

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

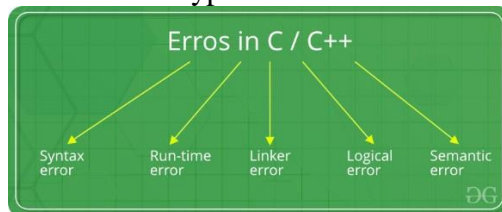
# Instructor Manual

**COURSE NAME: PRINCIPLES OF  
PROGRAMMING USING C**

**COURSE CODE:22CS23**

Pg no	PART A Practice Programs
1.	<i>Familiarization with programming environment, concept of naming the program files storing, compilation, execution and debugging. Create a new file using gedit/vi filename.c</i>
a.	<p>Write C statement(s) in the file which produces the following output:</p> <p>a) Have a Good day! b) Have/ \a / \Good / \ day!  c) Laughter is the best “Medicine”. Share the “Knowledge”.  d) Hello  World! (Using a only one printf statement)  e) Hello  World! (Using a single printf statement that has no blank space)  f) How are you?  I am Fine.  g) How are you?  I am Fine. (Using two printf statements which have no blank spaces)  h) How are you?  I am Fine. Thank You. (Using a single printf statement that has no blank space)  i) Bank interest is 9% in the year 2018.</p>
Soln	<pre>#include &lt;stdio.h&gt; int main() { printf("Have a Good day!\n"); //a printf("Have \a /\Good /\ day!\n"); //b printf("Laughter is the best \"Medicine\" .Share the \"Knowledge\" \n"); //c printf("Hello\n world!\n"); //d printf("Hello\vworld!\n"); //e printf("How are you?\n"); //f printf("I am Fine.\n"); //f printf("How\tare\tyou?\n"); //g printf("I\tam\tFine.\n"); //g printf("How\tare\tyou?\nI\tam\tFine\nThank\tYou\n"); //h printf("Something has gone crazy\a\n"); //i printf("Bank interest is 9%% in the year 2018.\n"); //j return 0; }</pre>
2	<p><i>Implementation and execution of simple programs to understand working of</i></p> <ul style="list-style-type: none"> <li>Formatted input and output functions- printf() and scanf().</li> <li>Escape sequences in C.</li> <li>Using formula in a C program for specific computation: For example: computing area of circle, converting Celsius to Fahrenheit, area of a triangle, converting distance in centimeters to inches, etc.</li> <li>Preprocessor directives (#include, #define).</li> </ul>

a	<p>Write a C program that produces the following output and also test for other Patterns: (Print using * or #)</p> <pre>       * ****       *      *       *      *       *       *       *       *       *      *       *      *       ***** </pre>
Soln	<pre> #include &lt;stdio.h&gt; int main() { printf(" *****\n"); printf(" * *\n"); printf(" * *\n"); printf(" *\n"); printf(" *\n"); printf(" *\n"); printf(" *\n"); printf(" * *\n"); printf(" * *\n"); printf(" *****\n"); return 0; } </pre>
b	<p>Design and develop C program that accepts a distance in centimeters and prints the corresponding value in inches. (Note that 1 inch = 2.54 cm.)</p>
Soln	<pre> #include &lt;stdio.h&gt; #define INCH_TO_CM 2.54 int main() { double inch,cm; printf("Enter the distance in cm:"); scanf("%lf",&amp;cm); inch=cm/INCH_TO_CM; printf("Distance %0.2lf cms is = %0.2lf inches\n",cm,inch); return 0; } </pre>
c	<p>Design and develop a C program to find the area of the following: Triangle, Square, Rectangle and Circle. Implement the C program for all possible inputs appropriate message.</p>
Soln	<pre> #include&lt;stdio.h&gt; #define PI 3.14 int main() { float ba,he,len,bre,ra; </pre>

	<pre> float area_tri,area_squ,area_rect,area_circle; printf("Enter the base and height\n"); scanf("%f%f",&amp;ba,&amp;he); printf("Enter the length and breadth\n"); scanf("%f%f",&amp;len,&amp;bre); printf("Enter the radius\n"); scanf("%f%f",&amp;ra); area_tri=1/2*ba*he; area_squ=len*bre; area_rect=len*bre; area_circle=3.14*r a*ra; printf("Area of Triangle=%f\n",area_tri); printf("Area of Square=%f\n",area_tsqu); printf("Area of Rectangle=%f\n",area_rect); printf("Area of Circle=%f\n",area_circle); return 0; } </pre>
3	<p><i>Execution of erroneous C programs to understand debugging and correcting the errors like:</i></p> <ul style="list-style-type: none"> <li>• <i>Syntax / compiler errors.</i></li> <li>• <i>Run-time errors.</i></li> <li>• <i>Linker errors.</i></li> <li>• <i>Logical errors.</i></li> <li>• <i>Semantical errors.</i></li> </ul>
	<p>Errors in C/C++</p> <p>Error is an illegal operation performed by the user which results in abnormal working of the program.</p> <p>Programming errors often remain undetected until the program is compiled or executed. Some of the errors inhibit the program from getting compiled or executed. Thus errors should be removed before compiling and executing.</p> <p>The most common errors can be broadly classified as follows.</p> <div style="text-align: center;"> <p>Type of errors</p>  <pre> graph TD     A[Errors in C / C++] --&gt; B[Syntax error]     A --&gt; C[Run-time error]     A --&gt; D[Linker error]     A --&gt; E[Logical error]     A --&gt; F[Semantic error] </pre> </div> <p>1. Syntax errors: Errors that occur when you violate the rules of writing C/C++ syntax are known as syntax errors. This compiler error indicates something that must be fixed before the code can be compiled. All these errors are detected by compiler and thus are known as compile-time errors.</p> <p>Most frequent syntax errors are:</p> <ul style="list-style-type: none"> <li>• Missing Parenthesis ( )</li> <li>• Printing the value of variable without declaring it</li> <li>• Missing semicolon like this</li> </ul>

```
// C program to illustrate syntax error
a)
#include<stdio.h>
void main()
{   int x = 10;
    int y = 15;
    printf("%d", (x, y)) // semicolon missed
}
```

Error:  
error: expected ';' before '}' token

```
b)
#include<stdio.h>
int main(void)
{
    // while() cannot contain "." as an argument.
    while(.)
    {   printf("hello");
    }
    return 0;
}
```

Error:  
error: expected expression before '.' token  
while(.)

2. Run-time Errors : Errors which occur during program execution(run-time) after successful compilation are called run-time errors. One of the most common run-time error is division by zero also known as Division error. These types of error are hard to find as the compiler doesn't point to the line at which the error occurs.

```
// C program to illustrate run-time error
#include<stdio.h>
void main()
{   int n = 9, div = 0;
    // wrong logic
    // number is divided by 0,
    // so this program abnormally terminates
    div = n/0;
    printf("resut = %d", div);
}
```

Error:  
warning: division by zero [-Wdiv-by-zero]  
div = n/0;

3. Linker Errors: These error occurs when after compilation we link the different object files with main's object using Ctrl+F9 key(RUN). These are errors generated when the executable of the program cannot be generated. This may be due to wrong

	<p>function prototyping, incorrect header files. One of the most common linker error is writing Main() instead of main().</p> <pre>// C program to illustrate // linker error #include&lt;stdio.h&gt; void Main() // Here Main() should be main() { int a = 10;   printf("%d", a); }</pre> <p>Error: (.text+0x20): undefined reference to `main'</p> <p>4. Logical Errors : On compilation and execution of a program, desired output is not obtained when certain input values are given. These types of errors which provide incorrect output but appears to be error free are called logical errors. These are one of the most common errors done by beginners of programming. These errors solely depend on the logical thinking of the programmer and are easy to detect if we follow the line of execution and determine why the program takes that path of execution.</p> <pre>// C program to illustrate logical error int main() {  int i = 0;     // logical error : a semicolon after loop     for(i = 0; i &lt; 3; i++);     { printf("loop ");       continue;     }     getchar();     return 0; }</pre> <p>No output</p> <p>5. Semantic errors : This error occurs when the statements written in the program are not meaningful to the compiler.</p> <pre>// C program to illustrate semantic error void main() {  int a, b, c;     a + b = c; //semantic error }</pre> <p>Error error: lvalue required as left operand of assignment a + b = c; //semantic error</p>
a.	<p>Find the errors in the following program:/* Is it a C program?*/</p> <pre>#include &lt;stdio.h&gt;</pre>

	<pre> int main( ) {     int a,b ,c;     a= 5.54;     b =a+2;     scanf( "%d", &amp;c);     printf( "%d %d %d\n",a, b,c);     return 0; } </pre>
b.	<p>Find the errors in the following program:</p> <pre> /* There are errors in /*the*/ code.*/ #include &lt;stdio.h&gt; #define CUBE(Y) (Y)*(Y)*(Y) ; #define SQUARE(Y) (Y)*(Y); int main() { double Int,y,_y1,2yb,a-b,y3z;   int Float,char,a,b,c,d,xy@c,qa.b;   char int,u,_2v,w=t;   a=2,b=3;   a+b;   c=a+b;   a+b=1;   b-a==c;   d=w;   a=CUBE(d);   b=SQUARE(d)   u=d+62;   c=u-1;   u='y';   _2v=z;   y3z=CUBE(c);   y=SQUARE(c);   _y1=SQUARE(c)*2;   c=y+u;   return 0; } </pre> <p>check it in the question variable names</p>
Soln	<p>An error free code for the same is as follows</p> <pre> /* There are errors in the code.* / #include &lt;stdio.h&gt; #define CUBE(Y) (Y)*(Y)*(Y) #define SQ(Y) (Y)*(Y) int main() {     double Int,y,_y1,2yb,a_b,y3z;     int Float,cHar,a,b,c,d,xy_c,qab;     char inT,u,_2v,w='t'; } </pre>

	<pre> a=2,b=3; a+b; c=a+b; a+b==1; b-a==c; d=w; a=CUBE(d); b=SQ(d); u=d+62; c=u-1; u='y'; _2v='z'; y3z=CUBE(c); x=SQ(c); _x1=2*SQ(c); c=x+u; return 0; } </pre>
3.c	<p>Type the program in a file and examine the output of the following code:.</p> <pre> #include&lt;stdio.h&gt; int main(void) { int a=234,b= -234,c=54321; printf("%2d\n",c); printf("%10.2d\n",c); printf("%-10.2d\n",c); printf("%-7d\n",a); printf("%07.2d\n",a); printf("%07d\n",a); printf("%+0-9.4d\n",a); printf("%+09.4d\n",a); printf("%+07d\n",a); printf("%+07.4d\n",a); printf("%+-07.4d\n",a); printf("%-08d\n",b); printf("%-08.2d\n",b); printf("%-8.4d\n",b); return 0; } </pre>
4	<p><i>Implementation and execution of simple programs to understand working of operators like:</i></p> <ul style="list-style-type: none"> <li>• <i>Unary.</i></li> <li>• <i>Arithmetic.</i></li> <li>• <i>Logical.</i></li> <li>• <i>Relational.</i></li> </ul>



	<ul style="list-style-type: none"> <li>• <i>Conditional.</i></li> <li>• <i>Bitwise.</i></li> </ul>
a	<p><u>C program to demonstrate working of Unary arithmetic operators</u></p> <pre>#include &lt;stdio.h&gt; int main() {     int a = 10, b = 4, res;     // post-increment example:     // res is assigned 10 only, a is not updated yet     res = a++;     printf("a is %d and res is %d\n", a, res); // a becomes 11 now     // post-decrement example:     // res is assigned 11 only, a is not updated yet     res = a--;     printf("a is %d and res is %d\n", a, res); // a becomes 10 now     // pre-increment example:     // res is assigned 11 now since a is updated here itself     res = ++a;     // a and res have same values = 11     printf("a is %d and res is %d\n", a, res);     // pre-decrement example:     // res is assigned 10 only since a is updated here itself     res = --a;     // a and res have same values = 10     printf("a is %d and res is %d\n", a, res);     return 0; }</pre> <p><b>Output:</b>  a is 11 and res is 10  a is 10 and res is 11  a is 11 and res is 11  a is 10 and res is 10</p>
b	<p>C program to demonstrate working of relational operators</p> <pre>#include &lt;stdio.h&gt; int main() {     int a = 10, b = 4;     // greater than example     if (a &gt; b)         printf("a is greater than b\n");     else         printf("a is less than or equal to b\n");      // greater than equal to     if (a &gt;= b)         printf("a is greater than or equal to b\n");     else</pre>

	<pre> printf("a is lesser than b\n");  // less than example if (a &lt; b)     printf("a is less than b\n"); else     printf("a is greater than or equal to b\n");  // lesser than equal to if (a &lt;= b)     printf("a is lesser than or equal to b\n"); else     printf("a is greater than b\n"); // equal to if (a == b)     printf("a is equal to b\n"); else     printf("a and b are not equal\n");  // not equal to if (a != b)     printf("a is not equal to b\n"); else     printf("a is equal b\n");  return 0; } Output: a is greater than b a is greater than or equal to b a is greater than or equal to b a is greater than b a and b are not equal a is not equal to b </pre>
c	<p>C program to demonstrate working of logical operators</p> <pre> #include &lt;stdio.h&gt; int main() {     int a = 10, b = 4, c = 10, d = 20;     // logical operators     // logical AND example     if (a &gt; b &amp;&amp; c == d)         printf("a is greater than b AND c is equal to d\n");     else         printf("AND condition not satisfied\n"); } </pre>

	<pre> // logical OR example if (a &gt; b    c == d)     printf("a is greater than b OR c is equal to d\n"); else     printf("Neither a is greater than b nor c is equal "         " to d\n");  // logical NOT example if (!a)     printf("a is zero\n"); else     printf("a is not zero");  return 0; } </pre> <p>Output:  AND condition not satisfied  a is greater than b OR c is equal to d  a is not zero</p>
	<p><b>Note: Following additional exercises can be carried out by students who finish above programs</b></p> <p>Implement a C Program to demonstrate the working of relational operator, logical and bitwise operator. Print the value for the following expressions and analyze the output.</p> <p>a)a=a+b+c    b)d= a+b+c    c)a=(b+c)*d    d)a=a&amp;&amp;b  c  e)a=!a&amp;&amp;b  c!  d&amp;&amp;e f) a= (b&amp;&amp;c)!d g)a=a++ + a++ h)b=a++ +a++  i) a= ++a + a++</p>

	Lab Programs
1.	Develop a C program to compute the roots of the equation $ax^2 + bx + c = 0$ .
sol	<pre> #include&lt;stdio.h&gt; #include&lt;math.h&gt; #include&lt;stdlib.h&gt; main() {     float a,b,c,root,disc,root1,root2;     printf("\nEnter the coefficients:\n");     scanf("%f%f%f",&amp;a,&amp;b,&amp;c);      if(a==0 &amp;&amp; c==0)     {         printf("Not Possible\n");         exit(0);     }     else if(a==0 &amp;&amp; b!=0)     {         root=-c/b;         printf("Linear root =%f\n",root);     }     else     {         disc=b*b-4*a*c;           // 'disc' indicates discriminant         //Find the distinct roots         if(disc&gt;0)         {             root1=(-b + sqrt(disc)) / (2*a);             root2=(-b - sqrt(disc)) / (2*a);             printf("\n Roots are real &amp; distinct! \n");             printf("\n The roots are: \n%f\n%f\n",root1,root2);         }          else if(disc==0)    //Find the equal roots         {             root1=root2= -b / (2*a);             printf("\n Roots are real &amp; equal! \n");             printf("\n The roots are \n%f\n%f\n",root1,root2);         }         else         {             //Find the complex roots             root1= -b / (2*a);             root2= sqrt(abs(disc)) / (2*a);             printf("\n The roots are imaginary!\n");             printf("\n The first root is %f + i%f \n",root1,root2);             printf("\n The second root is %f - i%f \n",root1,root2);         }     } } </pre>

	<p>-----</p> <p><b><u>Output</u></b></p> <p><b>Run 1:</b></p> <p>Enter the coefficients:</p> <p>1</p> <p>2</p> <p>1</p> <p>Roots are real &amp; equal!</p> <p>The roots are</p> <p>-1.000000</p> <p>-1.000000</p> <p><b>Run 2:</b></p> <p>Enter the coefficients:</p> <p>1</p> <p>5</p> <p>4</p> <p>Roots are real &amp; distinct!</p> <p>The roots are:</p> <p>-1.000000</p> <p>-4.000000</p> <p><b>Run 3:</b></p> <p>Enter the coefficients:</p> <p>2</p> <p>4</p> <p>6</p> <p>The roots are imaginary!</p> <p>The first root is   -1.000000 + i1.414214</p> <p>The second root is -1.000000 - i1.414214</p>
2	Develop a C program that reads N integer numbers and arrange them in ascending or descending order using selection sort and bubble sort technique.
soln	<pre>//SELECTION SORT #include &lt;stdio.h&gt; #define MAX 100 int main() {     int arr[MAX],i,j,n,temp,min;     printf("Enter the number of elements : ");     scanf("%d", &amp;n);     for(i=0; i&lt;n; i++) {         printf("Enter element %d : ",i+1);         scanf("%d", &amp;arr[i]);     } }</pre>

```
}  
for(i=0; i<n-1; i++) {  
    min = i;  
    for(j=i+1; j<n; j++) {  
        if(arr[min]>arr[j])  
            min=j;  
    }  
    if(i!=min) {  
        temp=arr[i];  
        arr[i]=arr[min];  
        arr[min]=temp;  
    }  
}  
printf("Sorted list is : \n");  
for(i=0; i<n; i++) {  
    printf(" %d", arr[i]);  
}  
printf("\n");  
}
```

# **//BUBBLE SORT**

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
    int a[10],n,i,j,temp;
```

```
    printf("\n Enter the size of n:\n" );
```

```
    scanf("%d",&n);
```

```
    printf("\n Enter the array elements: \n");
```

```
    for(i=0;i<n;i++)
```

```
    scanf("%d",&a[i]);
```

```
    for(i=1;i<n;i++)
```

**/\* i<sup>th</sup> smallest number bubbles up to its  
right spot in ith iteration \*/**

```
{
```

```
    for(j=0;j<n-i;j++)
```

**/\* bubbling starts from the "deepest numbers"  
and proceeds upwards \*/**

**/\* element at jth position is "lighter" than the one on top,  
therefore jth element bubbles up \*/**

```
    if(a[j]>=a[j+1])
```

```
{
```

```
        temp=a[j];
```

```
        a[j]=a[j+1];
```

```
        a[j+1]=temp;
```

```
}
```

	<pre> } printf("\n The sorted array is: \n" ); for(i=0;i&lt;n;i++) printf("%d\n",a[i]); } </pre> <p>-----</p> <p><b><u>Output (SELECTION SORT)</u></b></p> <p><b><u>Run1:</u></b>  Enter Number of elements  6  Enter 6 elements  12 89 980 9 0 8 18  Sorted elements:  0 9 12 18 89 980</p> <p><b><u>Output(BUBBLE SORT)</u></b></p> <p><b>Run 1:</b>  Enter the size of n:  5</p> <p>Enter the array elements:  65  84  91  20  8</p> <p>The sorted array is:  8  20  65  84  91</p>
3	Develop a C program for Matrix multiplication.
	<pre> #include &lt;stdio.h&gt;  int main() {     int a[10][10], b[10][10], result[10][10], r1, c1, r2, c2, i, j, k;      printf("Enter rows and column for first matrix: ");     scanf("%d %d", &amp;r1, &amp;c1);      printf("Enter rows and column for second matrix: ");     scanf("%d %d",&amp;r2, &amp;c2);      // Column of first matrix should be equal to column of second matrix and     while (c1 != r2)     {         printf("Error! column of first matrix not equal to row of second.\n\n");     } } </pre>



```

printf("Enter rows and column for first matrix: ");
scanf("%d %d", &r1, &c1);
printf("Enter rows and column for second matrix: ");
scanf("%d %d",&r2, &c2);
}

```

**// Storing elements of first matrix.**

```

printf("\nEnter elements of matrix 1:\n");
for(i=0; i<r1; ++i)
    for(j=0; j<c1; ++j)
    {
        printf("Enter elements a%d%d: ",i+1, j+1);
        scanf("%d", &a[i][j]);
    }

```

**// Storing elements of second matrix.**

```

printf("\nEnter elements of matrix 2:\n");
for(i=0; i<r2; ++i)
    for(j=0; j<c2; ++j)
    {
        printf("Enter elements b%d%d: ",i+1, j+1);
        scanf("%d",&b[i][j]);
    }

```

**// Initializing all elements of result matrix to 0**

```

for(i=0; i<r1; ++i)
    for(j=0; j<c2; ++j)
    {
        result[i][j] = 0;
    }

```

**// Multiplying matrices a and b and**

**// storing result in result matrix**

```

for(i=0; i<r1; ++i)
    for(j=0; j<c2; ++j)
        for(k=0; k<c1; ++k)
        {
            result[i][j]+=a[i][k]*b[k][j];
        }

```

**// Displaying the result**

```

printf("\nOutput Matrix:\n");
for(i=0; i<r1; ++i)
    for(j=0; j<c2; ++j)
    {
        printf("%d ", result[i][j]);
        if(j == c2-1)
            printf("\n\n");
    }
return 0;
}

```

	<p>-----</p> <p><b><u>Output</u></b></p> <p><b>Run 1:</b>  Enter rows and column for first matrix: 3  2  Enter rows and column for second matrix: 3  2  Error! column of first matrix not equal to row of second.</p> <p>Enter rows and column for first matrix: 2  3  Enter rows and column for second matrix: 3  2</p> <p>Enter elements of matrix 1:  Enter elements a11: 3  Enter elements a12: -2  Enter elements a13: 5  Enter elements a21: 3  Enter elements a22: 0  Enter elements a23: 4</p> <p>Enter elements of matrix 2:  Enter elements b11: 2  Enter elements b12: 3  Enter elements b21: -9  Enter elements b22: 0  Enter elements b31: 0  Enter elements b32: 4</p> <p>Output Matrix:  24 29    6 25</p>
4	Develop a C program to search an element using Binary search and linear search techniques.
	<pre>// <b>BINARY SEARCH</b> #include &lt;stdio.h&gt; int main() {     int c, first, last, middle, n, search, array[100];     printf("Enter number of elements\n");     scanf("%d", &amp;n);     printf("Enter %d integers\n", n);     for (c = 0; c &lt; n; c++)         scanf("%d", &amp;array[c]);     printf("Enter value to find\n");     scanf("%d", &amp;search);     first = 0;     last = n - 1;     middle = (first+last)/2;</pre>

```

while (first <= last) {
    if (array[middle] < search)
        first = middle + 1;
    else if (array[middle] == search) {
        printf("%d found at location %d.\n", search, middle+1);
        break;
    }
    else
        last = middle - 1;
    middle = (first + last)/2;
}
if (first > last)
    printf("Not found! %d isn't present in the list.\n", search);

return 0;
}

```

### **//LINEAR SEARCH**

```

#include <stdio.h>
int main()
{
    int array[100], search, c, n;

    printf("Enter number of elements in array\n");
    scanf("%d", &n);
    printf("Enter %d integer(s)\n", n);
    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);
    printf("Enter a number to search\n");
    scanf("%d", &search);
    for (c = 0; c < n; c++)
    {
        if (array[c] == search) /* If required element is found */
        {
            printf("%d is present at location %d.\n", search, c+1);
            break;
        }
    }
    if (c == n)
        printf("%d isn't present in the array.\n", search);

    return 0;
}

```

### **OUTPUT**

#### **BINARY SEARCH**

Enter number of elements

7

Enter 7 integers

-4 5 8 9 11 43 485

Enter the value to find

11

11 found at location 5.

	<b>LINEAR SEARCH</b> Enter the number of elements in array 5 Enter 5 numbers 5 6 4 2 9 Enter the number to search 4 4 is present at location
5	Using functions develop a C program to perform the following tasks by parameter passing to read a string from the user and print appropriate message for palindrome or not palindrome
sol	<pre> #include &lt;stdio.h&gt; #include &lt;string.h&gt; void check(char [], int); int main() { char word[15]; printf("Enter a string to check if it is a palindrome\n"); scanf("%s", word); check(word, 0); return 0; } void check (char word [], int index) { intlen = strlen(word) - (index + 1); if (word[index] == word[len]) { if (index + 1 == len    index == len) { printf("The entered word is a palindrome\n"); return; } check(word, index + 1); } else { printf("The entered word is not a palindrome\n"); } } } </pre> <hr/> <p><b><u>Output</u></b></p> <p><b>Run 1:</b></p> <p>Enter a string to check if it is a palindrome</p> <p>How are you</p> <p>The entered word is not a palindrome</p>

	<p><b>Run 2:</b> Enter a string to check if it is a palindrome</p> <p>Madam</p> <p>The entered word is a palindrome.</p> <p><b>Run 3:</b> Enter a string to check if it is a palindrome</p> <p>mam</p> <p>The entered word is a palindrome.</p>
6	Develop a C program to compute average marks of 'n' students (Name, Roll_No, Test Marks) and search a particular record based on 'Roll_No'.
sol	<pre> #include&lt;stdio.h&gt; #include&lt;string.h&gt; struct student {     char name[50];     int roll;     float marks[3]; }st[60]; int main() {     int i,n,rollno;     float sum=0,avg;     printf("Enter the number of students\n");     scanf("%d",&amp;n);     for(i=0;i&lt;n;i++)     {         printf("Enter The name of %d student\n",i+1);         scanf("%s",st[i].name);         printf("Enter The Roll no of %d student\n",i+1);         scanf("%d",&amp;st[i].roll);         printf("Enter The Marks of %d student\n",i+1);          scanf("%f%f%f",&amp;st[i].marks[0],&amp;st[i] .marks[1],&amp;st[i].marks[2]);     }     for(i=0;i&lt;n;i++)     {         sum= sum+ st[i].marks[0] + st[i].marks[1] + st[i].marks[2];     }     avg= sum/n;     printf("The Average Marks Of %d Students is : %f\n",n,avg);     printf("Enter the roll no. of student detail required : "); </pre>

```
scanf("%d",&rollno);
for(i=0;i<n;i++)
{
    if(st[i].roll==rollno)
    {
        printf("The name of
student is : %s\n",st[i].name);
        printf("The 3 test
scores are :\n %f %f
%f",st[i].marks[0],st[i].marks[1],st[i].marks[2]
);
        break;
    }
    else
        continue;
}
if(i==n)
    printf("Roll No. not found\n");
}
```

	<p><b><u>OUTPUT</u></b></p> <p>enter the name of 1 student= Ramesh</p> <p>enter the roll no of 1 student= 43</p> <p>enter the 3 test scores of 1 student= 45 48 50</p> <p>enter the name of 2 student= Ganesh</p> <p>enter the roll no of 2 student= 23</p> <p>enter the 3 test scores of 2 student= 35 38 40</p> <p>enter the name of 3 student= Vagesh</p> <p>enter the roll no of 3 student= 50</p> <p>enter the 3 test scores of 3 student= 41 43 45</p> <p>enter the rollno of the student details required= 23</p> <p>the name of the student is= Ganesh</p> <p>the 3 test scores of the student= 35 38 40</p>
7	<p>Develop a C program using pointers to function to find given two strings are equal or not.</p> <pre> #include&lt;stdio.h&gt; int compare_string(char*, char*); main() {     char first[100], second[100], result;     printf("Enter first string\n");     gets(first);     printf("Enter second string\n");     gets(second);     result = compare_string(first, second);     if ( result == 0 )         printf("Both strings are same.\n");     else         printf("Entered strings are not equal.\n");     return 0; } int compare_string(char *first, char *second) {     while(*first==*second)     {         if ( *first == '\0'    *second == '\0' )             break;          first++;         second++;     } } </pre>

	<pre>     }     if( *first == '\0' &amp;&amp; *second == '\0' )         return 0;     else         return -1;     } </pre> <hr/> <p><b>OUTPUT</b></p> <p>Run 1</p> <p>Enter first string</p> <p>RVCE</p> <p>Enter second string</p> <p>BANGALORE</p> <p>Entered strings are not equal.</p> <p>Run 2</p> <p>Enter first string</p> <p>CSE</p> <p>Enter second string</p> <p>CSE</p> <p>Both strings are same.</p>
8	Develop a C program using recursion, to determine GCD , LCM of two numbers and to perform binary to decimal conversion.
sol	<p><b>(a)GCD and LCM</b></p> <pre> #include &lt;stdio.h&gt; int find_gcd(int,int); int find_lcm(int,int); int main() {     int num1,num2,gcd,lcm;     printf("\nEnter two numbers:\n ");     scanf("%d %d",&amp;num1,&amp;num2);     gcd=find_gcd(num1,num2);      printf("\nGCD of %d and %d is: %d\n",num1,num2,gcd);      if(num1&gt;num2)         lcm = find_lcm(num1,num2);     else         lcm = find_lcm(num2,num1);      printf("\nLCM of %d and %d is: %d\n",num1,num2,lcm);     return 0; }  int find_gcd(int n1,int n2){ while(n1!=n2){     if(n1&gt;n2)         return find_gcd(n1-n2,n2);     else         return find_gcd(n1,n2-n1);     }     return n1; } </pre>



	<pre> int find_lcm(int n1,int n2){  static int temp = 1;  if(temp % n2 == 0 &amp;&amp; temp % n1 == 0) return temp; temp++; find_lcm(n1,n2);  return temp; } </pre> <p><b>(b) Binary to Decimal Conversion</b></p> <pre> #include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; #include &lt;math.h&gt;  int binary_to_decimal(int binum,int decnum,int bit) { int bitwt; if(binum&gt;0) { bitwt=binum% 10; decnum=decnum+bitwt*pow(2,bit); binum=binum/10; bit++; decnum=binary_to_decimal(binum,decnum,bit);//recursion taking place. } return decnum; }  void main () { int decimalnum=0,binarynum,bitweight=0; printf("Enter the binary number\n"); scanf("%d",&amp;binarynum); decimalnum=binary_to_decimal( binarynum,decimalnum,bitweight); printf("%d in binary %d",decimalnum, binarynum); } </pre> <hr/> <p><b>Output:</b></p> <p>Enter two numbers: 366 60  GCD of 366 and 60 is: 6  L.C.M of 366 and 60 is 3660.</p> <p>Enter a binary number:  110110111 110110111 in  binary = 439</p>
9	<ol style="list-style-type: none"> <li>Program to calculate the sum of n numbers entered by the user using malloc().</li> <li>Program to calculate the sum of n numbers entered by the user using calloc().</li> <li>Program in C using realloc().</li> </ol>

Sol.	<p><b>(a)</b></p> <pre> #include &lt;stdio.h&gt; #include &lt;stdlib.h&gt;  int main() {     int n, i, *ptr, sum = 0;      printf("Enter number of elements: ");     scanf("%d", &amp;n);      ptr = (int*) malloc(n * sizeof(int));      // if memory cannot be allocated     if(ptr == NULL) {         printf("Error! memory not allocated.");         exit(0);     }      printf("Enter elements: ");     for(i = 0; i &lt; n; ++i) {         scanf("%d", ptr + i);         sum += *(ptr + i);     }      printf("Sum = %d", sum);      // deallocating the memory     free(ptr);      return 0; } </pre> <hr/> <p><b>Output</b></p> <pre> Enter number of elements: 3 Enter elements: 100 20 36 Sum = 156 </pre> <p><b>(b)</b></p> <pre> #include &lt;stdio.h&gt; #include &lt;stdlib.h&gt;  int main() {     int n, i, *ptr, sum = 0;     printf("Enter number of elements: "); </pre>

```
scanf("%d", &n);

ptr = (int*) calloc(n, sizeof(int));
if(ptr == NULL) {
    printf("Error! memory not allocated.");
    exit(0);
}

printf("Enter elements: ");
for(i = 0; i < n; ++i) {
    scanf("%d", ptr + i);
    sum += *(ptr + i);
}

printf("Sum = %d", sum);
free(ptr);
return 0;
}
```

---

### Output

```
Enter number of elements: 3
Enter elements: 100
20
36
Sum = 156
```

(c)

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int *ptr, i, n1, n2;
    printf("Enter size: ");
    scanf("%d", &n1);

    ptr = (int*) malloc(n1 * sizeof(int));

    printf("Addresses of previously allocated memory:\n");
    for(i = 0; i < n1; ++i)
        printf("%pc\n", ptr + i);

    printf("\nEnter the new size: ");
    scanf("%d", &n2);

    // relocating the memory
    ptr = realloc(ptr, n2 * sizeof(int));

    printf("Addresses of newly allocated memory:\n");
```

	<pre> for(i = 0; i &lt; n2; ++i)     printf("%pc\n", ptr + i);  free(ptr);  return 0; } </pre> <hr/> <p><b>Output</b></p> <p>Enter size: 2  Addresses of previously allocated memory:  26855472  26855476</p> <p>Enter the new size: 4  Addresses of newly allocated memory:  26855472  26855476  26855480  26855484</p>
10	C Program to implement Singly Linked List to insert and delete a node from front and display the contents of the Singly List.

Sol

```
#include<stdio.h>
#include<malloc.h>
#include<stdlib.h>

struct node
{
    int info;
    struct node *link;
};

typedef struct node* NODE;

NODE getnode();
NODE insert_front(NODE , int);
NODE delete_front(NODE);
void display(NODE);

void main()
{
    NODE first;
    int choice, item;

    first = NULL;

    while(1)
    {
        printf("Enter\n");
        printf("1. Insert Front\n");
        printf("2. Delete Front\n");
        printf("3. Display the list\n");
        printf("4. Exit\n");

        scanf("%d", &choice);

        switch(choice)
        {
            case 1:
                printf("Enter item to be inserted\n");
                scanf("%d", &item);
                first = insert_front(first, item);

                break;

            case 2:
                first = delete_front(first);

                break;

            case 3:
                display(first);
```

```

                                break;

                                default:
                                    exit(0);
                                }
                            }
    }

NODE getnode()
{
    NODE x;

    x = (NODE) malloc(sizeof(struct node));

    if(x == NULL)
    {
        printf("Node creation error\n");
        return;
    }

    return x;
}

NODE insert_front(NODE first , int item)
{
    NODE temp;

    temp = getnode();
    temp->info = item;

    temp->link = first;

    return temp;
}

NODE delete_front(NODE first)
{
    NODE temp;

    if(first == NULL)
    {
        printf("Cannot delete. Empty List\n");
        return first;
    }

    temp = first;
    first = first->link;

    printf("Deleted node is %d\n", temp->info);
}

```

```

        free(temp);

        return first;
    }

void display(NODE first)
{
    NODE temp;

    printf("Contents of linked list is:\n");

    if(first == NULL)
    {
        printf("Cannot print. Empty list\n");
        return;
    }

    temp = first;

    while(temp != NULL)    //as long as there are elemens in the linked list
    {
        printf("%d\t", temp->info);
        temp = temp->link;
    }

    printf("\n");
}

```

---

### Output

Enter

1. Insert Front
2. Delete Front
3. Display the list
4. Exit

2

Cannot delete. Empty List

Enter

1. Insert Front
2. Delete Front
3. Display the list
4. Exit

3

Contents of linked list is:

Cannot print. Empty list

Enter

1. Insert Front
2. Delete Front
3. Display the list
4. Exit

1

Enter item to be inserted

10

Enter

1. Insert Front
2. Delete Front
3. Display the list
4. Exit

1

Enter item to be inserted

20

Enter

1. Insert Front
2. Delete Front
3. Display the list
4. Exit

1

Enter item to be inserted

30

Enter

1. Insert Front
2. Delete Front
3. Display the list
4. Exit

1

Enter item to be inserted

40

Enter

1. Insert Front
2. Delete Front
3. Display the list
4. Exit

1

Enter item to be inserted

50

Enter

1. Insert Front
2. Delete Front
3. Display the list
4. Exit

3

Contents of linked list is:

50      40      30      20      10

Enter

1. Insert Front
2. Delete Front
3. Display the list
4. Exit

2

Deleted node is 50

Enter

1. Insert Front



	<p>2. Delete Front</p> <p>3. Display the list</p> <p>4. Exit</p> <p>2</p> <p>Deleted node is 40</p> <p>Enter</p> <p>1. Insert Front</p> <p>2. Delete Front</p> <p>3. Display the list</p> <p>4. Exit</p> <p>2</p> <p>Deleted node is 30</p> <p>Enter</p> <p>1. Insert Front</p> <p>2. Delete Front</p> <p>3. Display the list</p> <p>4. Exit</p> <p>3</p> <p>Contents of linked list is:</p> <p>20        10</p>
11	Write a C program to count no of lines, blank lines and comments in a given program using files.
Sol	<pre> #include &lt;stdio.h&gt; void main() {     int line_count = 0, n_o_c_l = 0, n_o_n_b_l = 0, n_o_b_l = 0, n_e_c = 0;     FILE *fp1;     char ch;     fp1 = fopen("file.txt", "r");      while (((ch = fgetc(fp1)))!= EOF)     {         if (ch == '\n')         {             line_count++;         }         if (ch == '\n')         {             if ((ch = fgetc(fp1)) == '\n')             {                 fseek(fp1, -1, 1);                 n_o_b_l++;             }         }     } } </pre>

	<pre> if (ch == ';') {     if ((ch = fgetc(fp1)) == '\n')     {         fseek(fp1, -1, 1);         n_e_c++;     } } fseek(fp1, 0, 0); while ((ch = fgetc(fp1))!= EOF) {     if (ch == '/')     {         if ((ch = fgetc(fp1)) == '/')         {             n_o_c_l++;         }     } } printf("Total no of lines: %d\n", line_count); printf("Total no of comment line: %d\n", n_o_c_l); printf("Total no of blank lines: %d\n", n_o_b_l); printf("Total no of non blank lines: %d\n", line_count-n_o_b_l); printf("Total no of lines end with semicolon: %d\n", n_e_c); } input file contents: file.txt  department of CSE RVCE //this is a comment </pre> <hr/> <p><b>Output:</b></p> <pre> Total no of lines: 5 Total no of comment line:1 Total no of blank lines:2 Total no of non blank lines:3 Total no of lines end with semicolon:0 </pre>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## PART B

- Design and development of a working model using any of the following combination of hardware and software.
- Develop a model that helps the user to monitor weather, health condition, environment parameters, etc. using Arduino board.
- Develop a simple Robot that can assist the user to perform simple activities home sanitization, lifting things etc. using Raspberry pi.
- Hardware interfacing (**Arduinio Board, Finch, Lego WeDo 2.0**) with scratch to design various models to solve simple problems.
- Develop applications using Nvidia Jetson Kit.

## Laboratory Evaluation

Each program is evaluated for 10 marks.

Lab Write-up & Execution rubrics (Max: 6 marks) and Viva Voce rubrics (Max: 4 marks)

Lab Write-up and Execution rubrics (Max: 6 marks)						
Sl no	Criteria	Measuring methods	Excellent	Good	Poor	CO
1	<b>Understanding of problem and requirements</b>  (2 Marks)	Observations	Student exhibits thorough understanding of program requirements and applies C concepts. (2M)	Student has sufficient understanding of program requirements and applies C concepts. (1.5M - 1M)	Student does not have clear understanding of program requirements and is unable to apply C concepts. (0M)	CO 1
2	<b>Design &amp; Execution</b>  (2Marks)	Observations	Student demonstrates the execution of the program with optimized code with all the necessary conditions and test cases handled. (2M)	Student demonstrates the execution of the program without optimization of the code and handles only few test cases. (1M)	Student has not executed the program. (0 M)	CO 3 CO 4
3	<b>Results and Documentation</b>  (2Marks)	Observations	Documentation with appropriate comments and output is covered in data sheets and record. (2M)	Documentation with only few comments and only few output cases is covered in data sheets and record. (1M)	Documentation with no comments and no output cases is covered in data sheets and record. (0 M)	CO 3
Viva Voce rubrics (Max: 4 marks)						
1	<b>Conceptual Understanding</b>  (2 Marks)	Viva Voce	Explains C and related concepts involved. (2M)	Adequately explains the C and related concepts involved. (1M)	Unable to explain the concepts. (0M)	CO 1
2	<b>Use of appropriate Design Techniques</b> (1 Mark)	Viva Voce	Insightful explanation of appropriate design techniques for the given problem to derive solution. (1 M)	Sufficiently explains the use of appropriate design techniques for the given problem to derive solution. (0.5 M)	Unable to explain the design techniques for the given problem. (0 M)	CO 2
3	<b>Communication of Concepts</b> (1 Mark)	Viva Voce	Communicates the concept used in problem solving well.	Sufficiently communicates the concepts used in problem solving.	Unable to communicate the concepts used in problem.	CO 3, C

			(1 M)	(0.5 M)	(0 M)	O4
--	--	--	-------	---------	-------	----

**Part B program will be evaluated for 10 marks and rubrics is :**

<b>PART B Rubrics</b>						
<b>Sl no</b>	<b>Criteria</b>	<b>Measuring methods</b>	<b>Excellent</b>	<b>Good</b>	<b>Poor</b>	<b>CO</b>
1	<b>Identify and Understand the problem</b>	Application developed	Identified and Understood the problem exceptionally <b>(2M)</b>	Identified and Understood the problem moderately <b>(1.5M)</b>	Not identified and well understood the problem <b>(1M)</b>	CO1
2	<b>Design and development</b>	Application developed	Well designed by considering all the constraints <b>(3M)</b>	Moderately designed by considering few constraints <b>(2M)</b>	Not well Designed <b>(1M)</b>	CO2
3	<b>Testing and Analysis</b>	Application developed	Validated the obtained result for all test cases <b>(3M)</b>	Validated the obtained result for few test cases <b>(2M)</b>	Not Validated the obtained result <b>(1M)</b>	CO3 , CO4
4	<b>Demonstration, Documentation and submission</b>	Application developed	Submitted the work on time along with the report. <b>(2M)</b>	Submitted the work along with the report <b>(1M)</b>	Not submitted <b>(0M)</b>	CO4

## **PROGRAM BANK**

### **SET 1**

<b>Sl NO.</b>	<b>Question</b>
1	Distinguish between system software, application software and utility software.

2	<p>Write the algorithm and flowchart to do the following :</p> <p><b>(a)</b> Check whether a year given by the user is a leap year or not.</p> <p><b>(b)</b> Given an integer number in seconds as input, print the equivalent time in hours, minutes, and seconds as output. The recommended output format is something like: 7,322 seconds is equivalent to 2 hours 2 minutes 2seconds.</p> <p><b>(c)</b> Print the numbers that do not appear in the Fibonacci series. The number of terms to be printed should be given by the user.</p> <p><b>(d)</b> Convert the binary equivalent of an integer number.</p> <p><b>(e)</b> Find the prime factors of a number given by the user.</p> <p><b>(f)</b> Check whether a number given by the user is a Krishnamurthy number or not. A Krishnamurthy number is one for which the sum of the factorials of its digits equals the number. For example, 145 is a Krishnamurthy number.</p> <p><b>(g)</b> Print the second largest number of a list of numbers given by the user.</p> <p><b>(h)</b> Find the sum of N odd numbers given.</p> <p><b>(i)</b> Compute the sum of squares of integers from 1 to 50.</p>
3	<p>Which of the following is an incorrect assignment statement?</p> <p>(a) <code>n = m = 0</code></p> <p>(b) <code>value += 10</code></p> <p>(c) <code>mySize = x &lt; y ? 9 : 11</code></p> <p>(d) <code>testVal = (x &gt; 5    x &lt; 0)</code></p> <p>(e) none of the above</p>
4	<p>What will be the output:</p> <p><b>(a)</b></p> <pre>int main() { float c= 3.14; printf("%.2f", c); return 0; }</pre> <p><b>(b)</b></p> <pre>int main() { printf("%d", 'A'); return 0; }</pre>

	<pre> } (c) int main() { double d= 1/2.0 – 1/2; printf(“d=%.2lf”, d); return 0; } (d)int main() { int c = 1; c=c+2*c++; printf(“\n%f”,c); return 0; } </pre>
5.	<p>Use the following values for the next four questions.</p> <pre> int a = 8, b = 3, x1, x2, x3, x4 x1 = a * b x2 = a / b x3 = a % b x4 = a &amp;&amp; b </pre> <p> <b>(a)</b> The value of x1 is:  <b>(b)</b> The value of x2 is  <b>(c)</b> The value of x3 is  <b>(d)</b> The value of x4 is </p>
6	<p>What is the output of this C code?</p> <pre> int main() { char chr; chr = 128; printf(“%d\n”, chr); return 0; } </pre>
7.	<p>How many time the statement will be printed:</p> <pre> #include &lt;stdio.h&gt; int main() { int i = 100; for (; i &gt;= 1) printf("Inside for"); return 0; } </pre>
8.	<p>What is the output of the below program?</p> <pre> #include &lt;stdio.h&gt; int main() { </pre>

	<pre> int i = 0; switch (i) {     case '0': printf("0");                 break;     case '1': printf("1");                 break;     default: printf("Default"); } return 0; } </pre>
9.	<pre> #include &lt;stdio.h&gt; int main() {     int i = 3;     switch (i)     {         case 0+1: printf("1");                     break;         case 1+2: printf("2");                     break;         default: printf("3");     }     return 0; } </pre>
10.	<pre> #include &lt;stdio.h&gt; int i; int main() {     if (i);     else         printf("Else");     return 0; } </pre>
11.	<pre> #include&lt;stdio.h&gt; int main() {     int n;     for (n = 9; n!=0; n--)         printf("n = %d", n--);     return 0; } </pre>
12.	<pre> #include &lt;stdio.h&gt; </pre>

	<pre> int main() {     int c = 5, no = 10;     do {         no /= c;     } while(c--);      printf ("%dn", no);     return 0; } </pre>
13	<pre> #include &lt;stdio.h&gt;  int main() {     int arr[5];     arr[0] = 5;     arr[2] = -10;     arr[3 / 2] = 2; // this is same as arr[1] = 2     arr[3] = arr[0];      printf("%d %d %d %d", arr[0], arr[1], arr[2], arr[3]);      return 0; } </pre>
14	<pre> #include &lt;stdio.h&gt;  int main() {     int arr[2];      printf("%d ", arr[3]);     printf("%d ", arr[-2]);      return 0; } </pre>
15.	<p>Write the program to solve following Problems:</p> <ol style="list-style-type: none"> <li>1. Accept a list of data items and find the second largest and second smallest elements in it</li> <li>2. Copy element of one array into another</li> <li>3. Cyclically permute the elements of an array</li> <li>4. Delete duplicate elements in an array</li> <li>5. Delete the specified integer from the list</li> <li>6. Find unique element in two arrays</li> <li>7. Minimum element location in array</li> </ol>



	8. Accept an array of 10 elements and swap 3rd element with 4th element using pointers
16.	<pre>#include &lt;stdio.h&gt; int main() {     char p;     char buf[10] = { 1, 2, 3, 4, 5, 6, 9, 8};     p = (buf + 1)[5];     printf("%d\n", p);     return 0; }</pre>
17.	<p>Which of the following is true about arrays in C.</p> <p><b>(A)</b> For every type T, there can be an array of T.</p> <p><b>(B)</b> For every type T except void and function type, there can be an array of T.</p> <p><b>(C)</b> When an array is passed to a function, C compiler creates a copy of array.</p> <p><b>(D)</b> 2D arrays are stored in column major form</p>
18.	<p>An array elements are always stored in _____ memory locations.</p> <p>A. Sequential B. Random C. Sequential and Random D. None of the above</p>
19	<pre>void main() {     char str1[] = "abcd";     char str2[] = "abcd";     if(str1==str2)         printf("Equal");     else         printf("Unequal"); }</pre>
20.	<p>What will be the output of the program ?</p> <pre>#include void main() {     float arr[] = { 12.4, 2.3, 4.5, 6.7};     printf("%d", sizeof(arr)/sizeof(arr[0])); }</pre>
21.	The function that is used to find the first occurrence of a given string in another string is:
22.	<pre>#include &lt;stdio.h&gt;  int main() {     char st[] = "CODING";</pre>

	<pre> printf("While printing "); printf(", the value returned by printf() is : %d",         printf("%s", st));  return 0; } </pre>
23	<p>Print the following pattern on the screen</p> <pre> **** ** * ** **** </pre>
24	<pre> #include &lt;stdio.h&gt; int main() {     long int n = 123456789;      printf("While printing ");     printf(", the value returned by printf() is : %d",             printf("%d", n));      return 0; } </pre>
25	<pre> #include &lt;stdio.h&gt; int main() {     char a[100], b[100], c[100];      // scanf() with one input     printf("\n First scanf() returns : %d",             scanf("%s", a));      // scanf() with two inputs     printf("\n Second scanf() returns : %d",             scanf("%s%s", a, b));      // scanf() with three inputs     printf("\n Third scanf() returns : %d",             scanf("%s%s%s", a, b, c));      return 0; } </pre>
26	Write a program to reverse digits of a number.
27	Write a program to reverse an array or string.
28.	List of switch case statement programs in C:

	<p>C program to read weekday number and print weekday name.</p> <p>C program to check whether a character is VOWEL or CONSONANT</p>
29	Examples for sentinel control loop & counter control Loop?
30	<pre>#include &lt;stdio.h&gt; void main() {     int x=22;     if(x=10)         printf("TRUE");     else         printf("FALSE"); }</pre>
31	<p>What will be the output of following program?</p> <pre>#include &lt;stdio.h&gt; void main() {     if(!printf(""))         printf("IF");     else         printf("ELSE"); }</pre>
32	<pre>#include &lt;stdio.h&gt; int main() {     if( (-100 &amp;&amp; 100)   (20 &amp;&amp; -20) )         printf("%s","Condition is true.");     else         printf("%s","Condition is false.");     return 0; }</pre>
33	<pre>#include&lt;stdio.h&gt; int main() {     int i = 5, j = 6, k = 7;     if(i &gt; j == k)         printf("%d %d %d", i++, ++j, --k);     else         printf("%d %d %d", i, j, k);     return 0; }</pre>

34.	<pre> #include &lt;stdio.h&gt; int main() {     int i = 0;     for (i=0; i&lt;20; i++)     {         switch(i)         {             case 0:                 i += 5;             case 1:                 i += 2;             case 5:                 i += 5;             default:                 i += 4;                 break;         }         printf("%d ", i);     }     return 0; } </pre>
35.	<pre> #include&lt;stdio.h&gt; int main() {     int a = 5;     switch(a)     {         default:             a = 4;         case 6:             a--;         case 5:             a = a+1;         case 1:             a = a-1;     }     printf("%d n", a);     return 0; } </pre>
36	<p>Which combination of the integer variables x, y and z makes the variable a get the value 4 in the following expression?</p> $a = (x > y) ? ((x > z) ? x : z) : ((y > z) ? y : z)$

## SET 2

1	Simple computational problems using arithmetic expressions and use of each operator leading to implementation of a commercial calculator with appropriate output.																																																		
2	Write a C program to swap two numbers using pointers (Call by reference).																																																		
3	Create a structure called student with the following members student name, roll no, and a structure with marks details in three tests. Write a C program to create N records and <ol style="list-style-type: none"><li>Search on roll no and display all the records.</li><li>Average marks in each test.</li><li>Highest marks in each test.</li></ol>																																																		
4	Write a C program to check whether the entered year is leap or not.																																																		
5	Design a structure 'Complex' and write a C program to perform the following operations: <ol style="list-style-type: none"><li>Reading a complex number.</li><li>Addition of two complex numbers.</li></ol>																																																		
6	Write a C program to input a number and check whether the number is palindrome or not.																																																		
7	Compute the roots of the equation $ax^2 + bx + c = 0$ and print using three-decimal places. The roots are real $-b \pm \sqrt{D}/2a$ if the discriminant $D = b^2 - 4ac$ is non-negative. If the discriminant is negative, then the roots are complex conjugate $-b/2a \pm \sqrt{-D}/2a$ . The program proceeds in the following steps. <ol style="list-style-type: none"><li>The program should accept the values of a, b and c from the keyboard.</li><li>No solution if both a and b are zero. The program terminates with appropriate message.</li><li>Linear equation if <math>a = 0</math> but <math>b \neq 0</math> and the root is <math>-c/b</math>. The program prints out the root with appropriate message and the program terminates.</li><li>Calculate the discriminant D and determines the corresponding roots.</li><li>Display all possible roots of a quadratic equation with appropriate message.</li></ol>																																																		
8	Write a program to print the prime numbers in a range entered by user.																																																		
9	Design and develop using an iterative problem solving approach for Taylor series approximation for five decimal digits to compute $\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - \dots - X^n/n!$ . Read the value for N.																																																		
10	Write a program to print out a multiplication table as given below. <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr><tr><td>2</td><td>4</td><td>6</td><td>8</td><td>10</td><td>12</td><td>14</td><td>16</td><td>18</td><td>20</td></tr><tr><td>3</td><td>6</td><td>9</td><td>12</td><td>15</td><td>18</td><td>21</td><td>24</td><td>27</td><td>30</td></tr><tr><td>4</td><td>8</td><td>12</td><td>16</td><td>20</td><td>24</td><td>28</td><td>32</td><td>36</td><td>40</td></tr><tr><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td><td>50</td></tr></table>	1	2	3	4	5	6	7	8	9	10	2	4	6	8	10	12	14	16	18	20	3	6	9	12	15	18	21	24	27	30	4	8	12	16	20	24	28	32	36	40	5	10	15	20	25	30	35	40	45	50
1	2	3	4	5	6	7	8	9	10																																										
2	4	6	8	10	12	14	16	18	20																																										
3	6	9	12	15	18	21	24	27	30																																										
4	8	12	16	20	24	28	32	36	40																																										
5	10	15	20	25	30	35	40	45	50																																										
11	Write a C program to read a list of N integer numbers in an array and print the following: <ol style="list-style-type: none"><li>the maximum value</li><li>the minimum value</li><li>the range</li></ol>																																																		

	<p>Hint: This is computed as maximum-minimum.</p> <p>(iv) the average value</p> <p>Hint: To compute this, add all the numbers together into sum and count them all in Count. The average is Sum/Count</p>												
12	Write a C program to find the length of the string without using library function.												
13	Write a C program to count vowels and consonants in a string using pointer.												
14	<p>Write a C program to generate any one patterns as given below : (Accept the number of rows from user)</p> <table> <tr> <td>i) ( to print * if it is even number)</td><td>ii)</td></tr> <tr> <td>1</td><td>55555</td></tr> <tr> <td>**</td><td>4444</td></tr> <tr> <td>333</td><td>333</td></tr> <tr> <td>****</td><td>22</td></tr> <tr> <td>55555</td><td>1</td></tr> </table>	i) ( to print * if it is even number)	ii)	1	55555	**	4444	333	333	****	22	55555	1
i) ( to print * if it is even number)	ii)												
1	55555												
**	4444												
333	333												
****	22												
55555	1												
15	<p>Develop and demonstrate a C program for Matrix multiplication:</p> <p>a) Read the sizes of two matrices and check the compatibility for multiplication.</p> <p>b) Perform matrix multiplication and print the result along with the input matrix.</p>												
16	<p>Using functions develop a C program to perform the following tasks by parameter passing concept:</p> <p>a) To read a string from the user</p> <p>b) Print appropriate message for palindrome or not palindrome</p>												
17	<p>Write a C program to perform the following operations using recursive functions:</p> <p>a) GCD, LCM (Using GCD method)</p> <p>b) Factorial of a Number</p>												
18	<b>Write a C program to generate Fibonacci series using recursion.</b>												
19	Write a C program to read matrix of m X n order from the user and display whether it is lower triangular matrix or upper triangular matrix.												
20	Write a C program to check whether the entered number is positive, negative or zero.												
21	<p>Write C user defined functions</p> <p>a) To input N integer numbers into a single dimension array.</p> <p>b) To sort the integer numbers in descending order using selection sort technique.</p> <p>c) To print the single dimension array elements.</p> <p>Using these functions, write a C program to input N integer numbers into a single dimension array, sort them in descending order, and print both the given array &amp; the sorted array with suitable headings. <b>(Not simple selection sort)</b></p>												
22	Write a C program to insert a character at specified position and move the remaining character to right.												
23	Write a program to enter a sentence and print total number of vowels and each vowel count separately.												
24	Write a C program to read a string and sort it alphabetically.												
25	Write a C program to read and find the frequency of characters in a given string.												
26	Write a C program to search an element using Binary search.												
27	Demonstrate a C program that reads N integer numbers and arrange them in ascending or												

	descending order using bubble sort technique.
28	Write a C program to read a matrix A[m*n] and to find the following using functions: a) Sum of elements of each row. b) Sum of elements of each column. c) Sum of all the elements of matrix. d) Sum of principle diagonal elements
29	Write a C program to check whether a number is a power of 2 or not.
30	Write a C program to check whether the number entered is prime or not.