

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12 (B) of UGC ACT 1956

(Affiliated to JNTUH, Hyderabad, Approved by AICTE-Accredited by NBA & NACC-**'A' Grade** – ISO 9001:2008 Certified) Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad -500100, Telangana State, India

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

NAME OF THE STUDENT:
ROLL NO :
BRANCH:SECTION:
YEAR:SEMESTER:

FACULTY INCHARGE SIGNATURE

I Year B. Tech -I Sem

L T/P/D C

-/3/- 1.5

(R18A0581) PROGRAMMING FOR PROBLEM SOLVING LAB

Program Objectives:

- Understand the basic concept of C Programming, and its different modules that include conditional and looping expressions, Arrays, Strings, Functions, Pointers, and Structures.
- Acquire knowledge about the basic concept of writing a program.
- Role of constants, variables, identifiers, operators, type conversion and other building blocks of C Language.
- Use of conditional expressions and looping statements to solve problems associated with conditions and repetitions.
- Role of Functions involving the idea of modularity.
- Programming using gcc compiler in Linux.

Week 1:

a) Write a C program to find sum and average of three numbers.

b) Write a C program to find the sum of individual digits of a given positive integer.

Week 2:

a) Write a C program to generate the first n terms of the Fibonacci sequence.

b) Write a C program to generate prime numbers from 1 to n.

c) Write a C program to check whether given number is Armstrong Number or not. **Week 3:**

a) Write a C program to check whether given number is perfect number or not.

b) Write a C program to check whether given number is strong number or not.

Week 4:

a) Write a C program to find the roots of a quadratic equation.

b) Write a C program to perform arithmetic operations using switch statement.

Week 5:

a) Write a C program to find factorial of a given integer using non-recursive function.

b) Write a C program to find factorial of a given integer using recursive function.

Week 6:

a) Write C program to find GCD of two integers by using recursive function.

b) Write C program to find GCD of two integers using non-recursive function.

Week 7:

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

b) Write a C program to Sort the Array in an Ascending Order

c) Write a C program to find whether given matrix is symmetric or not.

Week 8:

Revision of programs

Week 9:

- a) Write a C program to perform addition of two matrices.
- b) Write a C program that uses functions to perform multiplication of two Matrices.

Week 10:

- a) Write a C program to use function to insert a sub-string in to given main string from a given position.
- b) Write a C program that uses functions to delete n Characters from a given position in a given string.

Week 11:

- a) Write a C program using user defined functions to determine whether the given string is palindrome or not.
- b) Write a C program that displays the position or index in the main string S where the sub string T begins, or 1 if S doesn't contain T.

Week 12:

- a) Write C program to count the number of lines, words and characters in a given text.
- b) Write a C program to find the sum of integer array elements using pointers.

Week 13:

a) Write a C program to Calculate Total and Percentage marks of a student using structure.

Week 14:

Revision of Programs

TEXT BOOKS

- 1. C Programming and Data Structures, P.Padmanabham, Third Edition, BS Publications
- 2. Computer programming in C.V.RAjaraman, PHI Publishers.
- 3. C Programming, E.Balagurusamy, 3rd edition, TMH Publishers.
- 4. C Programming, M.V.S.S.N Venkateswarlu and E.V.Prasad, S.Chand Publishers
- 5. Mastering C,K.R.Venugopal and S.R.Prasad, TMH Publishers.

Program Outcomes:

- Acquire knowledge about the basic concept of writing a program.
- Understand the Role of constants, variables, identifiers, operators, type conversion and other building blocks of C Language.
- Learn how to use of conditional expressions and looping statements to solve problems associated with conditions and repetitions.
- Understand the Role of Functions involving the idea of modularity.
- Understand the Concept of Array and pointers dealing with memory management.
- Learn Structures and unions through which derived data types can be formed.

CONTENTS

Week	Name of the program	P no				
1	a) Write a C program to find sum and average of three numbers.b) Write a C program to find the sum of individual digits of a given positive integer.	1				
2	a) Write a C program to generate the first n terms of the Fibonacci sequenceb) Write a C program to generate prime numbers between 1 to n.c) Write a C program to check if the given number is Armstrong or not	5				
3	a) Write a C program to check whether the given number is perfect or notb) Write a C program to check whether the given number is strong or not	12				
4	a) Write a C program to find the roots of a quadratic equation.b) Write a C program perform arithmetic operations using switch statement.	16				
5	a) Write a C program to find factorial of a given integer using non-recursive functionb) Write a C program to find factorial of a given integer using recursive function	23				
6	a) Write C program to find GCD of two integers by using recursive function.b) Write C program to find GCD of two integers by using non-recursive function.	27				
7	a) Write a C program to find the largest and smallest number in a list of integers.b) Write a C program to Sort the Array in an Ascending Order.c) Write a C program to find whether the given matrix is symmetric or not.	31				
8	Revision of Programs					
9	a) Write a C program to perform addition of two matrices.b) Write a C program using function to perform multiplication of two matrices.	41				
10	 a) Write a C program to use function to insert a sub-string in to given main string from a given position. b) Write a C program to swap the values of two variables using (i) Call by value (ii) Call by reference 	48				
11	 a) Write a C program using user-defined functions to determine whether the given string is palindrome or not. b) Write a C program that displays the position or index in the main string S where the sub string T begins, or - 1 if S doesn't contain T. 	55				
12	a) Write C program to count the number of lines, words and characters in a given text.b) Write a C program to find the sum of integer array elements using pointers.	59				
13	a) Write a C Program to Calculate Total and Percentage marks of a student using structure.	63				
14	Revision of Programs					

INSTRUCTIONS TO STUDENTS

- Before entering the lab the student should carry the following things (MANDATORY)
 - 1. Identity card issued by the college.
 - 2. Class notes
 - 3. Lab observation book
 - 4. Lab Manual
 - 5. Lab Record
- Student must sign in and sign out in the register provided when attending the lab session without fail.
- Come to the laboratory in time. Students, who are late more than 15 min., will not be allowed to attend the lab.
- Students need to maintain 100% attendance in lab if not a strict action will be taken.
- All students must follow a Dress Code while in the laboratory
- Foods, drinks are NOT allowed.
- All bags must be left at the indicated place.
- Refer to the lab staff if you need any help in using the lab.
- Respect the laboratory and its other users.
- Workspace must be kept clean and tidy after experiment is completed.
- Read the Manual carefully before coming to the laboratory and be sure about what you are supposed to do.
- Do the experiments as per the instructions given in the manual.
- Copy all the programs to observation which are taught in class before attending the lab session.
- Students are not supposed to use floppy disks, pen drives without permission of lab- in charge.
- Lab records need to be submitted on or before the date of submission.

Week 1:

1. a) Write a C program to find the sum and average of three numbers.

AIM: find the sum and average of three numbers Description:

sum=number1+number2+number3

Average=sum/3

ALGORITHM:

Step 1: Start

Step 2: Declare variables num1, num2, num3 and sum, average. Step

3: Read values num1,num2,num3

Step 4: Add num1,num2,num3 and assign the result to sum.

 $sum \leftarrow num1 + num2 + num3$

average $\leftarrow \text{sum/3}$

Step 5: Display sum and average

Step 6: Stop

FLOWCHART:



```
#include<stdio.h>
int main()
{
    int a,b,c;
    int sum,average;
    printf("Enter any three integers: ");
    scanf("%d%d %d",&a,&b,&c);
    sum = a+b+c;
    average=sum/3
    printf("Sum and average of three integers: %d %d",sum,average);
    return 0;
}
```

INPUT: Enter any three integers:2 4 5

OUTPUT:

Sum and average of three integers: 11 3

Record at least 2results

1. b) Write a C program to find the sum of individual digits of positive integer.

AIM: To find the sum of individual digits of positive integer.

Description:

Summation of digits of a number Ex: 1234

Summation =1+2+3+4=10

ALGORITHM:

Step 1: Start Step 2: Read n Step 3: Initialize sum $\leftarrow 0$ Step 4: while(n!=0) Begin Step 5: r \leftarrow n%10 Step 6: sum \leftarrow sum+r Step 7: n \leftarrow n/10 End Step 8: Print "sum" Step 9: Stop

FLOWCHART:



```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,r,sum=0;
clrscr();
      printf("ENTER A POSITIVE INTEGER \n");
      scanf("%d",&n);
      while(n!=0)
      {
             r=n%10;
             sum=sum+r;
             n=n/10;
      }
printf("THE SUMOF INDIVIDUAL DIGITS OF A POSITIVE INTEGER IS..%d",sum);
getch();
}
```

```
INPUT:
```

ENTER A POSITIVE INTEGER 5 3 2 1

```
OUTPUT:
```

THE SUM OF INDIVIDUAL DIGITS OF A POSITIVE INTEGER IS..11

Record at least 2 results

EXERCISE:

- 1. Program to find the sum of all numbers between 1 to 10
- 2. Program to find odd or even number using conditional operator
- 3. Program to find sum of even numbers between 50 to 100.
- 4. Program to find whether the given year is leap year or not.

2).a) Write a C program to generate the first n terms of the Fibonacci Sequence

AIM: To generate the first n terms of the Fibonacci sequence..

Description: Initial Fibonacci numbers are 0 and 1. Next number can be generated by adding two numbers. So 0+1=1. Therefore next number can be generated by adding two previous . so Fibonacci series is 0 1 1 2 3 5

ALGORITHM:

Step 1 : Start Step 2 : Read n Step 3 : Initialize $f0 \leftarrow 0, f1 \leftarrow 1, f \leftarrow 0$ Step 4 :i=0 Step 5 : while(i<=n) do as follows printf("%d\t",f0); f=f0+f1;f0=f1; f1=f; i=i+1;If not goto step 7 Step 6 : Stop **FLOWCHART:** Start f0=0, f1=1 $\mathbf{i} = \mathbf{0}$



```
#include<stdio.h>
#include<conio.h>
void main()
{
int f0,f1,f,n,i;
      clrscr();
      printf("ENTER THE VALUE FOR n \n");
      scanf("%d",&n);
      f0=0;
      f1=1;
      printf("FIBONACCI SEQUENCE FOR THE FIRST %d TERMS:\n",n);
      i=0;
      while(i<n)
       {
             printf("%d\t",f0);
             f=f0+f1;
             f0=f1;
             f1=f;
             i=i+1;
       }
}
INPUT:
```

ENTER THE VALUE FOR n 10

OUTPUT:FIBONACCI SEQUENCE FOR THE FIRST 10 TERMS:01123581321

Record at least 2results

```
Signature of faculty with date
```

34

2) b)Write a C program to generate all prime numbers between 1 and n. Where n is the value supplied by the user.

AIM: To print a prime numbers up to 1 to n

Description:

Prime number is a number which is exactly divisible by one and itself only Ex: 2, 3,5,7,.....;

ALGORITHM:

Step 1: start

Step 2: read n

Step 3: initialize i=1,c=0

Step 4:if i<=n goto step 5

If not goto step 10

Step 5: initialize j=1

Step 6: if $j \le i$ do the following. If no goto step 7

i)if i%j==0 increment c

ii) increment j

```
iii) goto Step 6
```

Step 7: if c== 2 print i

Step 8: increment i

Step 9: goto step 4

Step 10: stop

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL



```
#include<stdio.h>
#include<conio.h>
void main()
{
  int n,i,fact,j;
  clrscr();
  printf("enter the number:");
  scanf("%d",&n);
  for(i=1;i<=n;i++)
       fact=0:
  {
       //THIS LOOP WILL CHECK A NO TO BE PRIME NO. OR NOT.
       for(j=1;j<=i;j++)
       {
              if(i\% j==0)
              fact++;
       }
       if(fact==2)
             printf("\n %d",i);
  }
getch();
}
INPUT:
Enter the number : 5
OUTPUT:
  2 3 5
Record at least 2 results
```

2) c) Write a C program to Check whether given number is Armstrong Number or Not.

AIM: To Check whether given number is Armstrong Number or Not

Description:

An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since $3^{**}3 + 7^{**}3 + 1^{**}3 = 371$

ALGORITHM:

Armstrong number Step 1: start Step 2:read n Step 3:assign sum $\leftarrow 0, I \leftarrow m \leftarrow n, count = 0$ Step 4:if m>0 repeat Step 4.1:m $\leftarrow m/10$ Step 4.2:count++ Step 4.3:until the condition fail Step 5: if I>0 repeat step 4 until condition fail Step 5.1:rem $\leftarrow I\% 10$ Step 5.2:sum $\leftarrow sum + pow(rem, count)$ Step 5.3:I $\leftarrow I/10$ Step 6:if n=sum print Armstrong otherwise print not Armstrong

Step 7:stop

FLOWCHART:



```
#include <stdio.h>
int main()
{
 int n, n1, rem, num=0;
       printf("Enter a positive integer: ");
        scanf("%d", &n);
       n1=n;
       while(n1!=0)
       {
               rem=n1%10;
               num+=rem*rem*rem;
               n1/=10;
       }
 if(num==n)
       printf("%d is an Armstrong number.",n);
 else
       printf("%d is not an Armstrong number.",n);
}
INPUT:
Enter a positive integer: 371
OUTPUT:
371 is an Armstrong number.
```

Record at least 2 results

EXERCISE:

- 1) Program to count the number of digits in a given integer using loop
- 2) Program to check whether an alphabet is vowel or Consonant
- 3) Program to check whether the given number is positive, negative or zero
- 4) Program to generate multiplication table for a given element upto 10
- 5) Program to find the reverse of a given integer

Week: 3

3) a). Write a C program to check whether given number is perfect number or Not

AIM: To Check whether given number is perfect number or not **Description:**

A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself.

Ex: 6X1=6 3x2=6 2x3=6 1,2,3 are factors of 6 So 1+2+3=6

ALGORITHM:

```
Step 1: read n

Step 2: assign i=1,sum=0

Step 3: while(i<n) goto step 4

Step 4: if(n%i==0)

sum=sum+i

i++
```

step 5: if(sum==n) print given number is perfect number otherwise not a perfect number.

PROGRAM:

```
#include<stdio.h>
int main()
ł
 int n,i=1,sum=0;
 printf("Enter a number: ");
 scanf("%d",&n);
 while(i<n)
ł
       if(n\%i==0)
       sum=sum+i;
       i++;
 }
 if(sum==n)
    printf("%d is a perfect number",i);
 else
    printf("%d is not a perfect number",i);
```

return 0;

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

}

INPUT: Enter a number:6

OUTPUT:

6 is a perfect number

Record at least 2 results

Signature of faculty with date

•

3) b) Write a C program to check whether a number is strong number or not.

AIM: To check whether given number is strong number or not **Description:** Strong numbers are the numbers whose sum of factorial of digits is equal to the original number. Example: 145 is a strong number.Ex:1!+4!+5!=1+24+120=145

ALGORITHM:

```
Step 1:read num,i,f,r,sum=0,temp

Step 2: assign num to temp

Step 3: while(num) goto step 4

Step 4: i=1,f=1

r=num%10

while(i<=r) goto step 5

Step 5: f=f*i

i=i+1

Step 6: sum=sum+f;

Step 7: num=num/10;

Step 8: if sum and temp are equal got step 9

Step 9: print strong number otherwise not a strong number

PROGRAM:
```

```
#include<stdio.h>
int main()
{
      int num,i,f,r,sum=0,temp;
      printf("Enter a number: ");
      scanf("%d",&num);
      temp=num;
       while(num) {
              i=1,f=1;
              r=num%10;
              while(i<=r) {
                     f=f*i;
                     i++;
              }
              sum=sum+f;
              num=num/10;
       }
```

MRCET

```
if(sum==temp)
    printf("%d is a strong number",temp); else
    printf("%d is not a strong number",temp);
return 0;
```

INPUT: Enter a number: 145

}

OUTPUT: 145 is a strong number

Record at least 2 results

EXERCISE:

- **1.** Program to generate the sum of series 1+2/2!+3/3!+...+n/n!
- 2 Program to print the given triangular form 1

1			
1	2		
1	2	3	
1	2	3	4
		•	

- 3. Program to find the square root of a given number
- 4. program to display the following in triangular form

* * * * *

Week: 4

4) a) Write a C program to find the roots of a quadratic equation.

AIM: To find the roots of a quadratic equation. **Description:** roots of quadratic equation are $^{-b\pm\sqrt{b}\frac{2-4ac}{2a}}$

ALGORITHM:

Step 1: Start Step 2: Read a,b,c Step 3: calculate disc = b*b-4*a*cStep 4: if(disc>0) Begin Step 5: root1=(-b+sqrt(disc))/(2*a)Step 6: root2=(-b-sqrt(disc))/(2*a) Step 7: Print "Root1", "Root2" End Step 8: else if(disc=0) Begin Step 9: root1=-b/(2*a) Step 10: root2=root1; Step 11: Print "Root1", "Root2" End Step 12: else Step 13: Print Roots are imaginary Step 14: Stop **FLOW CHART:**



DEPARTMENT OF CSE/IT

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
int a,b,c;
float disc,root1,root2;
float img,real;
printf("ENTER VALUES FOR a,b,c:\n");
scanf("%d%d%d",&a,&b,&c);
disc=(float)b*b-4*a*c;
if(disc>0)
{
      printf("THE ROOTS ARE REAL & UNEQUAL:\n");
      root1=(-b+sqrt(disc))/(2*a);
      root2=(-b-sqrt(disc))/(2*a);
      printf("Root1=%f\n",root1);
      printf("Root2=%f\n",root2);
}
else if(disc==0)
{
             printf("THE ROOTS ARE REAL AND EQUAL:\n");
             root1=-b/(2*a);
             root2=root1;
             printf("Root1=%f\n",root1);
             printf("Root2=%f\n",root2);
}
    else
    { printf("THE ROOTS ARE IMAGINARY:\n");
      disc=-disc;
      img=(float)disc/2*a;
      real=(float)-b/2*a;
      if (img>0)
             printf("Root1=%f + i\% f n",real,img);
       {
             printf("Root2=%f - i%f\n",real,img);
       }
      else
             img=-img;
       {
      printf("Root1=%f + i\% f \ real, img);
      printf("Root2=%f - i%f\n",real,img);
       ł
  }
```

return 0;

MRCET

}

INPUT: ENTER VALUES FOR a,b,c 1 4 4 OUTPUT: THE ROOTS ARE EQUAL AND THEY ARE.. Root1=-2 Root2=-2

Record at least 2 results

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

4) b). Write a C program which takes two integer operands and one operator from the user, performs the operation and then prints the result.(Consider the operators +,-,*,/,% and use Switch Statement.)

AIM: To perform arithmetic operations using switch statement.

Description: It is also called as multi-way decision statement. The swith statement successively tests the value of an expression againts a given list for matching. If match is found ,then that block is executed. **ALGORITHM:**

Step 1: Read a,b Step 2: Print "Menu Options" Step 3: do Begin Step 4: Read ch Step 5: switch(ch) Begin Step 6: case 1: Begin Calculate c = a+bPrint "c" break: End case 2: Begin Calculate c = a-bPrint "c" break: End case 3: Begin Calculate c = a*bPrint "c" break; End case 4: Begin Calculate c = a/bPrint "c" break: End case 5: Begin Calculate c = a%bPrint "c" break;

MRCET

End

default:

Print "Invalid choice"

End

FLOWCHART



PROGRAM:

#include<stdio.h> #include<conio.h> void main() { int a,b,c,ch; clrscr(); printf("ENTER TWO VALUES FOR a & b\n"); scanf("%d %d",&a,&b); while(1) { printf("MENU OPTIONS \n"); printf("*******/n"); printf("1.Addition\n"); printf("2.Subtraction\n"); printf("3.Multiplication\n"); printf("4.Division\n"); printf("5.Modulus\n"); printf("6.Exit\n"); printf("\n"); printf("ENTER UR CHOICE\n"); scanf("%d",&ch); switch(ch)

MRCET

```
case 1: c=a+b;
    printf("The addition of %d and %d is..%d\n",a,b,c); break;
case 2: c=a-b;
    printf("The subtraction of %d and %d is..%d\n",a,b,c); break;
case 3: c=a*b;
    printf("The multiplication of %d and %d is..%d\n",a,b,c); break;
case 4: c=a/b;
    printf("The division of %d and %d is..%d\n",a,b,c); break;
case 5: c=a%b;
    printf("The modulus of %d and %d is..%d\n",a,b,c); break;
case 6:exit(0); default:printf("INVALID CHOICE\n"); }
}
```

INPUT:

{

ENTER TWO VALUES FOR a & b: 20 16

OUTPUT:

MENU OPTIONS 1.Addition 2.Subtraction 3.Multiplication 4.Division 5.Modulus 6.Exit

ENTER UR CHOICE 1 The addition of 20 and 16 is..36 ENTER UR CHOICE 2 The subtraction of 20 and 16 is..4 ENTER UR CHOICE 3 The multiplication of 20 and 16 is..320 ENTER UR CHOICE 4... ENTER UR CHOICE 5...

Record at least 2results

EXERCISE:

- **1.** Program to print name of days in a week using switch case
- 2. Program to print name of days in a week using else-if ladder

Week: 5

5) a) Write a C program to find the factorial of a given integer using non-recursive function.

AIM: To find the factorial of a given number using non-recursive function.3 **Description:** $n!=n^*(n-1)^*(n-2)....*1$

ALGORITHM:

Step 1: Start Step 2: Read n Step 3: Call fact(n) goto step 6 Step 4: Store result in "f" Step 5: Print "f" goto step 10 Step 6: Begin //sub program

Initialize $f \leftarrow 1$

Step 7: for i is 1 to n by step 2

- Step 8: Calculate $f = f^*i$
- Step 9: return "f"
- End
- Step 10: Stop



```
#include<stdio.h>
#include<conio.h>
int fact(int);
void main()
{
int n,i,f;
       clrscr();
       printf("ENTER A VALUE FOR n:\n");
       scanf("%d",&n);
       f=fact(n);
       printf("THE FACTORIAL OF A GIVEN NO IS..%d",f);
       getch();
}
int fact(int n)
{
       int i,f=1;
       for(i=1;i<=n;i++)
       f=f*i;
       return(f);
}
INPUT:
ENTER A VALUE FOR n
5
OUTPUT:
```

Record at least 2results

THE FACTORIAL OF A GIVEN NUMBER IS..120

5) b) Write a C program to find the factorial of a given integer using recursive function.

AIM: To find the factorial of a given number using recursive function.

Description: The function which calls itself is called recursive function

ALGORITHM:

Main program

Step 1: start Step 2: read n Step 3: call sub program as f=fact(n) Step 4: print f value Step 5: stop

Sub program:

Step 1: initialize the f Step 2: if n==0 or n==1 return 1 to main program if not goto step 3 Step 3: return n*fact(n-1) to main program

FLOW CHART:



PROGRAM:

#include<stdio.h>
#include<conio.h>
int fact(int);
void main()

MRCET

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

```
{
int n,res;
clrscr();
      printf("ENETR A NUMBER:\n");
      scanf("%d",&n);
      res=fact(n);
      printf("THE FACTORIAL OF A GIVEN NUMBER IS..%d",res);
      getch();
}
int fact(int n)
{
int r;
      if(n==0)
             return(1);
      else
              {
                    r=n*fact(n-1);
                    return(r);
              }
}
INPUT:
ENTER A VALUE FOR n
5
```

```
OUTPUT:
THE FACTORIAL OF A GIVEN NUMBER IS..120
```

Record at least 2 results

EXERCISE:

- 1. Program to find area and circumference of a circle using functions
- 2. Program to find square of a given number using function

Week: 6

6) a) Write a C program to find the GCD of two given integers by using the recursive function

AIM: To find the Gcd of two given integers by using the recursive function

Description: The greatest common divisor (gcd) of two or more integers, when at least one of them is not zero, is the largest positive integer that divides thenumbers without a remainder. For example, the GCD of 8 and 12 is 4.

ALGORITHM:

<u>Main program:</u>

Step 1: start Step 2: read a,b Step 3: call the sub program GCD(a,b) for print the value Step 4: stop

Sub program: GCD(n,m)

Step 1: if n>m return GCD(n,m) Step 2: if n==0 return m else goto step 3 Step 3: return GCD (n,m%n) Step 4: return to main program

FLOW CHART:



```
#include<stdio.h>
#include<conio.h>
int gcdrecursive(int m,int n)
{
        if(n>m)
              return gcdrecursive(n,m);
        if(n==0)
              return m;
         else
              return gcdrecursive(n,m%n);
                                                 // return to the main program
}
void main()
{
       int a,b,igcd; clrscr();
        printf("enter the two numbers whose gcd is to be found:");
        scanf("%d%d",&a,&b);
        printf("GCD of a,b is %d",gcdrecursive(a,b)); // return to the sub program getch();
}
INPUT:
```

Enter the two numbers whose gcd is to be found: 5 25

OUTPUT:

GCD of a,b is : 5

Record at least 2results

6) b) Write a C program to find the GCD of two given integers using non-recursive function.

AIM: To find the GCD of two given integers by using the non recursive function

Description:

GCD means Greatest Common Divisor. i.e the highest number which divides the given number Ex: GCD(12,24) is 12 Formula: GCD= product of numbers/ LCM of numbers

ALGORITHM:

Step 1: start Step 2: read a,b Step 3: call sub program g=GCD(a,b) Step 4: print the g value Step 5: stop

Sub program:

Step 1: initialize the p=1, q, remainder Step 2: remainder= $p-(p/q^*q)$

Step 3: remainder=0 return q else goto step 4

Step 4: GCD(q,remainder) return to main program

FLOWCHART:



PROGRAM:

#include<stdio.h>
#include<conio.h>
int gcd(int a,int b);

MRCET

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

```
int main()
{
int a,b;
int r,t;
printf("Enter any two integers");
scanf("%d%d",&a,&b);
       r=gcd(a,b);
       printf("GCD=%d",r);
       getch();
}
int gcd(int a,int b)
{
int t,rem;
       while(1)
       {
          if(b>a)
           ł
                t=a;
                a=b;
                b=t;
           }
          if(b==0)
               return a;
          else
           {
              rem=a%b;
               a=rem;
           }
       }
}
```

INPUT:

enter the two numbers whose gcd is to be found:5,25 **OUTPUT:** GCD of a,b is : 5

Record at least 2 results

Exercise:

- 1. Program to generate the Fibonacci series for a given element using
 - i) Recursive function (ii) Non-recursive function

Week: 7

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

AIM: To find the largest and smallest number in a list of integers.

Description: By comparing a list of sorted or unsorted list of integers, the smallest and largest integer is found.

ALGORITHM:

Step 1: start Step 2: read n Step 3: initialize i=0 Step 4: if i<n do as follows. If not goto step 5 Read a[i] Increment i Goto step 4 Step 5: small=a[0], large=a[0] Step 6: initialize i=0 Step 7: if i<n do as follows. If not goto step 8 If a[i]<small Assign small=a[i] If a[i]>large Assign large=a[i] Increment i goto Step 7 Step 8: print small, large Step 9: stop

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

FLOWCHART:



PROGRAM:

#include<stdio.h>
#include<conio.h>
void main()
{
 int a[10],i,n,small,large;
 clrscr();
 printf("Enter The Array Size:");
 scanf("%d",&n);
 printf("ENTER ELEMENTS OF ARRAY");
 for(i=0;i<n;i++) // read the elements of an array
 scanf("%d",&a[i]);</pre>

MRCET

DEPARTMENT OF CSE/IT

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

```
small=a[0];
large=a[0];
```

```
for(i=0;i<n;i++)// read the elements of an array
        if(a[i]<small)// check the condition for minimum value
 {
      small=a[i];
      if(a[i]>large)//check the condition for maximum value
      large=a[i];
 }
      printf("largest value is:%d\n",large);
      printf("smallest value is:%d\n",small);
      getch();
}
INPUT:
Enter The Array Size:10
ENTER THE ELEMENTS OF ARRAY
       10
                     8
                                           2
                                                  3
                                                         4
                                                                1
7
              9
                            6
                                   5
OUTPUT:
    largest value is : 10
   smallest value is : 1
```

Record at least 2 results

7) b) Write a C Program to Sort the Array in an Ascending Order.

AIM: C Program to Sort the Array in an Ascending Order

Description: The given elements in an are arranged in an ascending order by comparing each element with one another.

ALGORITHM:

Step 1: start Step 2: read n Step 3: initialize i=0 Step 4: if i<n do as follows. If not goto step 5 Read a[i] Increment i Goto step 4 Step 5: small=a[0], large=a[0]Step 6: initialize i=0 Step 7: if i<n do as follows. If not goto step 8 j=0 if j<n-1 do as follows If a[j] < a[j+1]t = a[j]a[j] = a[j+1]a[j+1] = tIncrement j Increment i goto Step 7 Step 8: if i<n do as follows. If not goto step 9 print a[i] Increment i Goto step 8 Step 9: stop

FLOWCHART:



```
#include <stdio.h>
void main()
{
  int i, j, a, n, number[30];
  printf("Enter the value of n:");
  scanf("%d", &n);
  printf("Enter the numbers \n");
  for (i = 0; i < n; ++i)
  scanf("%d", &number[i]);
  for (i = 0; i < n; ++i)
  {
     for (j = 0; j < n-1; ++j)
     ł
       if (number[j] > number[j+1])
        ł
          a = number[j];
          number[j] = number[j+1];
          number[i+1] = a;
       }
     }
  }
  printf("The numbers arranged in ascending order are given below \n");
  for (i = 0; i < n; ++i)
     printf("%d\n", number[i]);
}
```

INPUT:

Enter the value of n:4 Enter the numbers 78 3 90 456

OUTPUT:

The numbers arranged in ascending order are given below 3 78 90

456

Record at least 2 results

7) c) Write a C Program to find whether the given matrix is symmetric or not.

AIM: C Program to find whether the given matrix is symmetric or not.

Description: A square matrix is said to be symmetric matrix if the transpose of the matrix is same as the given matrix. Symmetric matrix can be obtained by changing row to column and column to row.

ALGORITHM:

Step 1: Start
Step 2: read row-order as m and col-order as n
for i is 0 to m by step 1
for j is 0 to n by step 1
Step 3: Read a[i][j]
Step 4: go to step 2
Step 5: compare a[i][j] with a[j][i]
Step 6: if(a[i][j]!=a[j][i]) print "Matrix is not symmetric"
go to step 9
Step 7: go to step 4
Step 8: print "Matrix is symmetric"
Step 9: Stop

```
#include<conio.h>
#include<stdio.h>
void main()
{
int a[10][10],i,j,m,n;
clrscr();
printf("Enter row-order and col-order of matrix: ");
scanf("%d%d",&m&n);
for(i=1;i<=m;i++)
{
 for(j=1;j<=n;j++)
   {
     printf("Enter %d values one by one : ",i*j);
     scanf("%d",&a[i][j]);
    }
}
for(i=1;i<=m;i++)
{
   for(j=1;j<=n;j++)
    {
      if(a[i][j]!=a[j][i])
       ł
        printf("\n\nMatrix is not symmetric");
        getch();
        exit(0);
       }
    }
}
printf("\n\nMatrix is symmetric");
getch();
}
```

INPUT:

Enter row-order and col-order of matrix:

- 2
- 2

Enter 4 values one by one

12

34

34 54 **OUTPUT:** Matrix is symmetric

Record at least 2 results

EXERCISE:

- 1. Program to find the sum and average of marks of a student using array
- 2. Program to reverse an array of integers

DEPARTMENT OF CSE/IT

Week:9

9) a) Write a C program to perform addition of two matrices.

AIM: To perform addition of two matrices.

Description: Consider two matrices and their order is R1xC1 and R2XC2. The condition is R1==R2 and C1==C2, then only the addition is possible.

ALGORITHM:

Step 1: Start
Step21: for i is 0 to 2 by step 1
 for j is 0 to 2 by step 1
Step 3: Read a[i][j],b[i][j]
Step 4: go to step 2
Step 5: calculate c[i][j]=a[i][j]+b[i][j]
Step 6: go to step 2
Step 7: Print c[i][j]
Step 8: Stop

FLOW CHART:



```
#include<stdio.h>
#include<conio.h>
void read_matrix(int a[5][5],int row,int col)
{
int i,j;
       for(i=0;i<row;i++)</pre>
               for(j=0;j<col;j++)
               scanf("%d",&a[i][j]);
}
void add_matrix(int a[5][5],int b[5][5],int c[5][5],int row,int col)
{
int i,j;
       for(i=0;i<row;i++)</pre>
               for(j=0;j<col;j++)
                       c[i][j]=a[i][j]+b[i][j];
}
void display_matrix(int a[5][5],int row,int col)
{
int i,j;
       for(i=0;i<row;i++)</pre>
        {
               for(j=0;j<col;j++)
                {
                       printf("%d\t",a[i][j]);
                ł
               printf("\n");
        }
}
void main()
{
int a[5][5],b[5][5],c[5][5];
int i,j,p,q,r,s;
clrscr();
printf("ENTER ORDER OF A MATRIX\n");
scanf("%d%d",&p,&q);
printf("ENTER ORDER OF B MATRIX\n");
scanf("%d%d",&r,&s);
if(p==r&&q==s)
{
```

printf("ENTER A MATRIX\n");

DEPARTMENT OF CSE/IT

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

```
read_matrix(a,p,q);
printf("ENTER B MATRIX\n");
```

```
read_matrix(b,p,q);
add_matrix(a,b,c,p,q);
```

printf(" After addition of two matrices :\n"); display_matrix(c,p,q);

```
}
else
{
```

printf("Addition not possible");

```
}
```

```
getch();
}
```

INPUT:

```
ENTER ORDER OF A MATRIX
2
2
ENTER ORDER OF B MATRIX
2
2
ENTER A MATRIX
1
2
3
4
ENTER B MATRIX
1
2
3
4
OUTPUT:
After addition of two matrices :
2
    4
```

```
6 8
```

Record at least 2 results

9) b)Write a C program that uses functions to perform Multiplication of Two Matrices.

AIM:

To perform multiplication of two matrices. **Description**: Consider two matrices and their order is R1xC1 and R2XC2. The condition is C1==R2, then only the multiplication is possible.

ALGORITHM:

Step 1: Start
Step21: for i is 0 to 2 by step 1
 for j is 0 to 2 by step 1
Step 3: Read a[i][j],b[i][j]
Step 4: goto step 2
Step 5: calculate c[i][j]=c[i][j]+a[i][k]*b[k][j]
Step 6: goto step 2
Step 7: Print c[i][j]
Step 8: Stop

```
#include<stdio.h >
#include<conio.h>
int i,j,k;
 void mul(int x[10][10],int y[10][10],int z[10][10],int m,int n,int p,int q)
 {
  for (i=0;i<m;i++)
    for(j=0;j<q;j++)
    {
        z[i][j]=0;
        for(k=0;k<n;k++)
           z[i][j] += x[i][k]*y[k][j];
    }
  }
 void read(int x[10][10], int m,int n)
 {
  printf("Enter Matrix Value Row by Row\n");
  for (i=0;i<m;i++)
    for(j=0;j<n;j++)
        scanf("%d",&x[i][j]);
  }
  void display(int x[10][10], int m,int n)
 {
   for (i=0;i<m;i++)
    {
       for(j=0;j<n;j++)
        printf("%5d",x[i][j]);
        printf("\n");
    }
  printf("\n");
  }
void main()
{
 int a[10][10],b[10][10],c[10][10],m,n,p,q,;
 void mul(int x[10][10],int y[10][10],int z[10][10],int m,int n,int p,int q);
 void read(int x[10][10],int m,int n);
 void display(int x[10][10], int m,int n);
 clrscr();
```

DEPARTMENT OF CSE/IT

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

```
printf("Enter the size of A Mtrix (Row and Col): \n");
scanf("%d%d",&m,&n);
printf("Enter the size of B Mtrix (Row and Col): \n");
scanf("%d%d",&p,&q);
if(n!=p)
 {
   printf("Multiplication Not Possible\n Please re-enter\n");
   printf("correct size and try again ..... \n");
 }
else
 {
  read(a,m,n);
  read(b,m,n);
  mul(a,b,c,m,n,p,q);
  printf("A Matrix is :\n");
  display(a,m,n);
  printf("B Matrix is :\n");
  display(b,m,n);
  printf("C Matrix is :\n");
  display(c,m,n);
 }
getch();
}
```

INPUT:

Enter the size of A Mtrix (Row and Col): 2 2 Enter the size of B Mtrix (Row and Col): 2 2 Enter Matrix Value Row by Row 1 0 2 6 Enter Matrix Value Row by Row 3 4 4 2 **OUTPUT:** A matrix is: 1 0 2 6 **B** Matrix is: 4 3 4 2 MRCET **EAMCET CODE: MLRD** C matrix is: 3 4 24 20

Record at least 2 results

EXERCISE:

1. Program that uses functions to perform transpose of a given Matrices.

Week: 10

10) a) Write a C program to use function to insert a sub-string in to given main string from a given position.

AIM: To insert a string into another string from a specified position.

Description: Consider a main string and a sub string should be included at the specified position. Ex:"PROG" is the main string and the sub string is "RAM", position =4,then the output is "PROGRAM"

ALGORITHM:

Step 1: start

Step 2: read main string and sub string

Step 3: find the length of main string(r)

Step 4: find length of sub string(n)

Step 5: copy main string into sub string

Step 6: read the position to insert the sub string(p)

Step 7: copy sub string into main string from position p-1

Step 8: copy temporary string into main string from position p+n-1

Step 9: print the strings

Step 10: stop

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

FLOW CHART :



MRCET

EAMCET CODE: MLRD

```
#include<stdio.h>
#include<string.h>
main()
{
char a[30],b[30],c[30];
int pos=0,i=0,L,La,Lb,Lc,j;
       puts("Enter a string");
       gets(a);
       puts("Enter sub string");
       gets(b);
       puts("enter position for insertion");
       scanf("%d",&pos);
       La=strlen(a);
       Lb=strlen(b);
       L=pos+Lb;
       Lc=La+Lb;
       for(i=0;i<pos;i++)</pre>
        {
               c[i]=a[i];
        }
       j=0;
       for(i=pos;i<=L;i++)</pre>
        {
               c[i]=b[j];
               j++;
        }
       j=pos;
       for(i=L;i<Lc;i++)</pre>
        {
               c[i]=a[j];
               j++;
        }
       c[i]='\0';
       puts("String after Insertion is:");
       printf("%s",c);
}
```

INPUT:

Enter First String: Comer Enter Second String: put

```
MRCET
```

OUTPUT:

Enter the position where the item has to be inserted:3

Computer

Record at least 2results

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

10) b) Write a C Program to swap the values of two variables usingi) Call by Value ii) Call by Reference

AIM: To Write a C Program to swap the values of two variables using

i) Call by Value ii) Call by Reference

Description:(i)In call by value ,the values of actual arguments are passed to called function. (ii) In call by reference, the address of actual arguments are passed to called function.

ALGORITHM:

Step 1: start Step 2: read a,b Step 3: t=a; Step 4: b=b; Step 5: b=t; Step 6:print a ,b values Step 7: stop

FLOW CHART:



PROGRAM: (i)c program to swap using call by value

```
#include<stdio.h>
#include<conio.h>
                                // Declaration of function
int swap(int , int);
void main( )
{
  int a = 10, b = 20;
                                        // call by value
                                        // a and b are actual parameters
  swap(a,b);
  printf ( "\na = % d b = % d", a, b );
  getch();
}
                                      // x and y are formal parameters
int swap( int x, int y )
{
  int t ;
  t = x;
  x = y;
  y = t;
  printf ( "nx = \% d y = \% d", x, y );
}
OUTPUT:
x=20 y=10
a=10 b=20
Record at least 2 results
```

ii) Program to swap two number using call by reference.

Program:

```
#include <stdio.h>
void swap(int *a,int *b);
int main()
{
    int num1,num2;
        printf("Enter any Two Integers:");
        scanf("%d%d",&num1,&num2);
        swap(&num1,&num2);
        printf("Number1 = %d\n",num1);
```

PROGRAMMING FOR PROBLEM SOLVING LABORATORY MANUAL

```
printf("Number2 = %d",num2);
return 0;
}
void swap(int *a,int *b)
{
    int temp;
        temp=*a;
        *a=*b;
        *b=temp;
}
```

INPUT: Enter any Two Integers: 5 10 **OUTPUT:**

Number1 = 10Number2 = 5

Record at least 2results

EXERCISE:

- 1. Program to convert the given string of lower case to upper case
- 2. Program to reverse a string using strrev()

MRCET

EAMCET CODE: MLRD

Week: 11

11) a) Write a C program using user defined functions to determine whether the given string is palindrome or not.

AIM: To determine if the given string is palindrome or not.

Description : Palindrome means string on reversal should be same as original Ex: madam on reversal is also madam

ALGORITHM:

Step 1: start Step 2: read string A Step 3: copy string A into B Step 4: reverse string B Step 5: compare A &B If A equals B to got step 6 Else goto step 7 Step 6:print given string A is palindrome Step 7:print given string is not palindrome Step 8: stop

FLOW CHART:



```
#include <stdio.h>
#include <string.h>
int main()
{
  char string[25], reverse_string[25] = \{ \langle 0' \rangle \};
   int i, length = 0, flag = 0;
          printf("Enter a string \n");
             gets(string);
             for (i = 0; string[i] != '\0'; i++)
          {
               length++;
          }
            printf("The length of the string \% s' = \% d n", string, length);
             for (i = \text{length} - 1; i \ge 0; i--)
            {
               reverse_string[length - i - 1] = string[i];
            }
            for (flag = 1, i = 0; i < \text{length}; i++)
             {
               if (reverse_string[i] != string[i])
                             flag = 0;
            }
             if (flag == 1)
                         printf ("%s is a palindrome \n", string);
            else
                        printf("%s is not a palindrome \n", string);
          return 0;
          }
INPUT:
Enter a string
madam
OUTPUT:
The length of the string 'madam' = 5
madam is a palindrome
Record at least 2 results
```

11. b) 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.

AIM: To display the position or index in the string S where the string T begins, or - 1 if S doesn't contain T

ALGORITHM:

- Step 1: start
- Step 2: read the string and then displayed
- Step 3: read the string to be searched and then displayed
- Step 4: searching the string T in string S and then perform the following steps
 - i. found=strstr(S,T)
 - ii. if found print the second string is found in the first string at the position. If not go o step 5

Step 5: print the -1

Step 6: stop

FLOW CHART:



```
#include<stdio.h>
#include<string.h>
#include<conio.h>
void main()
{
 char s[30], t[20];
 char *found;
 clrscr();
       /* Entering the main string */
        puts("Enter the first string: ");
        gets(s);
       /* Entering the string whose position or index to be displayed */
        puts("Enter the string to be searched: ");
         gets(t);
       /*Searching string t in string s */
         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();
}
```

INPUT:

Enter the first string: computer Enter the string to be seareched: mp

OUTPUT:

Second string is found in the first string at 2 position

Record at least 2 results

EXERCISE:

Read two strings and write a C program for finding whether the both strings are equal or not?
 Program to swap two strings using strcpy()

Week: 12

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

AIM:

To count the number of lines, words and characters in a given list.

Description: If some number of lines of data is given ,then number of lines, words and characters are found by considering null character, white space and index of the array.

ALGORITHM:

- Step 1: Start
- Step 2: Read the text until an empty line
- Step 3: Compare each character with newline char '\n' to count no of lines
- Step 4: Compare each character with tab char '\t\' or space char ' ' to count no of words
- Step 5: Compare first character with NULL char '\0' to find the end of text
- Step 6: No of characters = length of each line of text
- Step 7: Print no of lines, no of words, no of chars
- Step 8: Stop.

FLOW CHART:



```
#include <stdio.h>
void main()
{
    char line[81], ctr;
    int i.c.
               end = 0,
               characters = 0,
               words = 0,
               lines = 0;
    printf("TYPE ANY TEXT.\n");
    printf("GIVE ONE SPACE AFTER EACH WORD.\n");
     while (end == 0)
    {
          /* Reading a line of text */
          c = 0;
          while((ctr=getchar()) != '\n')
               line[c++] = ctr;
          line[c] = '\0';
          /* counting the words in a line */
          if(line[0] == '\0')
               break ;
          else
          {
               words++;
               for(i=0; line[i] != (0';i++)
                          if(line[i] == ' ' \parallel line[i] == '\t')
                                words++;
          }
          /* counting lines and characters */
          lines = lines +1;
          characters = characters + strlen(line);
    }
    printf ("\n");
    printf("Number of lines = %d\n", lines);
    printf("Number of words = %d n", words);
    printf("Number of characters = %d\n", characters);
}
```

INPUT:

TYPE ANY TEXT GIVE ONE SPACE AFTER EACH WORD. Ramu is a good boy.

OUTPUT:

THE NUMBER OF CHARACTERS IN A GIVEN TEXT IS..18 THE NUMBER OF WORDS IN A GIVEN TEXT IS..5 THE NUMBER OF LINES IN A GIVEN TEXT IS..1

Record at least 2 results

12) b) Write a C program to find the length of the string using Pointer.

AIM: To find the length of the string using Pointer.

Description: The length of the string is known by considering whether the character in the array is null or not. If it is not null ,the count will be incremented until it becomes null.

ALGORITHM:

Step 1:start Step 2:read string Step 3: count=0;i=0 Step 4: if string[i]!='\0' count:= count +1 i:=i+1 step 5:print count step 6 stop

FLOWCHART:



MRCET

```
#include<stdio.h>
#include<conio.h>
int string_Len(char*);
void main()
{
 char str[20];
 int length;
 clrscr();
   printf("\nEnter any string : ");
 gets(str);
 length = string_Len (str);
 printf("The length of the given string %s is : %d", str, length);
 getch();
}
int string_Len (char*p) /* p=&str[0] */
{
 int count = 0;
 while (*p != '\0') {
   count++;
   p++;
 }
 return count;
}
INPUT:
```

Enter the String : mrcet **OUTPUT:** Length of the given string mrcet is : 5

Record at least 2 results

EXERCISE:

- 1. Program to print value of a variable through pointer and pointer to pointer
- 2. Program using pointer to find sum of all elements stored in an array

Week: 13

13) a) Write a C Program to Calculate Total and Percentage marks of a student using structure.

AIM: C Program to Calculate Total and Percentage marks of a student using structure.

Description: The list of marks of a student are read and total is calculated and percentage is obtained.

Ex: total=m1+m2+m3,percentage=total/max-marks*100 ALGORITHM: Step1: start step2: Create a structure with a name STUDENT step3: Include members like M1,M2,M3... within STUDENT step4: Declare a structure variable to access the member of STUDENT step5: read the structure members step6:calculate total step7:calculate percentage step8:print the structure members step7:stop

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
struct student
{
       int rl;
       char nm[20];
       int m1;
       int m2;
       int m3:
       int t;
       int max_marks;
       float per;
};
void main()
{
 struct student a;
 clrscr();
 printf(" Enter RollNo, Name, three sub-marks, and max-marks in order n");
 scanf("%d%s%d%d%d%d",&a.rl,&a.nm,&a.m1,&a.m2,&a.m3,&a.max_marks);
 a.t=a.m1+a.m2+a.m3;
 a.per=a.t/max_marks*100;
```

printf("rollno=%d\n",a.rl); printf("Name:%sk\n",a.nm); printf("m1: %d\n",a.m1); printf("m2: %d\n",a.m2); printf("m3: %d\n",a.m3); printf("total: %d\n",a.t); printf("per: %f\n",a.per); getch(); }

INPUT:

Enter RollNo, Name and three sub marks ,max_marks 12 XYZ 30 40 50 300

OUTPUT:

Rollno:12 Name:XYZ m1: 30 m2: 40 m3: 50 total: 120 per: 40.000000

Record at least 2 results

EXERCISE:

- **1.** Program to create book structure and display the contents of a book.
- 2. Program to demonstrate the size-of operator to differentiate between structure and union

Signature of faculty with date

MRCET