C. General syntax: scanf(“control string/format specifier”, &arg1,&arg2..); control string/format specifier refers to the format of the input data. It includes the conversion character %, a data type character and an optional number that specifies the field width. It also may contain new line character or tab. &arg1, &arg2 refers to the address of memory locations where the data should be stored. Example: scanf(“%d”,&num1); Formatted output: printf is used for formatted output to standard output depending on the format specification. Format specifier/control string, along with the data to be output are the parameters to the function. General syntax: printf(“control string/format specifier”,data1,data2..) control string/format specifier indicates how many arguments follow and their data types. Data1,data2 are the variables whose data are formatted and printed according to the specifications of the control string. Example: printf(“%d %d”,no1,no2); The different format specifiers used are: %d-int values %f-float values %c-for char values %s- for string	4M <i>Explanation of each term 2M</i>	



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<p>(e) Ans</p>	<p>Write a program to add two 2 x 2 matrices and print result</p> <pre>#include <stdio.h> #include <conio.h> void main() { int c, d, first[2][2], second[2][2], sum[2][2]; clrscr(); printf("Enter the elements of first matrix\n"); for (c = 0; c < 2; c++) for (d = 0; d < 2; d++) scanf("%d", &first[c][d]); printf("Enter the elements of second matrix\n"); for (c = 0; c < 2; c++) for (d = 0; d < 2; d++) scanf("%d", &second[c][d]); printf("Sum of entered matrices:-\n"); for (c = 0; c < 2; c++) { for (d = 0; d < 2; d++) { sum[c][d] = first[c][d] + second[c][d]; printf("%d\t", sum[c][d]); } printf("\n"); } getch(); }</pre>	<p>4M</p> <p><i>Input two matrices 1M</i></p> <p><i>Addition logic 2M</i></p> <p><i>Display of matrix addition 1M</i></p>
	<p>(f) Ans</p>	<p>Write a program to swap the values of two integer numbers using pointer</p> <pre>#include <stdio.h> void main() { int a,b,*ptr1,*ptr2; int t,*temp; clrscr();</pre>	<p>4M</p> <p><i>Declarat ion of variable and pointers 1M</i></p>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

		<pre>ptr1=&a; ptr2=&b; printf("Enter 2 numbers:"); scanf("%d %d",&a,&b); printf("Before swap a=%d and b=%d\n",a,b); *temp=*ptr1; *ptr1=*ptr2; *ptr2=*temp; printf("after swap a=%d and b=%d\n",a,b); getch(); }</pre>	<p><i>Swapping logic</i> 2M</p> <p><i>Display swapped values</i> 1M</p>
6	(a) Write a program to read two strings and find whether they are equal or not. Ans	<pre>#include<stdio.h> void main() { char str1[10],str2[10]; clrscr(); printf("Enter first string:"); scanf("%s",str1); printf("Enter second string:"); scanf("%s",str2); if(strcmp(str1,str2)==0) printf("Strings are equal"); else printf("Strings are not equal"); getch(); }</pre>	<p>16 4M</p> <p><i>Reading (input) of two strings</i> 2M</p> <p><i>Checking equality</i> 2M</p>
	(b) Write a program to sort an array of 10 integers in ascending order. Ans <i>Note : Any other sorting logic shall be considered.</i>	<pre>#include <stdio.h> void main() { int array[10]; int i, j, num, temp; printf("Enter 10 integers"); for (i = 0; i<10 ;i++)</pre>	<p>4M</p> <p><i>Input of array</i> 1M</p>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<pre>{ scanf("%d", &array[i]); } printf("original array is \n"); for (i = 0; i < 10; i++) { printf("%d\n", array[i]); } /* Bubble sorting begins */ for (i = 0; i < 10; i++) { for (j = 0; j < (9 - i); j++) { if (array[j] > array[j + 1]) { temp = array[j]; array[j] = array[j + 1]; array[j + 1] = temp; } } } printf("Sorted array is...\n"); for (i = 0; i < 10; i++) { printf("%d\n", array[i]); } getch(); }</pre>	<p><i>Sorting</i> 2M</p> <p><i>Display of sorted array</i> 1M</p>
<p>(c) Ans</p>	<p>Write algorithm and draw flowchart to print whether entered number is even or odd.</p> <p>Algorithm :</p> <ol style="list-style-type: none">1. Start2. Input a number.3. Divide the number by 2 to find remainder.4. If remainder of the division is 0, display the number is even otherwise display the number as odd5. Stop.	<p>4M</p> <p><i>correct Algorithm</i> m 2M</p>



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

		<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 --> Stop([Stop]) DisplayOdd --> Stop </pre>	<p><i>Correct Flow chart</i> 2M</p>														
(d)	Differentiate between call by value and call by reference.	4M	4M														
Ans	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">S. No</th> <th style="width: 45%;">Call by value</th> <th style="width: 45%;">Call by reference</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>A copy of the value is passed to the function</td> <td>An address of value is passed to the function</td> </tr> <tr> <td style="text-align: center;">2</td> <td>The values of the actual parameters do not change by changing the formal parameters.</td> <td>The values of the actual parameters do change by changing the formal parameters.</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Actual and formal arguments are created at the different memory location</td> <td>Actual and formal arguments are created at the same memory location</td> </tr> <tr> <td style="text-align: center;">4</td> <td>Changes made inside the function is limited to the function only.</td> <td>Changes made inside the function remain outside of the function also.</td> </tr> </tbody> </table>	S. No	Call by value	Call by reference	1	A copy of the value is passed to the function	An address of value is passed to the function	2	The values of the actual parameters do not change by changing the formal parameters.	The values of the actual parameters do change by changing the formal parameters.	3	Actual and formal arguments are created at the different memory location	Actual and formal arguments are created at the same memory location	4	Changes made inside the function is limited to the function only.	Changes made inside the function remain outside of the function also.	<p><i>Any 4 relevant differences</i> 1M each</p>
S. No	Call by value	Call by reference															
1	A copy of the value is passed to the function	An address of value is passed to the function															
2	The values of the actual parameters do not change by changing the formal parameters.	The values of the actual parameters do change by changing the formal parameters.															
3	Actual and formal arguments are created at the different memory location	Actual and formal arguments are created at the same memory location															
4	Changes made inside the function is limited to the function only.	Changes made inside the function remain outside of the function also.															
(e)	Write a program to calculate area of circle using function.	4M	4M														
Ans	<p><i>Note: Any other correct logic shall be considered</i></p> <pre> #include<stdio.h> void main() { int radius; float area; float calc_area(int); </pre>																



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

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Programming in C

Subject Code: 17212

	<pre>clrscr(); printf("Enter radius:"); scanf("%d",&radius); area=calc_area(radius); printf("Area of circle :%f",area); getch(); } float calc_area(int r) { float a; a=3.14*r*r; return(a); }</pre>	<p><i>Input of radius</i> <i>1M</i></p> <p><i>Function to calculate area</i> <i>2M</i></p> <p><i>Display output</i> <i>1M</i></p>
<p>(f) Ans</p>	<p>Write a program to reverse a 3 digit integer number. <i>Note: Any other correct logic shall be considered</i></p> <pre>#include<stdio.h> void main() { int num,q,r,sum=0; clrscr(); printf("Enter 3 digit number :"); scanf("%d",&num); printf("Reverse number :\n"); while(num!=0) { r=num%10; q=num/10; printf("%d",r); num=q; } getch(); }</pre>	<p><i>4M</i></p> <p><i>Input of a 3 digit number</i> <i>1M</i></p> <p><i>Reverse logic</i> <i>2M</i></p> <p><i>Display of output</i> <i>1M</i></p>