



Department of Computer Science & Engineering

LAB MANUAL - 15CPL16/26

1. OBJECTIVES

After completing this lab course you will be able to:

- Develop the logic for a given problem.
- Write the algorithm.
- Draw a flow chart.
- Recognize and understand the syntax and construction of C code.
- Gain experience of procedural language programming.
- Know the steps involved in compiling, linking and debugging C code.
- Understand using header files.
- Make use of different data-structures like arrays, pointers, structures and files.
- Understand how to access and use library functions.
- Understand function declaration and definition.
- Feel more confident about writing your own functions.
- Able to write some simple output on the screen as well as in the files.
- Able to write some complex programs.
- Apply all the concepts that have been covered in the theory course, and
- To know the alternative ways of providing solution to a given problem.

2. SALIENT FEATURES OF C

Some of C language characteristics that define the language and also have lead to its popularity as a programming language.

- Small size.
- Extensive use of function calls.
- Structured language.
- Low level (Bitwise) programming readily available.
- Pointer implementation - extensive use of pointers for memory, array, structures and functions.
- It has high-level constructs.
- It can handle low-level activities.
- It produces efficient programs.
- It can be compiled on a variety of computers

NOTE: THE FOLLOWING PROGRAMMS ARE EXECUTED USING CODE BLOCK NOT TURBO C.

SYLLABUS
COMPUTER PROGRAMMING LABORATORY
Subject Code: 15CPL16 / 15CPL26

I A Marks: 20
Exam Hours: 03
Exam Marks: 80

Hrs/Week: 03
Total Hrs: 48

Demonstration of Personal Computer and its Accessories. Demonstration and Explanation on Disassembly and Assembly of a Personal Computer by the faculty-in-charge. Students have to prepare a write-up on the same and include it in the Lab record and evaluated.

Laboratory Session-1: Write-up on Functional block diagram of Computer, CPU, Buses, Mother Board, Chip sets, Operating System & types of OS, Basics of Networking & Topology and NIC.

Laboratory Session-2: Write-up on RAM, SDRAM, FLASH memory, Hard disks, Optical media, CD-ROM/R/RW, DVDs, Flash drives, Keyboard, Mouse, Printers and Plotters. Introduction to flowchart, algorithm and pseudo code.

Note: These TWO Laboratory sessions are used to fill the gap between theory classes and practical sessions. Both sessions are evaluated as lab experiments.

Laboratory Experiments implement the programs with WINDOWS / LINUX platform using appropriate C Compiler.

1. Design and develop a flowchart or an algorithm that takes three coefficients (*a*, *b*, and *c*) of a Quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots. Implement a C program for the developed flowchart/algorithm and execute the same to output the possible roots for a given set of coefficients with appropriate messages.

2. Design and develop an algorithm to find the *reverse* of an integer number **NUM** and check whether it is **PALINDROME** or **NOT**. Implement a C program for the developed algorithm that takes an integer number as input and output the reverse of the same with suitable messages. Ex: Num: **2014**, Reverse: **4102**, Not a Palindrome.

3a. Design and develop a flowchart to find the square root of a given number *N*. Implement a C program for the same and execute for all possible inputs with appropriate messages.

Note: **Don't use library function *sqrt(n)*.**

3b. Design and develop a C program to read a *year* as an input and find whether it is *leap year* or not. Also consider end of the centuries.

4. Design and develop an algorithm for evaluating the polynomial $f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$, for a given value of *x* and its coefficients using Horner's method. Implement a C program for the same and execute the program for different sets of values of coefficients and *x*.

5. Draw the flowchart and Write C Program to compute **Sin(x)** using Taylor series approximation given by $\text{Sin}(x) = x - (x^3/3!) + (x^5/5!) - (x^7/7!) + \dots$. Compare the result with the built-in Library function and print both the results with appropriate messages.

6. Develop an algorithm, implement and execute a C program that reads *N* integer numbers and arrange them in ascending order using **Bubble Sort**.

7. Develop, implement and execute a C program that reads two matrices **A (m x n)** and **B(p x q)** and Compute the product **A** and **B**. Read matrix **A** and matrix **B** in row major order and in column major order

respectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of in compatibility.

8. Develop, implement and execute a C program to search a Name in a list of names using **Binary searching** Technique.

9. Write and execute a C program that

- i. Implements string copy operation **STRCOPY**(str1, str2) that copies a string *str1* to another string *str2* without using library function.
- ii. Reads a *sentence* and prints frequency of each of the vowels and total count of consonants.

10a. Design and develop a C function **RightShift**(*x*, *n*) that takes two integers *x* and *n* as input and returns value of the integer *x* rotated to the right by *n* positions. Assume the integers are unsigned. Write a C program that invokes this function with different values for *x* and *n* and tabulate the results with suitable headings.

10b. Design and develop a C function **isprime** (num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given range.

11. Draw the flowchart and write a *recursive* C function to find the factorial of a number, ***n!***, defined by $fact(n)=1$, if $n=0$. Otherwise $fact(n)=n*fact(n-1)$. Using this function, write a C program to compute the binomial coefficient ***nCr***. Tabulate the results for different values of ***n*** and ***r*** with suitable messages.

12. Given two university information files “**studentname.txt**” and “**usn.txt**” that contains students Name and USN respectively. Write a C program to create a new file called “**output.txt**” and copy the content of files “studentname.txt” and “usn.txt” into output file in the sequence shown below. Display the contents of output file “output.txt” on to the screen.

Student Name

USN

Name 1

USN1

Name 2

USN2

.....

.....

.....

13. Write a C program to maintain a record of “***n***” student details using an array of structures with four fields (Roll number, Name, Marks, and Grade). Assume appropriate data type for each field. Print the marks of the student, given the student name as input.

14. Write a C program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of ***n*** real numbers. Heading

Reference Book:

1. Reema Thareja, Computer Fundamentals and Programming in C, Oxford Press, 2012.

Practical Examination Procedure:

1. All laboratory experiments (Fourteen) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
- 4. Change of experiment is allowed only once and 15% Marks allotted to the procedure part to be made zero.**

1. Design and develop a flowchart or an algorithm that takes three coefficients (a, b, and c) of a Quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots. Implement a C program for the developed flowchart/algorithm and execute the same to output the possible roots for a given set of coefficients with appropriate messages.

This program uses quadratic equation **quadratic equation** (from the Latin *quadratus* for "square") is any equation having the form

Where x represents an unknown, and a , b , and c are constants with a not equal to 0.

The above equation can be solved using the formula

We get two roots from this formula x_1 , x_2 .and we assign b^2-4ac to d

First the value of d is found. Then the conditions are checked

If $d>0$ the roots will be distinct

If $d==0$ the roots will be equal

If $d<0$ the roots will be complex

Program:

```
#include <stdio.h>
#include <math.h>
void main()
{
    float a,b,c;
    float d,x1,x2,r;
    printf("Enter the three co-efficient :\n");
    scanf("%f%f%f",&a,&b,&c);
    if (a* b* c == 0)
    {
        printf("\n Invalid Input ");
    }
    else
    {
        d = b * b - 4 * a * c;
        r=sqrt(fabs(d));
        if (d > 0)
        {
            x1 = (-b +r) / (2.0*a);
            x2 = (-b -r) / (2.0*a);
            printf("\n The roots are real and distinct\n");
            printf("\n The roots are \n 1) x1=%f\t\t \n 2) x2=%f",x1,x2);
        }
        else if (d == 0)
        {
            x1 = x2 = -b/(2.0*a);
            printf("\n The roots are real and equal\n");
        }
    }
}
```


2. Design and develop an algorithm to find the *reverse* of an integer number NUM and check whether it is PALINDROME or NOT. Implement a C program for the developed algorithm that takes an integer number as input and output the reverse of the same with suitable messages. Ex: Num: 2014, Reverse: 4102, Not a Palindrome

This program uses the while statement

A **while** loop statement in C programming language repeatedly executes a target statement as long as a given condition is true.

Syntax:

The syntax of a **while** loop in C programming language is:

```
while(condition)
{
    statement(s);
}
```

statement(s) may be a single statement or a block of statements. The **condition** may be any expression, and true is any nonzero value. The loop iterates while the condition is true.

This program uses while loop to find the least significant digit, separate least significant digit and to reverse the given number. Once the number is reversed the, reversed number is compared with original number to check it is palindrome or not.(A palindromic**number** or numeral **palindrome** is a **number** that remains the same when its digits are reversed. Like 16461, for example, it is "symmetrical". The term palindromic is derived from **palindrome**, which refers to a word (such as rotor or racecar) whose spelling is unchanged when its letters are reversed.)

Program

```
#include <stdio.h>
void main()
{
    int n,rev=0,rem,m;
    printf("Enter a number : ");
    scanf("%d",&n);
    m = n;
    if(n<=999 || n>9999)
    {
        printf(" Not a 4 digit number\n");
        exit(0);
    }
    while(n != 0)
    {
        rem=n%10;
        n = n / 10;
        rev= rev*10+rem;
    }
    if(m==rev)
    {
        printf("\n The given Number %d is Palindrome",m);
    }
}
```

```
    }  
else  
{  
    printf("\n The given Number %d is not Palindrome",m);  
}  
}
```

Sample Output 1:

Enter a number:
201
Not a four digit number

Sample Output 2:

Enter a number:
5642
The Number is not Palindrome

Sample Output 3:

Enter a number:
1221
The Number is Palindrome

3a. Design and develop a flowchart to find the square root of a given number N . Implement a C program for the same and execute for all possible inputs with appropriate messages. Note: Don't use library function $\text{sqrt}(n)$.

This program uses function

A function is a group of statements that together perform a task. Every C program has at least one function, which is **main** (), and all the most trivial programs can define additional functions.

Function will have calling function and called function where calling function will be in main function, when a function is called the control is transferred from main () to function definition (called function).

After executing the last statement ie return the control will be passed to function main.

In program the calling function is main () and called function is squareroot() which returns the power of two numbers and returns result to main.

Program

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
float squareroot(float n);
void main()
{
    float n,result;
    printf("Enter a number\n");
    scanf("%f",&n);
    result=squareroot(n);
    printf("square root of %f is %f",n,result);
}
float squareroot(float num)
{
    float x=num;
    if(num>=0)
    {
        int i;
        for(i=0;i<20;i++)
        {
            x = ((x*x)+num) / (2*x);
        }
    }
    return x;
}
```

Sample Output 1:

```
Enter a number
4
Square root of 4 is 2
```

Sample Output 2:

```
Enter a number
100
Square root of 100 is 10
```

3b. Design and develop a C program to read a year as an input and find whether it is leap year or not. Also consider end of the centuries.

This program uses nested if_statement

An if or if-else statement within another if or if-else statement is called nested if statement.

Program

```
#include <stdio.h>
void main()
{
    int year;
    printf("Enter a year: ");
    scanf("%d",&year);
    if((year%4 == 0 && year%100 != 0) || year%400 == 0)
    {
        printf("%d is a leap year.", year);
    }
    else
    {
        printf("%d is not a leap year.", year);
    }
}
```

Sample Output 1:

```
Enter a year:
2000
2000 is a leap year
```

Sample Output 2:

```
Enter a year:
1997
1997 is not a leap year
```

4. Design and develop an algorithm for evaluating the polynomial $f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$, for a given value of x and its coefficients using Horner's method. Implement a C program for the developed algorithm and execute for different sets of values of coefficients and x .

Program

```
#include <stdio.h>
void main()
{
    int n,i,x,a[10],sum=0;
    printf("\n Enter the degree of the polynomial : ");
    scanf("%d",&n);
    printf("\n Enter the %d coefficients\n",n+1);
    for(i=0; i<=n; i++)
    {
        scanf("%d",&a[i]);
    }
    printf("\n Enter the value of x :");
    scanf("%d",&x);
    sum=a[n]*x;
    for(i=n-1; i>=1; i--)
    {
        sum = (sum+a[i])*x;
    }
    sum=sum+a[0];
    printf("\n The sum of polynomial = %d",sum);
}
```

Sample Output 1:

```
Enter the degree of the polynomial : 4
Enter the 5 coefficient
1 2 3 4 5
Enter the value of x : 1
The sum of polynomial = 15
```

Sample Output 2:

```
Enter the degree of the polynomial : 4
Enter the 5 coefficient
1 2 3 4 5
Enter the value of x : 2
The sum of polynomial = 129
```

5. Write C Program to compute Sin(x) using Taylor series approximation given by $\text{Sin}(x) = x - (x^3/3!) + (x^5/5!) - (x^7/7!) + \dots$. Compare the result with the built-in Library function and print both the results.

This program uses Taylor's series. In mathematics, a **Taylor series** is a representation of a function as an infinite sum of terms that are calculated from the values of the function's derivatives at a single point.

Taylor series expansion for sin(x), we obtain as the value of n increases the accuracy increases so the value of n must be very large and we cannot anticipate the value of n to get accurate result better approach is to keep adding the terms till term is smaller than the possible smallest fraction.

Program

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

float mysine(float x, int n);
int fact(int n);

void main()
{
    int x,n;
    float rad,res;
    printf("enter the degree and number of terms \n");
    scanf("%d%d",&x,&n);
    rad=(x*3.14)/180;
    res=mysine(rad,n);
    printf("my defined sine function (%d)=%f \n",x,res);
    printf("\n using library function sine(%d)=%f \n",x,sin(rad));
    getch();
}

float mysine(float x, int n)
{
    int i;
    float sum=0;
    for(i=1; i<=n; i+=2)
    {
        if((i-1)%4==0)
            sum=sum+(pow(x,i)/fact(i));
        else
            sum=sum-(pow(x,i)/fact(i));
    }
    return sum;
}

int fact(int n)
{
    if(n == 0)
        return 1;
    return n*fact(n-1);
}
```

Sample Output 1:

```
enter the degree and number of terms
90
3
my defined sine function (90)=0.925018
using library function sine(90)=1.000000
```

Sample Output 2:

```
enter the degree and number of terms
30
2
my defined sine function (30)=0.523333
using library function sine(30)=0.499770
```

6. Develop, implement and execute a C program that reads N integer numbers and arrange them in ascending order using *Bubble Sort* technique.

This program uses both bubble sort and binary search

Bubble sort, sometimes referred to as **sinking sort**, is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm gets its name from the way smaller elements "bubble" to the top of the list. Because it only uses comparisons to operate on elements, it is a comparison sort.

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[50], n, i, j, temp = 0;
    printf("Enter the numbers of elements:\n");
    scanf("%d", &n);
    printf("Enter the %d elements:\n", n);
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    printf("\n \n \t The given array is: \n");

    for (i = 0; i < n; i++)
    {
        printf("\n\t\t %d", a[i]);
    }
    for (i = 1; i < n; i++)
    {
        for (j = 0 ; j < n-i; j++)
        {
            if (a[j] > a[j+1])
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
    printf("\n\n\n\t\tThe sorted array using Buble sort is:\n");
    for (i = 0; i < n; i++)
    {
        printf("\n\t\t %d", a[i]);
    }
    getch();
}
```

}
Out Put

Sample Output 1:

Enter the numbers of elements:

5

Enter the 5 elements:

5 4 3 2 1

The given array is:

5

4

3

2

1

The sorted array using Buble sort is:

1

2

3

4

5

7. Develop, implement and execute a C program that reads two matrices A($m \times n$) and B($p \times q$) and Compute the product A and B. Read matrix A in row major order and matrix B in column major order. Print both the input matrices and resultant matrix with suitable headings and in matrix format. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.

This program uses matrix multiplication. Matrix multiplication is a binary operation that takes a pair of matrices, and produces another matrix. Matrix multiplication is often called the matrix product. In words, if A is an $n \times m$ matrix and B is a $m \times p$ matrix, their matrix product AB is an $n \times p$ matrix, in which the m entries across the rows of A are multiplied with the m entries down the columns of B.

Program:

```
#include <stdio.h>
#include<conio.h>
void main()
{
    int a[10][10],b[10][10],c[10][10];
    int m,n,p,q,i,j,k;
    printf("\n Enter the order of the matrix A :");
    scanf("%d%d",&m,&n);
    printf("\n Enter the order of the matrix B :");
    scanf("%d%d",&p,&q);
    if(n!=p)
    {
        printf("Matrix A & B is not multiplicable");
        getch();
    }
    printf("\n Enter the elements of matrix A \n");
    for(i = 0 ; i < m ; i++)
    {
        for(j = 0 ; j < n ; j++)
            scanf("%d",&a[i][j]);
    }
    printf("\n Enter the elements of matrix B \n");
    for(i = 0 ; i < p ; i++)
    {
        for(j = 0 ; j < q ; j++)
            scanf("%d",&b[i][j]);
    }
    for(i = 0 ; i < m ; i++)
    {
        for(j = 0 ; j < q ; j++)
        {
            c[i][j]=0;
            for(k = 0 ; k < n ; k++)
                c[i][j] += a[i][k] * b[k][j];
        }
    }
    printf("\n MATRIX A \n");
    for(i = 0 ; i < m ; i++)
```



```
{
    for(j = 0 ; j < n ; j++)
    {
        printf(" %d \t", a[i][j]);
    }
    printf("\n");
}
printf("\n MATRIX B \n");
for(i = 0 ; i < p ; i++)
{
    for(j = 0 ; j < q ; j++)
    {
        printf(" %d \t", b[i][j]);
    }
}
printf("\n");
}
printf("\n MATRIX C \n");
for(i = 0 ; i < m ; i++)
{
    for(j = 0 ; j < q ; j++)
    {
        printf(" %d \t", c[i][j]);
    }
}
printf("\n");
}
}
```

Sample Output 1:

Enter the order of the matrix A :2 2
Enter the order of the matrix B :2 2
Enter the elements of matrix A
1 2
3 4
Enter the elements of matrix B
2 3
4 5
MATRIX A
1 2
3 4
MATRIX B
2 3
4 5
MATRIX C
10 13
22 29

Output 2:

Enter the order of the matrix A :2 3
Enter the order of the matrix B :2 3
Matrix A & B is not multiplicable.

8. Develop, implement and execute a C program to search a Name in a list of names using **Binary Search**.

Program:

```
#include<stdio.h>
void main()
{
    int start=1, end=5, mid,i,flag=0,value;
    char name[25][20], nm[15];
    printf("Enter 5 names in alphabetic order: \n");
    for (i=1; i<=5; i++)
        scanf("%s",name[i]);
    printf("\n Enter the name to be search \n ");
    scanf("%s",nm);
    mid = (start+end)/2;
    while(strcmp(nm,name[mid])!=0 && start<=end)
    {
        value = strcmp(nm,name[mid]);
        if(value > 0)
        {
            start = mid + 1;
            mid = (start + end) / 2;
        }
        else
        {
            end = mid - 1;
            mid = (start + end) / 2;
        }
    }
    if(strcmp(nm,name[mid]) == 0)
    {
        flag = 1;
    }
    if(flag == 1)
        printf("\n The name %s is found at position %d successfully",nm,mid);
    else
        printf("\n The name %s is not found",nm);
}
```

Output 1:

Enter 5 names in alphabetic order:

Abhi

Brinda

Chethan

Dheeraj

Enrique

Enter names to be search:

Brinda

The name Brinda is found at position 2 successfully

Output 2:

Enter 5 names in alphabetic order:

Abhi

Brinda

Chethan

Dheeraj

Enrique

Enter names to be search:

ManiSharma

The name ManiSharma is not found

9. Write and execute a C program that

- i. Implements string copy operation **STRCOPY** (str1,str2) that copies a string str1 to another string str2 without using library function.
- ii. Reads a sentence and prints frequency of each of the vowels and total count of consonants.

Program I:

```
#include<stdio.h>

void myStrCopy(char dest[ ], char src[ ] );
void main()
{
    char src[50],dest[50];

    printf("\n Enter the Source String: ");
    scanf("%s",src);
    myStrCopy(dest, src);
    printf("\n AFTER COPYING the destination String = %s .",dest);
}

void myStrCopy(char dest[ ], char src[ ] )
{
    int i=0;
    while(src[i]!='\0')
    {
        dest[i]=src[i];
        i++;
    }
    dest[i]='\0';
}
```

Output:

```
Enter the Source String:
Spr
AFTER COPYING the destination String = Spr
```

Program II

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

void main()
{
    char str[50],ch;
    int j,con_count;
    printf("\n Enter the sentence: ");
    gets(str);
    con_count = 0;
    int a=0, e=0, i=0, o=0, u=0;
    for(j=0; j<strlen(str);j++)
```

```
{
    ch = tolower(str[j]);
    if(ch=='a' ||ch=='e' ||ch=='i' ||ch=='o' ||ch=='u' || ch ==' ')
    {
        if(ch=='a')
        {
            a++;
        }
        if(ch=='e')
        {
            e++;
        }
        if(ch=='i')
        {
            i++;
        }
        if(ch=='o')
        {
            o++;
        }
        if(ch=='u')
        {
            u++;
        }
    }
    else
        con_count++;
}

printf("\n Frequency of vowel 'a'=%d ",a);
printf("\n Frequency of vowel 'e'=%d ",e);
printf("\n Frequency of vowel 'i'=%d ",i);
printf("\n Frequency of vowel 'o'=%d ",o);
printf("\n Frequency of vowel 'u'=%d ",u);

printf("\n Number of Consonants= %d \n",con_count);
}
```

Output:

Enter the sentence:

india is my country

Frequency of vowel 'a'=1

Frequency of vowel 'e'=0

Frequency of vowel 'i'=3

Frequency of vowel 'o'=1

Frequency of vowel 'u'=1

Number of Consonants = 10

10a. Design and develop a C function RightShift(x ,n) that takes two integers x and n as input and returns value of the integer x rotated to the right by n positions. Assume the integers are unsigned. Write a C program that invokes this function with different values for x and n and tabulate the results with suitable headings.

Program:

```
#include<stdio.h>

unsigned int rightshift(unsigned int x,int n);
void main()
{
    unsigned int x;
    unsigned int res;
    int n;
    printf("\n enter an unsinged integer<=65535\n");
    scanf("%u",&x);
    printf("\n rotate %u how many times: \n",x);
    scanf("%d",&n);
    res=rightshift(x,n);
    printf("\n rightshift(%u,%d)=%u\n",x,n,res);
}

unsigned int rightshift(unsigned int x,int n)
{
    int i;
    for(i=1;i<=n; i++)
    {
        if(x%2==0)
            x=x>>1;
        else
        {
            x=x>>1;
            x=x+32768;
        }
    }
    return x;
}
```

Output 1:

```
enter an unsinged integer<=65535
10
rotate 10 how many times:
1
rightshift(10,1) = 5
```

10 b. Design and develop a C function **isprime(num)** that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given ranges.

```
#include<stdio.h>

int isprime(int m);

void main()
{
    int i,n1,n2;
    printf("Enter a Range\n");
    scanf("%d%d",&n1,&n2);
    printf("The prime Numbers are between %d and %d are\n",n1,n2);
    for(i=n1;i<=n2;i++)
    {
        if(isprime(i))
            printf("%d\t",i);
    }
}

int isprime(int m)
{
    int i;
    for(i=2;i<=m/2;i++)
    {
        if(m%i == 0)
        {
            return 0;
        }
    }
    return 1;
}
```

Output 1:

Enter a Range

5

30

The prime Numbers are between 5 and 30 are

5 7 11 13 17 19 23 29

11. Draw the flowchart and write a recursive C function to find the factorial of a number, n!, defined by $fact(n)=1$, if $n=0$. Otherwise $fact(n)=n*fact(n-1)$. Using this function, write a C program to compute the binomial coefficient (nCr) . Tabulate the results for different values of n and r with suitable messages.

Program:

```
#include<stdio.h>
long int fact(int a);
void main()
{
    long int n,r,ncr;
    printf("Enter the values of n and r \n");
    scanf("%ld%ld",&n,&r);
    ncr=fact(n)/((fact(n-r))*(fact(r)));
    printf("ncr=%d\n",ncr);
}
long int fact(int a)
{
    if(a==0)
        return 1;
    else
        return a*fact(a-1);
}
```

Output :

```
Enter the values of n and r
5
2
ncr=10
```

12. Given two university information files “studentname.txt” and “usn.txt” that contains students Name and USN respectively. Write a C program to create a new file called “output.txt” and copy the content of files “studentname.txt” and “usn.txt” into output file in the sequence shown below. Display the contents of output file “output.txt” on to the screen.

Student Name	USN	Heading
Name 1	USN1	
Name 2	USN2	
....	
....	

NOTE: It is required to create two files "studname.txt" and "studusn.txt" with the contents.

Program:

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

void main()
{
    FILE *fp1,*fp2,*fp3;
    char usn[20],name[20];
    fp1=fopen("studname.txt","r");
    if(fp1==NULL)
    {
        printf("File not found");
        exit(0);
    }

    fp2=fopen("studusn.txt","r");
    if(fp2==NULL)
    {
        printf("File not found");
        exit(0);
    }

    fp3=fopen("output.txt","w");
    while(!feof(fp1)&&!feof(fp2))
    {
        fscanf(fp1,"%s",name);
        fscanf(fp2,"%s",usn);
        fprintf(fp3,"%s%10s\n",name,usn);
    }

    fclose(fp1);
    fclose(fp2);
    fclose(fp3);
    fp3=fopen("output.txt","r");
```

```
printf("\n..... \n");
printf("Name USN \n");
printf("..... \n");
while(!feof(fp3))
{
    fscanf(fp3,"%s",name);
    fscanf(fp3,"%s\n",usn);
    printf("%-15s %-10s\n",name,usn);
}
fclose(fp3);
}
```

Output:

studname.txt	studusn.txt
Arun	3BK14IS001
Kumar	3BK14IS002
Raj	3BK15CS001
shashi	3BK15CS002
Tarun	3BK15CS003

13. Write a C program to maintain a record of “n” student details using an array of structures with four fields (Roll number, Name, Marks, and Grade). Each field is of an appropriate data type. Print the marks of the student given student name as input.

Program:

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

struct student
{
    char name[50];
    int roll;
    int marks;
    char grade;
};

void main()
{
    struct student s[10];
    int i,flag=0,n;
    char dname[50];
    printf("Enter the no: of students");
    scanf("%d",&n);
    printf("Enter information of students:\n");
    for(i=0;i<n;i++)
    {
        fflush(stdin);
        printf("\nEnter the roll number ");
        scanf("%d",&s[i].roll);
        printf("Enter name: ");
        scanf("%s",s[i].name);
        printf("Enter marks: ");
        scanf("%d",&s[i].marks);
        fflush(stdin);
        printf("Enter the Grade(A,B,C)");
        scanf("%c",&s[i].grade);
        printf("\n");
    }
    printf("Enter the name of student whose details need to be displayed");
    scanf("%s",dname);
    for(i=0;i<n;i++)
    {
        if(strcmp(s[i].name,dname)==0)
        {
            printf("The Details are:\n");
            printf("\nRoll number: %d",s[i].roll);
            printf("\nName: ");
            puts(s[i].name);
        }
    }
}
```

```
        printf("\nMarks: %d",s[i].marks);
        printf("\nGrade: %c",s[i].grade);
        flag=1;
        break;
    }
}
if(flag==0)
printf("The student details are not found");
getch();
}
```

Output:

Enter the no: of students:

2

Enter

information

of

students:

Enter the roll number: 1

Enter name: Mohit

Enter marks: 85

Enter

the

Grade(A,B,C)

:A

Enter the roll number : 2

Enter name: Rahul

Enter marks: 75

Enter the Grade(A,B,C) :B

Enter the name of student whose details need to be displayed:

Rahul

The Details are:

Roll number: 2

Name: Rahul

Marks: 75

Grade: B

14. Write a C program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers.

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

void main()
{
    float a[10],*ptr, mean, std, sum=0,sumstd=0;
    int n, i;
    printf("Enter the no of elements\n");
    scanf("%d",&n);
    printf("Enter the array elements\n");
    for(i=0;i<n;i++)
    {
        scanf("%f",&a[i]);
    }
    ptr=a;
    for(i=0;i<n;i++)
    {
        sum=sum+*ptr;
        ptr++;
    }
    mean=sum/n;
    ptr=a;
    for(i=0;i<n;i++)
    {
        sumstd=sumstd+pow((*ptr-mean),2);
        ptr++;
    }
    std=sqrt(sumstd/n);
    printf("sum=%f\n",sum);
    printf("Mean=%f\n",mean);
    printf("standard deviation=%f\n",std);
    getch();
}
```

Output:

```
Enter the no of elements
5
Enter the array elements
1 2 3 4 5
sum=15.000000
Mean=3.000000
standard deviation=1.414214
```

VIVA QUESTIONS

- 1) What is an algorithm?
- 2) What is high level language?
- 3) What is compiler?
- 4) What are tokens?
- 5) What are identifiers?
- 6) What are keywords? How many keywords is their in C programming language?
- 7) What is a variable?
- 8) What are the rules to be followed while declaring a variable?
- 9) What is a constant?
- 10) What is a datatype? What are the different datatypes?
- 11) What are escape sequence characters?
- 12) List the size and range of basic datatypes.
- 13) What is the difference between a character and string?
- 14) What is implicit type conversion and explicit type conversion (type casting)?
- 15) What is precedence of an operator means?
- 16) What is the difference between printf() and puts() functions.
- 17) What is function? What are the advantages of functions?
- 18) What are the different types of functions?
- 19) What is a library function?
- 20) What is calling function and called function?
- 21) What is the meaning of actual parameter and formal parameter?
- 22) What is the purpose of switch statement? Explain with syntax.
- 23) What is loop? List the differences between pre-test and post-test loop.
- 24) What is the meaning of event controlled loop and counter controlled loop?
- 25) What are the advantages of loops?
- 26) What is control statement? What are the various types of control statements available in C language?
- 27) Explain for loop with syntax.
- 28) What is the difference between while and do-while loop?
- 29) What are unconditional control statements?
- 30) What is the use of break statement?
- 31) What is an array? What is the difference between an ordinary variable and an array variable?
- 32) What are the differences between recursion and iteration?
- 33) What is a pointer?
- 34) What is a NULL pointer?
- 35) What is a Structure? What are the differences between structures and arrays?
- 36) What is memory leak? Why it should be avoided.
- 37) Which header file should be included to use functions like malloc() and calloc()?
- 38) List string.h Library functions in C.
- 39) What is the purpose of main() function?
- 40) What is the difference between static and dynamic memory allocation in C?