DATA STRUCTURES USING C
LABORATORYMANUAL

Name of the program

1	**Write C programs that use both recursive and non-recursive functions** **a. To find the factorial of a given integer.** **b. To find the GCD (greatest common divisor) of two given integers.** **c. To solve Towers of Hanoi problem.**
2	**a) Write a C program to find both the largest and smallest number in a list of integers.** **b) Write a C program that uses functions to perform the following:** **i) Addition of Two Matrices ii) Multiplication of Two Matrices**
3	**a) Write a C program that uses functions to perform the following operations:** **i) To insert a sub-string in to a given main string from a given position.** **ii) To delete n Characters from a given position in a given string.**
4	**a) Write a C program that displays the position or index in the string S where the string T begins, or – 1 if S doesn‘t contain T.** **b) Write a C program to count the lines, words and characters in a given text.**
5	**Write a C program that uses functions to perform the following operations:** **i) Reading a complex number** **ii) Writing a complex number** **iii) Addition of two complex numbers**
6	**Write C programs that implement stack (its operations) using** **i) Arrays ii) Pointers**
7	**Write a C program that uses functions to perform the following operations on singly linked list.** **i) Creation ii) Insertion iii) Deletion iv) Traversal**
8	**Write C programs that use both recursive and non-recursive functions to perform the following searching operations for a Key value in a given list of integers:** **i) Linear search**
9	**Write a C program that uses functions to perform the following:** **Creating a Binary Tree of integers** **Traversing the above binary tree in preorder, inorder and postorder.**
10	**Write a C program that implements the following sorting methods to sort a given list of integers in ascending order** **i) Bubble sort**

Dr K V Subba Reddy School of Business Management

1)Write C programs that use both recursive and non-recursive functions
i. To Find Factorial of given number

a) USING NON RECURSIVE FUNCTION

```
#include <stdio.h>
#include<conio.h>
int factorial(int);
void main()
{
         intn,recFact;
         clrscr();
          printf(“Enter any number: “);
          scanf(“%d”, &n);
         recFact= factorial(n);
         printf(“The factorial of a given number usingnon recursion%d = %d”,n, recFact) ;
        getch();
}
int factorial(intnum)
{
      inti=1,f=1;
            while(i<=num)
           {
                  f=f*i;
                  i++;
           }
return f;
}
```

b) USING RECURSIVE FUNCTION

```
#include <stdio.h>
#include<conio.h>
int factorial(int);
void main()
{
        int n,nonRecFact;
        clrscr();
       printf(“Enter any number: “);
       scanf(“%d”, &n);
       nonRecFact= factorial(n);
       printf(“The factorial of a given number using recursion%d = %d”,n, nonRecFact) ;
     getch();
}
int factorial(int n)
{
        if (n==1)
             return 1;
        else
         return n*factorial(n-1);
```

Dr. K V Subba Reddy School of Business Management

```
}
```

Output:

```
Enter any number
5
The factorial of a given number using recursion is 120
The factorial of a given number using nonrecursion is 120.
```

ii) To find the GCD (greatest common divisor) of two given integers

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

void main()
{
        inta,b,g,l;
        printf("\n Enter First Number: ");
        scanf("%d",&a);
        printf("\n Enter Second Number: ");
        scanf("%d",&b);
        g=gcd(a,b);    // function to get GCD
        l=(a*b)/g;
        printf("\n GCD of %d and %d : %d",a,b,g);
}
```

a) USING RECURSIVE FUNCTION

```
intgcd(int m,int n)
{
        if(n==0)
        return m;
        else
        gcd(n,m%n);
}
```

b)USING NON RECURSIVE FUNCTION

```
intgcd(int a,int b)
{
        int r;
        while(b!=0)
        {
        r=a%b;
        a=b;
        b=r;
        }
        return a;
}
```

Output:

```
Enter First Number: 10
Enter SecondNumber : 20
GCD of 10 and 20 : 10
```

Dr K V Subba Reddy School of Business Management

iii) To solve Towers of Hanoi problem

```
#include <stdio.h>
#include<conio.h>
void main()
{
        int no;
        printf("Enter the no. of disk to be transferred:");
        scanf("%d",&no);
        if(no<1)
        printf("\n There's nothing to move");
        else
        printf("\n recursive");
        Hanoirecursion(no,'A','B','C');
}
```

USING RECURSIVE FUNCTION

```
void Hanoirecursion(intnum,char ndl1,char ndl2,char ndl3)
{
        if(num==1)
        {
        printf("Move disk 1 from rod %c to rod%c",ndl1,ndl2);
        return;
        }
        Hanoirecursion(num-1,ndl1,ndl3,ndl2);
        printf("Move disk %d from rod %c to rod %c",ndl1,ndl2);
        Hanoirecursion(num-1,ndl3,ndl2,ndl1);
}
```

Output:

```
        Move disk 1 from rod A to rod B
         Move disk 2 from rod A to rod C
         Move disk 1 from rod B to rod C
         Move disk 3 from rod A to rod B
         Move disk 1 from rod C to rod A
         Move disk 2 from rod C to rod B
         Move disk 1 from rod A to rod B
         Move disk 4 from rod A to rod C
         Move disk 1 from rod B to rod C
         Move disk 2 from rod B to rod A
         Move disk 1 from rod C to rod A
         Move disk 3 from rod B to rod C
         Move disk 1 from rod A to rod B
         Move disk 2 from rod A to rod C
         Move disk 1 from rod B to rod C
```

Dr K V Subba Reddy School of Business Management

2)

i) Write C program to find both the largest and smallest number in a list of integers

```
#include <stdio.h>
#include<conio.h>
intmain()
{
        inti, n, lar,sm, elem;
        printf ("Enter total number of elements n");
        scanf ("%d", &elem);
        printf ("Enter first number n");
        scanf ("%d", &n);
           lar = n;
        sm=n;
           for (i=1; i<= elem -1 ;i++)
           {
        printf ("n Enter another number n");
        scanf ("%d",&n);
              if (n>lar)
              lar=n;
              if (n<sm)
        sm=n;
           }
        printf ("n The largest number is %d", lar);
        printf ("n The smallest number is %d", sm);
           return 0;
}
```

Output:

```
        Enter total number of elements
        5
        Enter first number
        3
        Enter another number
        890
        Enter another number
        411
        Enter another number
        42
        Enter another number
        89
        Enter another number
        328

        The largest number is 890
        The smallest number is 3
```

Dr K V Subba Reddy School of Business Management

ii) Write a C program that uses functions to perform the following:

a) Addition of Two Matrices

```
#include <stdio.h>
#include <conio.h>
void main()
{
        inta[3][3], b[3][3], c[3][3], i, j;
        clrscr();
        printf("Enter the elements of 3*3 matrix a \n");
                for(i = 0; i< 3; i++)
                   {
                for(j = 0; j < 3; j++)
                      {
                scanf("%d", &a[i][j]);
                      }
                   }
        printf("Enter the elements of 3*3 matrix b \n");
                for(i = 0; i< 3; i++)
                   {
                for(j = 0; j < 3; j++)
                      {
                scanf("%d", &b[i][j]);
                      }
                   }
                for(i = 0; i< 3; i++)
                   {
                for(j = 0; j < 3; j++)
                      {
                    c[i][j] = a[i][j] + b[i][j];
                      }
                   }
        printf("The resultant 3*3 matrix c is \n");
                for(i = 0; i< 3; i++)
                   {
                for(j = 0; j < 3; j++)
                      {
                printf("%d\t", c[i][j]);
                      }
                printf("\n");
                   }
        getch();
}
```

Output:

```
Enter the elements of 3*3 matrix  a
        1 2 3 4 5 6 7 8 9
        Enter the elements of 3*3 matrix  b
        1 2 3 4 5 6 7 8 9
        The resultant 3*3 matrix  c is
        2      4      6
        8     10     12
        14    16     18
```

Dr K V Subba Reddy School of Business Management

b) Multiplication of TwoMatrices

```
#include<stdio.h>
#include<conio.h>
void main(){
inta[3][3], b[3][3], c[3][3], i, j, k;
        clrscr();
        printf("Enter the elements of 3*3 matrix a \n");
                for(i = 0; i< 3; i++)
                  {
                for(j = 0; j < 3; j++)
                    {
                scanf("%d", &a[i][j]);
                    }
                  }
        printf("Enter the elements of 3*3 matrix b \n");
                for(i = 0; i< 3; i++)
                  {
                for(j = 0; j < 3; j++)
                    {
                scanf("%d", &b[i][j]);
                    }
                  }
                for(i = 0; i< 3; i++)
                  {
                for(j = 0; j < 3; j++)
                    {
                    c[i][j] = 0
                for(k = 0; k < 3; k++)
                    {
                      c[i][j] = c[i][j] + (a[i][k] * b[k][j])
                    }
                    }
                  }
        printf("The resultant 3*3 matrix c is \n");
                for(i = 0; i< 3; i++)
                  {
                for(j = 0; j < 3; j++)
                    {
                printf("%d\t", c[i][j]);
                    }
                printf("\n");
                  }
        getch();
}
```

Output:

```
Enter the elements of 3*3 matrix  a 1 2 3 4 5 6 7 8 9
Enter the elements of 3*3 matrix  b 1 2 3 4 5 6 7 8 9
  The resultant 3*3 matrix  c is
             30    36     42
             55    81     96
             102   126     150
```

Dr K V Subba Reddy School of Business Management

3) Write a C program that uses functions to perform the following operations

i) To insert a sub-string in to a given main string from a given position.

```
#include <stdio.h>
#include <string.h>
int main()
{
char a[10];
char b[10];
char c[10];
int p=0,r=0,i=0;
int t=0;
int x,g,s,n,o;
puts("Enter First String:");
gets(a);
puts("Enter Second String:");
gets(b);
printf("Enter the position where the item has to be inserted: ");
scanf("%d",&p);
r = strlen(a);
n = strlen(b);
i=0;

// Copying the input string into another array
while(i <= r)
{
 c[i]=a[i];
 i++;
}
s = n+r;
o = p+n;

// Adding the sub-string
for(i=p;i<s;i++)
{
 x = c[i];
 if(t<n)
 {
  a[i] = b[t];
  t=t+1;
 }
 a[o]=x;
 o=o+1;
}

printf("%s", a);
return 0;
}
```

Dr K V Subba Reddy School of Business Management

Output:

```
Enter First String: program9
Enter Second String: ming
Enter the position where the item has to be inserted: 7
programming9
```

ii) To delete n Characters from a given position in a given string

```
#include <stdio.h>
#include <conio.h>
// prototype of function
void del_str(char [],int, int);
main(){
   int n,p;
   char str[30];
   printf("
 Enter the String:");
   gets(str);
   fflush(stdin);
   printf("
 Enter the position from where the characters are to be deleted:");
   scanf("%d",&p);
   printf("
 Enter Number of characters to be deleted:");
   scanf("%d",&n);
   del_str(str,p,n);
}
//function call
void del_str(char str[],int p, int n){
   int i,j;
   for(i=0,j=0;str[i]!='\0';i++,j++){
      if(i==(p-1)){
         i=i+n;
      }
      str[j]=str[i];
   }
   str[j]='\0';
   puts(" The string after deletion of characters:");
   puts(str);
}
```

Output

```
Enter the String: Tutorials Point C programming
Enter the position from where the characters are to be deleted:10
Enter Number of characters to be deleted: 6
The string after deletion of characters:
Tutorials C programming
```

Dr K V Subba Reddy School of Business Management

4. Write a C program that displays the position or index in the string S where the string T begins, or – 1 if S doesn‘t contain T.

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
void main()
{
  char s[30], t[20];
  char *found;
  clrscr();
  puts("Enter the first string: ");
  gets(s);
  puts("Enter the string to be searched: ");
  gets(t);
  found = strstr(s, t);
  if(found)
  {
    printf("Second String is found in the First String at %d position.\n", found - s);
  }
  else
  {
    printf("-1");
  }
  getch();
}
```

Output

```
1.Enter the first string:
kali
Enter the string to be searched:
li
second string is found in the first string at 2 position

2.Enter the first string:
nagaraju
Enter the string to be searched:
raju
second string is found in the first string at 4 position

3.Enter the first string:
nagarjuna
Enter the string to be searched:
ma
-1
```

Dr K V Subba Reddy School of Business Management

b) Write a C program to count the lines, words and characters in a given text.

```
#include <stdio.h>

int main() {
long ctr_char, ctr_word, ctr_line; // Variables to count characters, words, and lines
int c; // Variable to hold input characters
int flag; // Flag to track word boundaries

ctr_char = 0; // Initialize the count of characters
flag = ctr_line = ctr_word = 0; // Initialize flag and counts for words and lines

printf("Input a string and get number of characters, words and lines:\n");

  // Loop to read characters until end-of-file (EOF) is encountered
while ((c = getchar()) != EOF) {
    ++ctr_char; // Increment the count of characters

if (c == ' ' || c == '\t') {
flag = 0; // Reset the flag when a space or tab is encountered
    } else if (c == '\n') {
      ++ctr_line; // Increment the count of lines
flag = 0; // Reset the flag on a newline
    } else {
if (flag == 0) {
        ++ctr_word; // Increment the count of words when a new word begins
      }
flag = 1; // Set the flag to indicate a word is in progress
    }
  }

  // Print the counts of characters, words, and lines
printf("\nNumber of Characters = %ld", ctr_char);
printf("\nNumber of words = %d", ctr_word);
printf("\nNumber of lines = %d", ctr_line);
}
```

Output

```
Input a string and get number of characters, words and lines terminated with ~ :
Hello, how are you?
Welcome to the programming world.
Programming is fun.
~
Number of Characters = 71
Number of words = 12
Number of lines = = 3
```

Dr K V Subba Reddy School of Business Management

5.Write a C program that uses functions to perform the following operations:

i) Reading a complex number
ii) Writing a complex number
iii) Addition of two complex numbers

```
#include <stdio.h>
#include <conio.h>
struct complex
{
   float real, imag;
}a, b, c;
   struct complex read(void);
   void write(struct complex);
   struct complex add(struct complex, struct complex);
   struct complex sub(struct complex, struct complex);
   struct complex mul(struct complex, struct complex);
   struct complex div(struct complex, struct complex);
void main ()
{
   clrscr();
   printf("Enter the 1st complex number\n ");
   a = read();
   write(a);
   printf("Enter  the 2nd complex number\n");
   b = read();
   write(b);
   printf("Addition\n ");
   c = add(a, b);
   write(c);
   printf("Substraction\n ");
   c = sub(a, b);
   write(c);
   printf("Multiplication\n");
   c = mul(a, b);
   write(c);
   printf("Division\n");
   c = div(a, b);
  write(c);
  getch();
}
```

Dr. K V Subba Reddy School of Business Management

```
struct complex read(void)
{
   struct complex t;
   printf("Enter the real part\n");
   scanf("%f", &t.real);
   printf("Enter the imaginary part\n");
   scanf("%f", &t.imag);
   return t;
}
void write(struct complex a)
{
   printf("Complex number  is\n");
   printf(" %.1f + i %.1f", a.real, a.imag);
   printf("\n");
}
struct complex add(struct complex p, struct complex q)
{
   struct complex t;
   t.real = (p.real + q.real);
   t.imag = (p.imag + q.imag);
   return t;
}
```

Output

```
Enter the real part
2
Enter the imaginary part
4
Complex number is
2.0 + i4.0
Enter the real part
4
Enter the imaginary part
2
Complex number is
4.0 + i2.0
Addition
Complex number is
6.0 + i6.0
```

Dr K V Subba Reddy School of Business Management

6. Write C programs that implement stack (its operations) using
i) Arrays

```
#include<stdio.h>
int stack[100],choice,n,top,x,i;
void push(void);
void pop(void);
void display(void);
int main()
{
   top=-1;
   printf("\n Enter the size of STACK[MAX=100]:");
   scanf("%d",&n);
   printf("\n\t STACK OPERATIONS USING ARRAY");
   printf("\n\t--------------------------------");
   printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t 4.EXIT");
   do
   {
      printf("\n Enter the Choice:");
      scanf("%d",&choice);
      switch(choice)
      {
         case 1:
         {
            push();
            break;
         }
         case 2:
         {
            pop();
            break;
         }
         case 3:
         {
            display();
            break;
         }
         case 4:
         {
            printf("\n\t EXIT POINT ");
            break;
         }
         default:
```

Dr K V Subba Reddy School of Business Management

```
            {
                printf ("\n\t Please Enter a Valid Choice(1/2/3/4)");
            }

        }
    }
    while(choice!=4);
    return 0;
}
void push()
{
    if(top>=n-1)
    {
        printf("\n\tSTACK is over flow");

    }
    else
    {
        printf(" Enter a value to be pushed:");
        scanf("%d",&x);
        top++;
        stack[top]=x;
    }
}
void pop()
{
    if(top<=-1)
    {
        printf("\n\t Stack is under flow");
    }
    else
    {
        printf("\n\t The popped elements is %d",stack[top]);
        top--;
    }
}
void display()
{
    if(top>=0)
    {
        printf("\n The elements in STACK \n");
        for(i=top; i>=0; i--)
            printf("\n%d",stack[i]);
        printf("\n Press Next Choice");
```

Dr K V Subba Reddy School of Business Management

```
    }
    else
    {
       printf("\n The STACK is empty");
    }

}
```

Output

```
Enter the size of STACK[MAX=100]:10

        STACK OPERATIONS USING ARRAY
       --------------------------------
        1.PUSH
        2.POP
        3.DISPLAY
        4.EXIT
 Enter the Choice:1
 Enter a value to be pushed:12

 Enter the Choice:1
 Enter a value to be pushed:24

 Enter the Choice:1
 Enter a value to be pushed:98

 Enter the Choice:3
 The elements in STACK
98
24
12
 Press Next Choice
 Enter the Choice:2
  The popped elements is 98
 Enter the Choice:3
 The elements in STACK
24
12
 Press Next Choice
 Enter the Choice:4

        EXIT POINT
```

Dr. K V Subba Reddy School of Business Management

7. Write C programs that use both recursive and non-recursive functions to perform the following searching operations for a Key value in a given list of integers:

i) Linear search

```
#include <stdio.h>
#include <conio.h>
#define MAX_LEN 10

void l_search_recursive(int l[],int num,int ele);
void l_search_nonrecursive(int l[],int num,int ele);
void read_list(int l[],int num);
void print_list(int l[],int num);

int main()
{
   int l[MAX_LEN], num, ele;
   int ch;


printf("========================================================
");
   printf("\n\t\t\tMENU");

printf("\n=======================================================
");
   printf("\n[1] Linary Search using Recursion method");
   printf("\n[2] Linary Search using Non-Recursion method");
   printf("\n\nEnter your Choice:");
   scanf("%d",&ch);

   if(ch<=2 & ch>0)
   {
      printf("Enter the number of elements :");
      scanf("%d",&num);
      read_list(l,num);
      printf("\nElements present in the list are:\n\n");
      print_list(l,num);
      printf("\n\nElement you want to search:\n\n");
      scanf("%d",&ele);

      switch(ch)
```

Dr. K V Subba Reddy School of Business Management

```
        {
        case 1:
            printf("\n**Recursion method**\n");
            l_search_recursive(l,num,ele);
            getch();
            break;

        case 2:
            printf("\n**Non-Recursion method**\n");
            l_search_nonrecursive(l,num,ele);
            getch();
            break;
        }
    }
    getch();
}

//end main

//Non-Recursive method

void l_search_nonrecursive(int l[],int num,int ele)
{
    int j, f=0;
    for(j=0; j<num; j++)
        if( l[j] == ele)
        {
            printf("\nThe element %d is present at position %d in list\n",ele,j);
            f=1;
            break;
        }
    if(f==0)
        printf("\nThe element is %d is not present in the list\n",ele);
}

//Recursive method
void l_search_recursive(int l[],int num,int ele)
{
    int f = 0;

    if( l[num] == ele)
    {
        printf("\nThe element %d is present at position %d in list\n",ele,num);
```

Dr.K V Subba Reddy School of Business Management

```
  f=1;
   }
   else
   {
      if((num==0) && (f==0))
      {
         printf("The element %d is not found.",ele);
      }
      else
      {
         l_search_nonrecursive(l,num-1,ele);
      }
   }
   getch();
}

void read_list(int l[],int num)
{
   int j;
   printf("\nEnter the elements:\n");
   for(j=0; j<num; j++)
      scanf("%d",&l[j]);
}

void print_list(int l[],int num)
{
   int j;
   for(j=0; j<num; j++)
      printf("%d\t",l[j]);
}
```

Output

```
Enter the size of an array 6
Enter the array elements 50 10 5 200 20 1
Enter the key element 1
The key Element is found at location 6
```

Dr. K V Subba Reddy School of Business Management

9. Write a C program that uses functions to perform the following:
Creating a Binary Tree of integers
Traversing the above binary tree in preorder, inorder and postorder.

```
// Tree traversal in C

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

struct node {
  int item;
  struct node* left;
  struct node* right;
};

// Inorder traversal
void inorderTraversal(struct node* root) {
  if (root == NULL) return;
  inorderTraversal(root->left);
  printf("%d ->", root->item);
  inorderTraversal(root->right);
}

// preorderTraversal traversal
void preorderTraversal(struct node* root) {
  if (root == NULL) return;
  printf("%d ->", root->item);
  preorderTraversal(root->left);
  preorderTraversal(root->right);
}

// postorderTraversal traversal
void postorderTraversal(struct node* root) {
  if (root == NULL) return;
  postorderTraversal(root->left);
  postorderTraversal(root->right);
  printf("%d ->", root->item);
}
// Create a new Node
struct node* createNode(value) {
  struct node* newNode = malloc(sizeof(struct node));
```

Dr K V Subba Reddy School of Business Management

```
  newNode->item = value;
  newNode->left = NULL;
  newNode->right = NULL;

  return newNode;
}

// Insert on the left of the node
struct node* insertLeft(struct node* root, int value) {
  root->left = createNode(value);
  return root->left;
}

// Insert on the right of the node
struct node* insertRight(struct node* root, int value) {
  root->right = createNode(value);
  return root->right;
}

int main() {
  struct node* root = createNode(1);
  insertLeft(root, 12);
  insertRight(root, 9);

  insertLeft(root->left, 5);
  insertRight(root->left, 6);

  printf("Inorder traversal \n");
  inorderTraversal(root);

  printf("\nPreorder traversal \n");
  preorderTraversal(root);

  printf("\nPostorder traversal \n");
  postorderTraversal(root);
}
```

Dr K V Subba Reddy School of Business Management

10. Write a C program that implements the following sorting methods to sort a given list of integers in ascending order

i) Bubble sort

```
#include <stdio.h>

void bubble_sort(int arr[], int n) {
  int i, j;
  for (i = 0; i < n - 1; i++) {
    for (j = 0; j < n - i - 1; j++) {
      if (arr[j] > arr[j + 1]) {
        int temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
      }
    }
  }
}
int main() {
  int arr[] = {64, 34, 25, 12, 22, 11, 90};
  int n = sizeof(arr) / sizeof(arr[0]);
  bubble_sort(arr, n);
  printf("Sorted array: ");
  for (int i = 0; i < n; i++) {
    printf("%d ", arr[i]);
  }
  return 0;
}
```

Output

```
Sorted array: 11 12 22 25 34 64 90
```

Dr K V Subba Reddy School of Business Management