



# V2V EDTECH LLP

Online Coaching at an Affordable Price.

## OUR SERVICES:

- Diploma in All Branches, All Subjects
- Degree in All Branches, All Subjects
- BSCIT / CS
- Professional Courses



**+91 93260 50669**



**v2vedtech.com**



**V2V EdTech LLP**



**v2vedtech**



SUMMER– 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

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 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 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.	Answers	Marking Scheme								
1.		<b>Attempt any FIVE of the following:</b>	<b>10 Marks</b>								
	A)	<b>State different data types supported by 'C' language.</b>	<b>5 X 2M</b>								
	Ans.:	<p><i>(Note: Any four other correct data type shall be considered)</i></p> <p><b>Data types in C language are :</b></p> <table border="1" style="width: 100%;"> <tr> <td rowspan="5" style="vertical-align: top;"><b>• Primary or basic data types</b></td> <td><b>Character (char)</b> is used to store single character or number at a time.</td> </tr> <tr> <td><b>Integer (int)</b> is used to store only integer values with no decimal points.</td> </tr> <tr> <td><b>Float (float)</b> is used to store only floating point numbers with decimal points are allowed.</td> </tr> <tr> <td><b>Double (double)</b> has double value than float</td> </tr> <tr> <td><b>Void – void</b></td> </tr> <tr> <td><b>• User defined data types</b></td> <td>Defined by users as per their need <b>Array , structure</b></td> </tr> </table>	<b>• Primary or basic data types</b>	<b>Character (char)</b> is used to store single character or number at a time.	<b>Integer (int)</b> is used to store only integer values with no decimal points.	<b>Float (float)</b> is used to store only floating point numbers with decimal points are allowed.	<b>Double (double)</b> has double value than float	<b>Void – void</b>	<b>• User defined data types</b>	Defined by users as per their need <b>Array , structure</b>	<p><b>2M</b> <i>(1/2 mark each for correct Any four data type)</i></p>
<b>• Primary or basic data types</b>	<b>Character (char)</b> is used to store single character or number at a time.										
	<b>Integer (int)</b> is used to store only integer values with no decimal points.										
	<b>Float (float)</b> is used to store only floating point numbers with decimal points are allowed.										
	<b>Double (double)</b> has double value than float										
	<b>Void – void</b>										
<b>• User defined data types</b>	Defined by users as per their need <b>Array , structure</b>										
	B)	<b>State use of continue statement.</b>	<b>2M</b>								
	Ans.:	<p><b>Use of continue :</b></p> <ul style="list-style-type: none"> <li>• Continue statement is used to continue the loop with the next iteration after skipping any statement in between.</li> <li>• The continue statement tells the compiler that, skip the following statements and continue with the next iteration.</li> </ul> <p><b>Syntax: continue;</b></p>	<p><i>(Minimum two uses 1 mark for use 1 M for syntax)</i></p>								
	C)	<b>Give syntax of switch case statement.</b>	<b>2M</b>								



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

**22218**

	<p><b>Ans.:</b> <b>Switch statement</b></p> <ul style="list-style-type: none"><li>• Uses single expression/condition for multiple choices.</li></ul> <p><b>Syntax of switch case statement:</b> switch(expression or variable ) { case value1:     {         Statement;         break;     } case value2:     {         Statement;         break;     }     .     . Default:     {         Statement;     } }</p>	<p><i>2 M For Correct syntax</i></p>
	<p><b>D) Give syntax of declaring user defined function. Give one example.</b></p> <p><b>Ans.:</b> <b>Function declaration:</b></p> <p>A function declaration specifies function's name, parameters and return type. It doesn't contain function body. A function declaration gives information to the compiler that the function may later be used in the program.</p> <p><b>Syntax of function declaration:</b></p> <p><b>returnType functionName(type1 argument1, type2 argument2,...);</b></p> <p>For example, int addNumbers(int a, int b); is the function declaration which provides following information to the compiler:</p> <ul style="list-style-type: none"><li>• name of the function is addNumbers()</li><li>• return type of the function is int</li><li>• two arguments of type int are passed to the function</li></ul> <p>The function declaration is not needed if the user-defined function is defined before the main() function.</p> <p>OR (Optional)</p> <p><b>Example:</b> #include&lt;stdio.h&gt;</p> <p>float square ( float x );                      // function declaration</p>	<p><b>2M</b></p> <p><i>1 Mark for declaration / syntax of user defined function and one mark for any one relevant use</i></p>



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

	<pre>// main function, program starts from here  int main() {     float m, n ;     printf ( "\nEnter some number for finding square \n");     scanf ( "%f", &amp;m );      n = square ( m ) ;           // function call     printf ( "\nSquare of the given number %f is %f",m,n ); }  float square ( float x ) // function definition {     float p ;     p = x * x ;     return ( p ) ; }</pre>	
	<p><b>E) Give the meaning of declaration int *ptr.</b></p> <p><b>Ans.:</b></p> <ul style="list-style-type: none"> <li>• A pointer is a variable that stores memory address of another variable which is of similar data type.</li> <li>• Indirection operator (*) is an operator used to obtain the value of a variable to which a pointer points.</li> </ul> <p><b>int *ptr;</b> The above statement declares ptr as an integer pointer variable. It is also used as value at operator i.e. it reads the value from the address stored in pointer variable.</p> <p><b>Example: printf("%d", *ptr);</b> The above statement displays value present at the address stored in ptr variable.</p>	<p>2M</p> <p><b>One mark for meaning and one mark for one relevant example</b></p>
	<p><b>F) Explain initialization of pointer with example.</b></p> <p><b>Ans.:</b> Pointer is variable used to store the memory address of the variable. Variables store the values and pointers stores their addresses at which these variables are located.</p> <p><b>Pointer declaration &amp; initialization:</b> In initialization statement of pointer name of variable is preceded by &amp; (address operator) operator.</p> <p><b>Syntax of initialization of pointer:-</b> Pointer_name = &amp; variable_name;</p> <p><b>Example:</b> int *ptr ; /* declaration of pointer ptr of int type*/ int a; /* declaration of integer variable a*/</p>	<p>2M</p> <p><b>One mark for meaning and one mark for one relevant example</b></p>



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

		<code>ptr = &amp;a; /* pointer ptr is pointing to variable a*/</code>	
	<b>G)</b>	<b>Give syntax of declaring and initializing of structure.</b>	<b>2M</b>
	<b>Ans.:</b>	<p><b>Structure:</b> A structure is a collection of one or more variables of same or different data types grouped together under a single name.</p> <p><b>Syntax of declaration of structure:</b></p> <pre>struct structure_name {     Data_type1 variable 1;     Data_type2 variable 2;     .     .     Data_typen variable n; };</pre> <p><b>Syntax of initialization of structure:</b></p> <pre>struct structure_name {     Data_type1 variable 1;     Data_type2 variable 2;     .     .     Data_typen variable n; }variable_name;</pre> <p style="text-align: center;"><b>(OPTIONAL)</b></p> <p><b>Example:</b></p> <pre>struct book {     char tit[20];     char auth[20];     int price; }b1;</pre>	<p><i>One mark for declaration and one mark for initialization with relevant example</i></p>
<b>2.</b>		<b>Attempt any THREE of the following :</b>	<b>12 Marks 3 X 4M</b>
	<b>A)</b>	<b>State the use of %d and %f and write the printf statement of 'C' using above mentioned symbols.</b>	<b>4M</b>



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

	<p><b>Ans.:</b> %d and %f are format specifier used to access or display integer data types and float data types respectively using variable in printf and scanf statements in c programming.</p> <p><b>General syntax of %d :</b> Use of %d to declare and access the integer data types.</p> <p><b>Example:</b> scanf(“%d”,&amp;num1);</p> <p><b>General syntax of %f :</b> Use of %f to declare and access the float data types.</p> <p><b>Example:</b> scanf(“%f”,&amp;percent);</p> <p><b>example of printf statements using %d and % f:</b> printf(“the marks of subject 1 are: %d”,num1); printf(“the percentage of student is : %f”, percent);</p>	<p><i>Two marks for use of %d and %f and 2M for example showing use of these symbols</i></p>																					
	<p><b>B) Compare while and do-while loop.</b></p>	<p>4M</p>																					
	<p><b>Ans.:</b> <b>Comparison of while and do-while loop:</b></p> <table border="1" data-bbox="329 974 1330 1430"> <thead> <tr> <th>While</th> <th>Do-while</th> </tr> </thead> <tbody> <tr> <td>Entry controlled loop</td> <td>Exit controlled loop</td> </tr> <tr> <td>Condition is checked first</td> <td>Condition is checked last</td> </tr> <tr> <td>Executes only if satisfies the Condition</td> <td>Executes at least once even if the condition is not satisfied.</td> </tr> <tr> <td>Syntax : while(condition) { Code; }</td> <td>Syntax: do { Code; } while(condition);</td> </tr> </tbody> </table>	While	Do-while	Entry controlled loop	Exit controlled loop	Condition is checked first	Condition is checked last	Executes only if satisfies the Condition	Executes at least once even if the condition is not satisfied.	Syntax : while(condition) { Code; }	Syntax: do { Code; } while(condition);	<p><i>Any four differences 1M each</i></p>											
While	Do-while																						
Entry controlled loop	Exit controlled loop																						
Condition is checked first	Condition is checked last																						
Executes only if satisfies the Condition	Executes at least once even if the condition is not satisfied.																						
Syntax : while(condition) { Code; }	Syntax: do { Code; } while(condition);																						
	<p><b>C) State the ways of declaration and initialization of string variables.</b></p>	<p>4M</p>																					
	<p><b>Ans.:</b> String is collection of characters, numbers and special symbols. A string is terminated by a null character \0 (NULL Character).</p> <p><b>Syntax for declaring string :</b> Char string_name[size]; Declaring string of 8 characters. char str[8];</p> <table border="1" data-bbox="553 1749 886 2009"> <tr><td>Str[0]</td><td></td><td>1000</td></tr> <tr><td>Str[1]</td><td></td><td>1001</td></tr> <tr><td>Str[2]</td><td></td><td>1002</td></tr> <tr><td>Str[3]</td><td></td><td>1003</td></tr> <tr><td>Str[4]</td><td></td><td>1004</td></tr> <tr><td>Str[5]</td><td></td><td>1005</td></tr> <tr><td>Str[6]</td><td></td><td>1006</td></tr> </table>	Str[0]		1000	Str[1]		1001	Str[2]		1002	Str[3]		1003	Str[4]		1004	Str[5]		1005	Str[6]		1006	<p><i>Declaration with example: 2 marks, Initialization with example: 2 marks</i></p>
Str[0]		1000																					
Str[1]		1001																					
Str[2]		1002																					
Str[3]		1003																					
Str[4]		1004																					
Str[5]		1005																					
Str[6]		1006																					



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

	<p style="text-align: right;">Str[7] <input type="text"/> 1007</p> <p>• <b>Syntax for initializing string :</b> Str1[subscript] =value; Str1 = "PRADEEP"; char str[8];</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>Str[0]</td><td>P</td><td>1000</td></tr> <tr><td>Str[1]</td><td>R</td><td>1001</td></tr> <tr><td>Str[2]</td><td>A</td><td>1002</td></tr> <tr><td>Str[3]</td><td>D</td><td>1003</td></tr> <tr><td>Str[4]</td><td>E</td><td>1004</td></tr> <tr><td>Str[5]</td><td>E</td><td>1005</td></tr> <tr><td>Str[6]</td><td>P</td><td>1006</td></tr> <tr><td>Str[7]</td><td>\0</td><td>1007</td></tr> </table> <p><b>Another way of declaring and initializing string is :</b></p> <ul style="list-style-type: none"> <li>• char Str1[ ]={'P','R','A','D','E','E','P','\0'}; //as an unsized array This method requires the user to put a '\0' at the end</li> <li>• char name[10]={'C','O','M','P','U','T','E','R','S','\0'}; for sized array.</li> <li>• char name[ ]="PRADEEP"; for unsized array. Puts '\0' automatically</li> </ul> <p>char name[10]="COOMPUTERS"; //sized array.</p>	Str[0]	P	1000	Str[1]	R	1001	Str[2]	A	1002	Str[3]	D	1003	Str[4]	E	1004	Str[5]	E	1005	Str[6]	P	1006	Str[7]	\0	1007	
Str[0]	P	1000																								
Str[1]	R	1001																								
Str[2]	A	1002																								
Str[3]	D	1003																								
Str[4]	E	1004																								
Str[5]	E	1005																								
Str[6]	P	1006																								
Str[7]	\0	1007																								
	<p><b>D) Explain recursion function with example and state its advantages.</b></p>	<p><b>4M</b></p>																								
<p><b>Ans.:</b></p>	<p><b>Recursive function:</b> Recursion is the process of function calling itself again and again.</p> <p><b>Definition :</b> Recursion function is the process in which function calls itself.</p> <p><b>Recursive function:</b> Recursion is the process of function calling itself again and again. A Recursive function contains function call to itself in the body of function.</p> <pre>void recurse() {     recurse(); /* Function calls itself */ }  int main() {     recurse(); /* Sets off the recursion */     return 0; }</pre> <p><b>Example:</b></p>	<p><i>(For explanation: 2 M, Example: 1 mark, 1M any two advantages)</i></p>																								



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

	<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; void main() {     int n,fact;     clrscr();     printf("enter the number");     scanf("%d",&amp;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)); -----Recursive function call     } }  In the above example recursive function factorial() is used to print the Factorial of a number. <b>Advantages :</b> <ul style="list-style-type: none"> <li>• Reduces length of the program</li> <li>• Reduces unnecessary calling of a function.</li> <li>• Useful when same solution is to be applied many times.</li> </ul> </pre>	
3.	<b>Attempt any THREE:</b>	<b>12 Marks</b>
	<b>(A) Explain the use of increment &amp; decrement operator. Also Give difference between i++ &amp; ++i statement with example.</b>	<b>4M</b>
	<p><b>Ans:</b></p> <ul style="list-style-type: none"> <li>• Increment operator (++) is used to increase the value by one.</li> <li>• Decrement operator (--) is used to reduce the value by one.</li> </ul> <p><b>Example:</b> Pre-incremental Operator, Post-Incremental Operator ++x is similar x=x+1 if x is 5 then after ++x or x++, x will become 6.</p> <p><b>Or</b> Pre-decremental Operator, Post-decremental Operator x++ is similar to x=x+1.</p>	<p><i>(Use of increment, Decrement - 02 Marks (1 Mark each) Difference with example -</i></p>





SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

	<p>--y is similar to y=y-1 if y=5 then after --y or y--,y become 4. Or y--is similar to y=y-1 Difference between i++ &amp; ++i with Example</p> <ul style="list-style-type: none"> <li> <b>Postfix increment operator (i++):</b>            When postfix ++ or (--) is used with a variable in an expression ,the expression is evaluated first using the original value of the variable and then the variable is incremented (or decremented)by one.  <b>Example:</b>  <pre>main() { int a,z,i=10,j=20; a=i * j++; z=i * j; printf("\n a=%d z=%d",a,z); getch(); }</pre> <b>Output:</b>            a=200 z=210         </li> <li> <b>Prefix Increment operator(++i):</b>            When prefix ++ or (--) is used in an expression, the variable is incremented (or decrement) first and then the expression is evaluated using the new value of the variable.  <b>Example:</b>  <pre>main() { int a,z,i=10,j=20; a=i * ++j; z=i* j; printf("\n a=%d z=%d",a,z); getch(); }</pre> <b>Output:</b>            a=210 z=210         </li> </ul>	<p>2Marks)</p>
	<p><b>(B) Declare and initialize the one dimensional integer array with 10 elements.</b></p>	<p>4M</p>
	<p><b>Ans:</b> Declaration of one dimensional array: Syntax: datatype variable-name[size]; Declaration of 10 array element is : int a[10]; Where a is variable name or array name, 10 is size of an array, int is datatype</p> <p>Initialization of one dimensional array: Syntax: datatype array-name[size] ={list of values}; Initialization of 10 array elements: int a[10]={ 10,20,30,40,50,60,71,70,80,90};</p>	<p><i>Declaration: 2 marks Initialization: 2 marks</i></p>
	<p><b>(C) Explain concept of pointer's arithmetic operation with example</b></p>	<p>4M</p>
	<p><b>Ans:</b> Pointer is a variable that points to a memory location. Memory addresses are numeric value that ranges from zero to maximum memory size in bytes. These</p>	<p><i>(Introduction: 1</i></p>



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

	<p>addresses can be manipulated like simple variables. You can increment, decrement, calculate or compare these addresses manually.</p> <p>C language provides a set of operators to perform arithmetic and comparison of memory addresses. Pointer arithmetic and comparison in C is supported by following operators -</p> <ul style="list-style-type: none"> <li>• Increment and decrement ++ and –</li> <li>• Addition and Subtraction + and –</li> <li>• Comparison &lt;, &gt;, &lt;=, &gt;=, ==, !=</li> </ul> <p><b>Example of pointer increment and decrement:</b> Increment operator when used with a pointer variable returns next address pointed by the pointer. The next address returned is the sum of current pointed address and size of pointer data type.</p> <p>Similarly, decrement operator returns the previous address pointed by the pointer. The returned address is the difference of current pointed address and size of pointer data type.</p> <p>For example, consider the below statements.</p> <pre>int num = 5; // Suppose address of num = 0x1230 int *ptr; // Pointer variable  ptr = &amp;num; // ptr points to 0x1230 or ptr points to num ptr++; // ptr now points to 0x1234, since integer size is 4 bytes ptr--; // ptr now points to 0x1230</pre>	<p><i>mark, List of operations : 1 mark Example: 2 marks)</i></p>
	<p><b>(D) Explain array of structure with example.</b></p>	<p><b>4M</b></p>
<p><b>Ans:</b></p>	<p><b>Array of structure:-</b></p> <p>A structure is a composite datatype with a collection of variables. These variables can have different data types and collectively form a structure of a composite datatype. An array of structures is a sequential collection of structures. With structures, you can store mixed record types and with an array supporting this, you can have a list of mixed record types. It can be used when we want to use many variables of the same structure.</p> <p><b>Example:</b></p> <p>If a structure for student data is defined and it has to be used for 10 different students, then array of structure can be declared as</p> <pre>struct student {     int rollno;</pre>	<p><i>(Explanation – 2M, example - 2M)</i></p>



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

		<pre>char name[20]; } s[10];</pre> <p>Here data in the form of rollno and name can be stored or accessed for 10 students. Here s[0].rollno and s[0].name will be the data for first student. s[1].rollno and s[1].name will be the data for second student and so on.</p>	
4.		<b>Attempt any THREE of the following</b>	12M
	(A)	<b>Write a 'C' program to enter basic salary. Calculate gross salary with 5% DA and 15% TA on basic salary. Display calculated gross salary.</b>	4M
	Ans.	<pre>#include&lt;conio.h&gt; #include&lt;stdio.h&gt; void main() { int b_salary,DA,TA,g_salary; clrscr(); printf("Enter basic salary:"); scanf("%d",&amp;b_salary); DA=0.05*b_salary; TA=0.15*b_salary; g_salary=b_salary+DA+TA; printf("Gross salary is:%d",g_salary); getch(); }</pre> <p><b>Output:</b> Enter basic salary:1000 Gross salary is:1200</p>	<i>Correct Program: 3 marks Output: 1 mark</i>
	(B)	<b>Write a C program to find whether the given number is prime or not prime.</b>	4M
	Ans:	<pre>#include &lt;stdio.h&gt; #include&lt;conio.h&gt; void main() { int n, i, c = 0; printf("Enter the number :"); scanf("%d", &amp;n); for (i = 1; i &lt;= n; i++) { if (n % i == 0) { c++; } } if (c == 2) { printf("%d is a Prime number",n);</pre>	<i>Correct Program: 3 marks Output: 1 mark</i>



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

	<pre> } else { printf("%d is not a Prime number",n); } return 0; } </pre> <p><b>Output:</b> Enter the number:7 7 is a prime number</p>	
	<p><b>(C) Define array and explain how elements of array can be accessed.</b></p>	4M
<p><b>Ans:</b></p>	<p><b>Definition:</b> Array is a collection of variables having same data type referred by the same name.</p> <p><b>Accessing elements of array:</b></p> <p>while accessing array elements we can use loop. The following code is used to access elements of array,</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> for(i=0;i&lt;10;i++) { printf("\n Percent of student %d :\t %f",i+1,percentage[i]); } </pre> </div> <ul style="list-style-type: none"> <li>• The for loop is used to repeat the statements.</li> <li>• printf() function is used to display the array elements</li> <li>• the %f specifies the compiler that the data which is going to be accessed is of type float type.</li> <li>• The value of i varies from 0 to 9 so percentage[i] specifies which array elements to be read.</li> </ul>	<p><b>Definition :1 mark</b> <b>Accessing elements of array:3 marks</b></p>
	<p><b>(D) Write a C program using pointer to swap the value of two integer numbers.</b></p>	4M
<p><b>Ans:</b></p>	<pre> #include&lt;conio.h&gt; #include&lt;stdio.h&gt; void swap(int *a,int *b); void main() { int n1,n2; printf("Enter two numbers:"); scanf("%d%d",&amp;n1,&amp;n2); printf("Numbers before swap:n1=%d n2=%d",n1,n2); swap(&amp;n1,&amp;n2); printf("Numbers after swapping: n1=%d n2=%d",n1,n2); getch(); } </pre>	<p><b>Correct Program: 3 marks</b> <b>Output: 1 mark</b></p>



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

	<pre>void swap(int *a,int *b) { int temp=*a; *a=*b; *b=temp; }</pre> <p><b>Output:</b> Enter two numbers: 10 20 Numbers before swap:n1=10 n2=20  Numbers after swap:n1=20 n2=10</p>	
	<b>(E) Write a C program to declare a structure 'student' with members as Roll no, name and marks. Accept and display data for one instance.</b>	<b>4M</b>
	<p><b>Ans:</b></p> <pre>#include&lt;conio.h&gt; #include&lt;stdio.h&gt; struct student { int roll_no; char name[10]; float marks; }s; void main() { clrscr(); printf("Enter roll number:"); scanf("%d",&amp;s.roll_no); printf("Enter name:"); scanf("%s",&amp;s.name); printf("Enter marks:"); scanf("%f",&amp;s.marks); printf("The given information is:\nRoll no=%d\tName=%s\tMarks=%f", s.roll_no,s.name,s.marks); getch(); }</pre> <p><b>Output:</b> Enter roll number:10 Enter name:ABC Enter marks:75.89 The given information is: Roll no=10 Name=ABC Marks=75.89</p>	<i>Correct Program: 3 marks Output: 1 mark</i>
<b>5.</b>	<b>Attempt any Two of the following:</b>	<b>12 Marks</b>
	<b>A) Explain else-if ladder with syntax and its execution with example. Also draw flow chart for else-if ladder.</b>	<b>6M</b>
	<p><b>Ans: if-else Ladder Statement:</b> The if-else ladder statement in C programming language is used to test set of</p>	<i>(Introduction: 1</i>



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

**22218**

conditions in sequence. if condition is tested only when all previous if conditions in if-else ladder is false. If any of the conditional expression evaluates to true, then it will execute the corresponding code block and exits whole if-else ladder.

**Syntax of if-else ladder statement:**

```
if(condition_expression_One)
{
    statement1;
}
else if (condition_expression_Two)
{
    statement2;
}
else if (condition_expression_Three)
{
    statement3;
}
else
{
    statement4;
}
```

First of all condition\_expression\_One is tested and if it is true then statement1 will be executed and control comes out of whole if else ladder. If condition\_expression\_One is false then only condition\_expression\_Two is tested. Control will keep on flowing downward, If none of the conditional expression is true. The last else is the default block of code which will gets executed if none of the conditional expression is true.

**Flowchart of if-else ladder:**

*mark,  
Syntax: 1  
mark,  
Explanation:  
1 mark,  
Flowchart  
: 1 mark,  
Example:  
any  
program  
using if-  
else  
ladder: 2  
marks)*



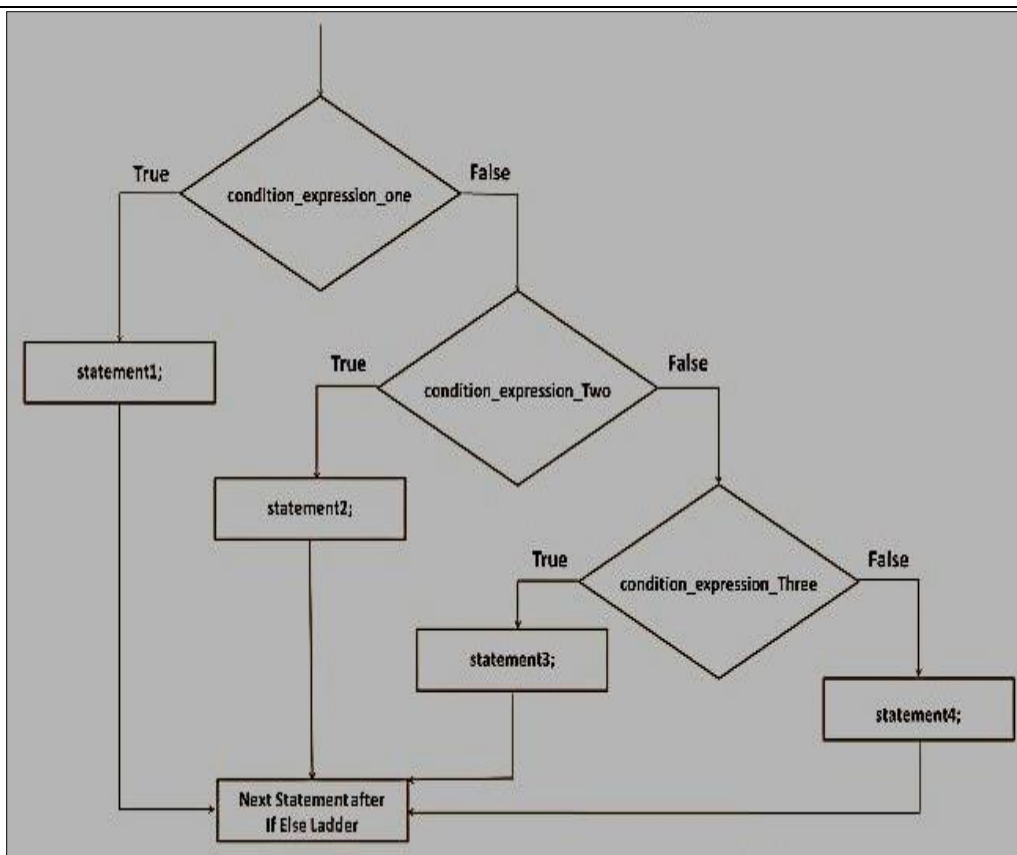
SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218



**Example of if-else ladder:**

C Program to print grade of a student using if -else Ladder Statement

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int marks;
    printf("Enter your marks between 0-100\n");
    scanf("%d", &marks);
    /* Using if else ladder statement to print
    Grade of a Student */
    if(marks >= 90)
    {
        /* Marks between 90-100 */
        printf("YOUR GRADE : A\n");
    }
    else if (marks >= 70 && marks < 90)
    {
        /* Marks between 70-89 */
        printf("YOUR GRADE : B\n");
    }
}
```



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

**22218**

	<pre>else if (marks &gt;= 50 &amp;&amp; marks &lt; 70) {     /* Marks between 50-69 */     printf("YOUR GRADE : C\n"); } else {     /* Marks less than 50 */     printf("YOUR GRADE : Failed\n"); } getch(); }</pre> <p><b>Output:</b> Enter your marks 96 YOUR GRADE : A Enter your marks 75 YOUR GRADE : B Enter your marks 60 YOUR GRADE : C Enter your marks 35 YOUR GRADE : Failed</p>	
	<p><b>B) Write the program to accept 10 (ten) numbers from user using array, search and print the location of a given number.</b></p>	<p><b>6M</b></p>
<p><b>Ans:</b></p>	<p><b>Program:</b></p> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; void main( ) {     int array[100], search, c;     printf("Enter 10 numbers\n");     for (c = 0; c &lt; 10; c++)         scanf("%d", &amp;array[c]);     printf("Enter a number to search\n");     scanf("%d", &amp;search);     for (c = 0; c &lt; 10; c++)     {         if (array[c] == search) /* If required element is found */         {             printf("%d is present at location %d.\n", search, c+1);             break;         }     } }</pre>	<p><b>(Syntax: 3 marks, Logic: 3 marks)</b></p>





SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

**22218**

	<pre>if (c == 10)     printf("%d isn't present in the array.\n", search); getch( ); }</pre> <p><b>Output:</b> Enter 10 numbers 4 3 7 2 9 6 5 1 8 10 Enter a number to search 2 2 is present at location 4.</p>	
	<p><b>c) Write a 'C' program to print factorial of number n (i.e. <math>n! = n \times (n-1) \times (n-2) \times \dots</math>) using recursion function.</b></p>	<b>6M</b>
<b>Ans:</b>	<p><b>Program:</b></p> <pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; int factorial(int n); void main( ) { int fact,num; printf("\n Enter Number="); scanf("%d",&amp;num); fact=factorial(num); printf("\n Factorial of a number = %d",fact); getch( ); }  int factorial(int n) { int f; if(n==1) return 1; else { f = n * factorial(n-1); return f; } }</pre> <p><b>OUTPUT:</b></p>	<p><i>(Syntax: 3 marks, Logic: 3 marks)</i></p>



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

22218

		Enter Number=5 Factorial of a number=120	
6.		<b>Attempt any Two of the following:</b>	<b>12 Marks</b>
	A)	<b>Write a 'C' program to copy one string into another without using strcpy function.</b>	<b>6M</b>
	<b>Ans:</b>	<p><b>Program:</b>  <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; void main( ) {     char s1[100], s2[100], i;     printf("Enter string s1: ");     scanf("%s",s1);     for(i = 0; s1[i] != '\0'; i++)     {         s2[i] = s1[i];     }     s2[i] = '\0';     printf("String s2: %s", s2);     getch( ); }</pre> </p> <p><b>Output:</b>  Enter String s1: hello  String s2: hello</p>	(Syntax: 3 marks, Logic: 3 marks)
	B)	<b>Write a 'C' program to find sum of natural number entered by user.</b>	<b>6M</b>
	<b>Ans:</b>	<pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; void main( ) {     int n, i, sum = 0;     printf("Enter a positive integer: ");     scanf("%d",&amp;n);     for(i=1; i &lt;= n; ++i)     {         sum += i; // sum = sum+i;     }     printf("Sum = %d",sum);     getch( ); }</pre> <p><b>Output:</b>  Enter a positive integer: 100  Sum = 5050</p>	(Syntax: 3 marks, Logic: 3 marks)
	C)	<b>Declare a structure circle containing data members as radius, area, perimeter. Accept radius for one variable from user and find out perimeter and area.</b>	<b>6M</b>
	<b>Ans:</b>	<b>Program:</b>	(Syntax: 3



SUMMER- 18 EXAMINATION

Subject Name: Programming in 'C'

Model Answer

Subject Code:

**22218**

	<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; struct circle { float radius; float area; float perimeter; }c;  void main( ) { printf(" Enter radius:"); scanf("%f",&amp;c.radius); c.area = 3.14 * c.radius * c.radius; c.perimeter = 2 * 3.14 * c.radius; printf("\n Area of circle=%f \n Perimeter of Circle=%f",c.area,c.perimeter); getch( ); } <b>Output:</b> Enter radius:5.0 Area of circle=78.500000 Perimeter of Circle=31.400000</pre>	<p><i>marks, Logic: 3 marks)</i></p>
--	--	--