

**Internal Assessment Test 1 – March 2018-SCHEME & SOLUTION**

<b>Sub:</b>	Programming in C and Data Structures						<b>Code:</b>	17PCD23	
<b>Date:</b>	13/03/2018	<b>Duration:</b>	90mins	<b>Max Marks:</b>	50	<b>Sem:</b>	II	<b>Branch:</b>	ALL

**Note:** Answer any five questions:

		Marks	CO	RBT
<b>1.(a)</b>	<p><b>What is an identifier? What are the rules to construct identifier? Classify following as valid/invalid identifier.</b>  <b>num1,\$num1,a-2,for,1st-paper-marks.</b></p> <p>An identifier is a string of alphanumeric characters that begins with an alphabetic character or an underscore character that are used to represent various programming elements such as variables, functions, arrays, structures, unions and so on.</p> <ul style="list-style-type: none"> <li>Identifier names may consist of letters, digits, and the underscore(_) character, subject to the rules given below: <ol style="list-style-type: none"> <li>The identifier must always begin with a letter. Some systems permit underscore as the first character.</li> <li>ANSI standard recognizes a length of 31 characters. However, the length should not be normally more than eight characters.</li> <li>Uppercase and lowercase are significant. That is ,the identifier Rate is not the same as rate.</li> <li>The identifier name should not be a keyword.</li> <li>White space is not allowed.</li> </ol> </li> </ul> <p><b>Valid Identifier: num,a-2</b>  <b>Invalid Identifier : \$num1,for,1st-paper-marks</b></p>	<b>06M</b>	CO 2	L2
<b>1.(b)</b>	<p><b>Explain any 4 types of operators in C.</b></p> <p>An operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C language is rich in built-in operators and provides the following types of operators –</p> <ul style="list-style-type: none"> <li><b>Arithmetic Operators</b> +, -, /, *, % Ex: a+b, x/y</li> <li><b>Relational Operators</b> &gt;, &lt;, &lt;=, &gt;=, ==, != Ex: a=10,b=5 a&gt;=b, a==6</li> <li><b>Logical Operators</b> &amp;&amp;,   , ! Ex: a&amp;&amp; b</li> <li><b>Bitwise Operators</b> &amp;,  , ~, ^, &gt;&gt;, &lt;&lt; Ex: a=3, a&gt;&gt;1</li> </ul>	<b>04M</b>	CO 2	L1

2	<p><b>What is an algorithm? Write algorithm for the following: i) To find area of triangle by given 3 sides. ii) To check given number is palindrome or not.</b></p> <p>An algorithm is a step by step procedure to solve a given problem. (2M)</p> <p>i) Algorithm to find area of triangle by given 3 sides.(4M)  Input: 3 sides of the triangle  Output: Area of triangle  Step 1: Start  Step 2: Read 3 sides of the triangle  Step 3: Calculate perimeter <math>p=(a+b+c)/2</math> and <math>area=\sqrt{p*(p-a)*(p-b)*(p-c)}</math>  Step 4: Output area  Step 5: Stop</p> <p>ii) Algorithm to check given no. is palindrome or not.(4M)  Input: An Integer number  Output: Number is palindrome or not message.  Step 1:Start  Step 2:Read a integer number  Step 3:Reverse the number  Step 4: Compare the reverse and original number.  If they are same Display “Palindrome”, goto step 5  Else Display “Not Palindrome”, goto step 5  Step 5:Stop</p>	10M	CO 1	L1
3.(a)	<p><b>With Syntax and example explain printf() and scanf() functions.</b></p> <p>i)printf() syntax: (3M)</p> <p>a. printf(“String literal”);  ex: printf(“Hello world\n”);</p> <p>b. printf(“String literal and/or format specifiers”,variable list);  int a=10;  ex: printf(“The value of a is %d”,a);</p> <p>ii)scanf() syntax: (3M)</p> <p>scanf(“format string”,address of variables);  ex: scanf(“%d%d”,&amp;a,&amp;b);</p>	6M	CO 2	L1
3.(b)	<p><b>Evaluate the expression: i) <math>100\%20 \leq 20-5+100\%10-20==5 &gt;= 1 != 20</math> ii) <math>a+=b*=c-=10</math> (Initial values, a=5,b=10 and c=15)</b></p> <p>i) <math>0 \leq 20-5+100\%10-20==5 &gt;= 1 != 20</math> (2M)</p> <p><math>0 \leq 20-5+0-20==5 &gt;= 1 != 20</math></p> <p><math>0 \leq 15+0-20==5 &gt;= 1 != 20</math></p> <p><math>0 \leq 15-20==5 &gt;= 1 != 20</math></p>	4M	CO 2	L3

	<pre> <b>0&lt;=-5==5&gt;=1!=20</b>  <b>0==5&gt;=1!=20</b>  <b>0==1!=20</b>  <b>0!=20--→1</b>  ii) <b>a+=b*=c-=10 (2M)</b>  <b>a=a+b=b*c=c-10</b>  <b>a=a+b=b*5</b>  <b>a=a+50</b>  <b>a=55</b> </pre>			
4.(a)	<p><b>Write a C program to calculate and display area of a rectangle.</b></p> <pre> /*Program to calculate and display area of rectangle*/ #include&lt;stdio.h&gt; int main() {     float l,b,area;     printf("\n Enter len and breadth of rectangle:");     scanf("%f%f",&amp;l,&amp;b);     area=l*b;     printf("The area of rectangle is %f",area);     return 0; } </pre>	05M	Co 2	L3
4.(b)	<p><b>Write a C program to calculate and display sum and average of 5 integers.</b></p> <pre> /*Program to calculate and display sum and average of 5 integers */ #include&lt;stdio.h&gt; int main() {     int a,sum=0,i;     float avg;     for(i=1;i&lt;=5;i++)     {         scanf("%d",&amp;a);         sum=sum+a;     }     avg=sum/5;     printf("The sum of 5 integers is %d",sum);     printf("\nThe average of 5 integers is %f",avg);     return 0; } </pre>	05M	Co 2	L3

5.(a)	<p><b>Explain cascaded if else (else if ladder) and nested if statements with syntax and suitable example.</b></p>	05M	Co 2	L2
<p><b>Cascaded if-else or else if ladder Statement (2.5M)</b></p>				
<p>It is multi-way selection statement. It is used when we must alternatives.</p>				
<p>The syntax of cascade if-else or else if ladder statement is:</p>				
<pre> <b>if</b> ( expression-1 ) {     statement-block1; <b>else if</b> (expression-2 ) {     statement-block-2; } <b>else if</b> (expression-3) {     statement-block-3; } <b>else if</b> (expression4) {     statement-block4; } <b>else</b> {     statement-block-5; } statement-block6; </pre>				
<p>The expression is evaluated in top to bottom order. If an expression is evaluated to true, then the statement-block associated with that expression is executed and the control comes out of the entire else if ladder and continues execution from statement-block-6 if any.</p>				
<p>If all the expressions are evaluated to false, then the last statement- block-5 (default) is executed and the control comes out of the entire else if ladder and continues execution from statement-block-6 if any.</p>				
<p><b>Nested if-else Statement (2.5M)</b></p>				
<p>It is multi-way selection statement. It is used when an action has to be performed based on many decisions.</p>				
<p>An if-else statement within another if-else statement: is called nested if-else statement.</p>				
<p>The syntax of nested if-else statement is:</p>				
<pre> <b>if</b> ( expression-1 ) {     <b>if</b> ( expression-2 )         statement-block-1;     <b>else</b>         statement-block-2;  <b>else</b> { </pre>				

```

if ( expression-3 )
    statement-block-3;
else
    statement-block-4;
}
statement-block-5;

```

The expression-1 is evaluated to true or false. If expression-1 is evaluated to true, then expression2 is evaluated to true or false. If expression-2 is evaluated to true, then statement- block-1 is executed. If expression-2 is evaluated to false, then statement-block-2 is executed.

If expression-1 is evaluated to false, then expression-3 is evaluated to true or false. If expression-3 is evaluated to true, then statement-block-3 is executed. If expression-3 is evaluated to false, then statement-block-4 is executed.

5.(b) **Explain structure of C program with an example.**

05M Co 2 L2

- The basic structure of a C program is shown below

Documentation Section

Link Section

Definition Section

Global Declaration Section

main() Function Section

{

    Declaration Part & Executable Part

}

Subprogram section

Function 1

Function 2

.

.

Function n

- The documentation section consists of a set of comment lines giving the name of the program, the name of the author and other details which the programmer would like to use later.
- The link section provides instructions to the compiler to link functions from the system library.
- The definition section contains all symbolic constants.
- There are some variables that are used in more than one function. Such variables are called global variables and are declared in the global declaration section.
- Every C program must have one main() function section.
- The subprogram section contains all the user-defined functions that are called in the main function. The main function is very important compared to other sections.

	<pre> // C Program to calculate Simple Interest #include&lt;stdio. h&gt; int main() {     float p,t,r,si;     printf("\n Enter the principal, time and rate of interest:");     scanf("%f%f%f",&amp;p,&amp;t,&amp;r);     si=(p*t*r)/100;     printf("\n The simple interest is %f",si);     return 0; } </pre>			
6	<p><b>Explain switch case statement with syntax and example of a calculator program to do simple operations like addition, subtraction, multiplication and division.</b></p> <p>The syntax for a <b>switch</b> statement in C programming language is as follows –</p> <pre> switch(expression) {     case constant-expression :                                 statement(s);                                 break; /* optional */      case constant-expression :                                 statement(s);                                 break; /* optional */                                  default : /* Optional */                                 statement(s); } </pre> <p>The following rules apply to a switch statement –</p> <ul style="list-style-type: none"> <li>• The expression used in a switch statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type.</li> <li>• You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.</li> <li>• The constant-expression for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.</li> <li>• When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.</li> </ul> <p>When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.</p>	10M	Co 2	L3

	<ul style="list-style-type: none"> <li>• A switch statement can have an optional default case, which must appear at the end of the switch.</li> </ul> <pre> /* program to perform calculator*/  #include&lt;stdio.h&gt;  int main() {     int a, b;     char ch;      printf("\n Enter the values of a &amp; b:");     scanf("%d%d",&amp;a,&amp;b);     printf("\n Enter the character for add, sub, mul &amp; div:");     scanf("%c",&amp;ch);     switch(ch)     {         case „+“ : printf(“Sum=%d”,a+b);                     break;         case „-“ : printf(“Diff=%d”,a-b);                     break;         case „*“ : printf(“Prod=%d”,a*b);                     break;         case „/“ : if(b==0)                     printf(“\n Number cannot be divided by zero”);                     else                     printf(“Div=%d”,a/b); break;         default : printf(“\n Invalid Character”);     } </pre>			
7	<p><b>Write a C program to find roots of a quadratic equation of the form <math>ax^2+bx+c=0</math>, where a, b and c are coefficients. Explain the output for all possible cases.</b></p> <pre> /*Program to find roots of a Quadratic Equation. */ #include&lt;stdio.h&gt; #include&lt;math.h&gt; int main() { float a, b, c, d, rpart, ipart, r1, r2; // Input the co-efficients of a, b and c. </pre>	10 M	CO 2	L3

	<pre> printf ( "\nEnter three non-zero coefficients ( a, b and c ) of the Quadratic Equation: " ); scanf ( "%f%f%f", &amp;a, &amp;b, &amp;c ); // Compute the discriminant. d = ( b * b ) - ( 4 * a * c ); printf ( "\nThe discriminant is: %f\n", d ); // Compute Real and Equal roots. if ( d == 0 ) { r1 = r2 = -b / ( 2.0 * a ); printf ( "\nRoots are Real and Equal: \n r1= %f \n r2= %f \n\n", r1, r2 ); } // Compute Real and Distinct roots. else if ( d &gt; 0 ) { r1 = ( -b - ( sqrt ( d ) ) ) / ( 2.0 * a ); r2 = ( -b + ( sqrt ( d ) ) ) / ( 2.0 * a ); printf ( "\nRoots are Real and Distinct: \n r1= %f \n r2= %f \n\n", r1, r2); } // Compute Imaginary roots. else { rpart = - b / ( 2.0 * a ); ipart = sqrt ( ( - d ) ) / ( 2.0 * a ); printf ( "\nRoots are Imaginary: \n r1 = %f + i * %f \n r2 = %f - i * %f \n\n", rpart, ipart, rpart, ipart); } return 0; } </pre>			
8.(a)	<p><b>Explain while loop with an example.</b></p> <p>It is an Entry Controlled Loop.</p> <p>Syntax: while(condition)</p> <pre> { statements; } </pre> <p>In 'while' loop the controlling condition appears at the start of the loop.</p> <p>The iterations do not occur if, the condition at the first iteration, appears false.</p> <p>Ex:/*Program to display 5 natural no.s*/  #include&lt;stdio.h&gt;  int main()  {  int i;  i=1;  while(i&lt;=5)  {  printf("\n%d",i);</p>	05M	CO 2	L1



	<pre> } return 0; } </pre>			
<b>8.(b)</b>	<p><b>Write a C program to check given year is leap year or not.(Considering century year also)</b>  /*Program to check given year is leap year or not.*/  #include&lt;stdio.h&gt;  int main()  {  int year;  printf("Enter a year to check if it is a leap year\n");  scanf("%d", &amp;year);   if ( year%400 == 0)  printf("%d is a leap year.\n", year);  else if ( year%100 == 0)  printf("%d is not a leap year.\n", year);  else if ( year%4 == 0 )  printf("%d is a leap year.\n", year);  else  printf("%d is not a leap year.\n", year);   return 0;  }</p>	<b>05M</b>	<b>CO 2</b>	<b>L3</b>