

Lab1 Simple computational problems using arithmetic expressions.

a. For the given Basic salary, compute DA, HRA and PF using the following criteria and write a C program to find out the Net Salary of an Employee by deducting PF and IT.

$$DA = (\text{Basic salary} * 25) / 1000$$

$$HRA = (\text{Basic salary} * 15) / 100$$

$$\text{Gross salary} = \text{Basic salary} + DA + HRA$$

$$PF = \text{Gross salary} * 10 / 100$$

$$IT = \text{Gross salary} * 10 / 100$$

$$\text{Net Salary} = \text{Basic Salary} + DA + HRA - (PF + IT)$$

program::-

```
#include <stdio.h>
void main()
{
    int b;
    float d,h,g,p,i,n;
    printf("enter basic salary of employee\n");
    scanf("%d",&b);
    d=0.25*b,h=0.15*b;
    g=b+d+h;
    p=0.10*g;
    i=(g*10)/100;
    n=g-(p+i);
    printf("The net salary of employee is Basic salary+DA+HRA-PF-IT= %d+ %f+ %f- (%f+%f)=%f",b,d,h,p,i,n);
}
```

Output1:

enter basic salary of employee

25000

The net salary of employee is Basic salary+DA+HRA-PF-IT= 25000+ 6250.000000+ 3750.000000- (3500.000000+3500.000000)=28000.000000

[Program finished]

Output 2:

enter basic salary of employee

48000

The net salary of employee is Basic salary+DA+HRA-PF-IT= 48000+ 12000.000000+ 7200.000000- (6720.000000+6720.000000)=53760.000000

[Program finished]

b. Write a program to Convert the given minutes into days, hours and remaining minutes

Ex: M=2000 Answer is 2000 minutes contain 1 day 9 Hours 20 Minutes

Program:-

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

```
void main()
```

```
{
```

```
    int m,d,f,h;
```

```
    printf("enter the minutes\n");
```

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

```
    h=m/60;
```

```
    m=m%60;
```

```
    d=h/24;
```

```
    h=h%24;
```

```
    printf("%d days,%d hours and %d minutes",d,h,m);
```

```
}
```

Output 1:

enter the minutes

3000

2 days,2 hours and 0 minutes

[Program finished]

Output 2:

enter the minutes

8000

5 days,13 hours and 20 minutes

[Program finished]

Lab2 Problems involving if-then-else & switch.

a. The marks obtained by a student in 'n' different subjects are given as an input by the user. Write a C program that calculates the average marks of given n subjects and display the grade. The student gets a grade as per the following rules:

Average	Grade
90-100	O
80-89	E
70-79	A
60-69	B
50-59	C
<50	F

Program:-

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

```
void main()
```

```
{
```

```
    int i,n,sum=0,a[20];
```

```
    float avg;
```

```

printf("enter number of subjects\n");
scanf("%d",&n);
printf("enter marks in each subject\n");
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
    sum=sum+a[i];
}
float m=n;
avg=sum/m;
printf("The average marks of student is %f\n",avg);
if(avg>=90&&avg<=100)
printf("The grade obtained is 'O'\n");
else if(avg>=80&&avg<=89)
printf("The grade obtained is 'E'\n");
else if(avg>=70&&avg<=79)
printf("The grade obtained is 'A'\n");
else if(avg>=60&&avg<=69)
printf("The grade obtained is 'B'\n");
else if(avg>=50&&avg<=59)
printf("The grade obtained is 'C'\n");
else
printf("The grade obtained is 'F'\n");
}

```

Output 1:

```

enter number of subjects
5
enter marks in each subject
90
80
85
86
76
The average marks of student is 83.400002
The grade obtained is 'E'

```

[Program finished]

Output 2::

```

enter number of subjects
6
enter marks in each subject
90
80
72
65
76

```

85

The average marks of student is 78.000000

The grade obtained is 'A'

[Program finished]

b. Write a C program to find the Largest and smallest values among the given 4 values.

Program:-

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

```
void main()
```

```
{
```

```
    int a,b,c,d;
```

```
    printf("enter four numbers\n");
```

```
    scanf("%d%d%d%d",&a,&b,&c,&d);
```

```
    if(a>b&&a>c&&a>d)
```

```
        printf("Biggest number is %d\n",a);
```

```
    else if (b>c&&b>d)
```

```
        printf("Biggest number is %d\n",b);
```

```
    else if(c>d)
```

```
        printf("Biggest number is %d\n",c);
```

```
    else
```

```
        printf("Biggest number is %d\n",d);
```

```
    if(a<b&&a<c&&a<d)
```

```
        printf("Smallest number is %d\n",a);
```

```
    else if(b<c&&b<d)
```

```
        printf("Smallest number is %d\n",b);
```

```
    else if(c<d)
```

```
        printf("Smallest number is %d\n",c);
```

```
    else
```

```
        printf("Smallest number is %d\n",d);
```

```
}
```

Output 1:

enter four numbers

15

17

12

32

Biggest number is 32

Smallest number is 12

[Program finished]

Output 2:

enter four numbers

20
-30
100
25
Biggest number is 100
Smallest number is -30

[Program finished]

c. **Write a C program to Print the month name when month number is given and Print the week name when week number is given.**

Program::-

```
#include <stdio.h>
void main()
{
    int m, w;
    printf("enter month number\n");
    scanf("%d", &m);
    switch (m)
    {
        case 1:
            printf("January\n");
            break;
        case 2:
            printf("February\n");
            break;
        case 3:
            printf("March\n");
            break;
        case 4:
            printf("April\n");
            break;
        case 5:
            printf("May\n");
            break;
        case 6:
            printf("June\n");
            break;
        case 7:
            printf("July\n");
            break;
        case 8:
            printf("August\n");
            break;
        case 9:
            printf("September\n");
```

```

        break;
    case 10:
        printf("October\n");
        break;
    case 11:
        printf("November\n");
        break;
    case 12:
        printf("December\n");
        break;
    default:
        printf("error\n");
        break;
}
printf("enter week number\n");
scanf("%d", &w);
switch (w)
{
    case 1:
        printf("Sunday\n");
        break;
    case 2:
        printf("Monday\n");
        break;
    case 3:
        printf("Tuesday\n");
        break;
    case 4:
        printf("Wednesday\n");
        break;
    case 5:
        printf("Thursday\n");
        break;
    case 6:
        printf("Friday\n");
        break;
    case 7:
        printf("Saturday\n");
        break;
    default:
        printf("error\n");
        break;
}
}

```

Output1:-
enter month number

12
December
enter week number
7
Saturday

[Program finished]

Output 2:
enter month number
6
June
enter week number
2
Monday

[Program finished]

d. **Write a C program to Check whether the given character is numeric or non-numeric.**

Program::-

```
#include <stdio.h>
void main()
{
    char ch;
    printf("enter character\n");
    scanf("%c",&ch);
    if (ch>=97&&ch<=122||ch>=65&&ch<=90)
        printf("non numeric\n");
    else
        printf("numeric\n");
}
```

Output 1:
enter character
b
non numeric

[Program finished]

Output 2:
enter character
98
numeric

[Program finished]

Lab3 Iterative problems.

a. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: $1+x+x^2+x^3+\dots+x^n$. For example: if n is 3 and x is 5, then the program computes $1+5+25+125$. Print x, n, the sum. Perform error checking. For example, the formula does not make sense for negative exponents – if n is less than 0. Have your program print an error message if n is negative

Program::-

```
#include <stdio.h>
#include<math.h>
void main()
{
    int x,n,sum=0,i;
    printf("enter values of x and n\n");

    scanf("%d%d",&x,&n);
    if(n<0)
        printf("n should be positive\n");
    else
    {
        for(i=0;i<=n;i++)
            sum=sum+pow(x,i);

        printf("x=%d\nn=%d\nsum=%d\n",x,n,sum);
    }
}
```

Output1:

```
enter values of x and n
5
6
x=5
n=6
sum=19531
```

[Program finished]

Output2:

```
enter values of x and n
5
-6
n should be positive
```


[Program finished]

b. Write a C Program to print Floyd triangle for the user given number of rows. If the user entered 4 rows, then the output follows:

```
1
2   3
4 5 6
7 8 9 10
```

Program::-

```
#include <stdio.h>
void main()
{
    int i,j,k=0,n;
    printf("enter number of rows\n");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=i;j++)
            printf("%d\t",++k);
        printf("\n");
    }
}
```

Output1:

```
enter number of rows
6
1
2   3
4   5   6
7   8   9   10
11  12  13  14  15
16  17  18  19  20  21
```

[Program finished]

c. Write a C Program to check whether the given number is a palindrome or not.

Hint: To check whether a number is a palindrome or not, reverse the given number

And compare the reversed number with the given number, if both are same then the

Number is palindrome otherwise not.

Example: Given Number = 121, Reversed number = 121. Hence, given number is

palindrome.

Program::-

```
#include <stdio.h>
void main()
{
    int n,m,rev=0,rem;
    printf("enter n value\n");
    scanf("%d",&n);
    m=n;
    while(m!=0)
    {
        rem=m%10;
        rev=(rev*10)+rem;
        m=m/10;
    }
    if(n==rev)
    printf("The given no. is palindrome\n");
    else
    printf("The given no. is not palindrome\n");
}
```

Output 1:

enter n value

1221

The given no. is palindrome

[Program finished]

Output 2;:

enter n value

658

The given no. is not palindrome

[Program finished]

d. Write a C program to find all prime numbers, strong numbers and Armstrong numbers between the given two numbers.

Program::-

```
#include<stdio.h>
#include<math.h>
void main()
{
    int a,b,i,c,n,m,arm,p,j,f,s,r1,r2,u,v,w,k;
```

```

printf("enter two numbers\n");
scanf("%d%d",&a,&b);
printf("enter 1 for prime,2 for armstrong, 3 for strong numbers\n");
scanf("%d",&k);
switch (k)
{
    case 1:
for(u=a;u<=b;u++)
{
    c=0;
    for(i=1;i<=u;i++)
    {
        if(u%i==0)
            c++;
    }
    if(c==2)
        printf("%d",u);
}
printf("are prime numbers\n\n\n");
break;

    case 2:
for(u=a;u<=b;u++)
{
    c=0,arm=0;
    for(n=u;n!=0;n=n/10)
        c++;
    for(m=u;m!=0;m=m/10)
    {
        r1=m%10;
        arm=arm+pow(r1,c);
    }
    if(arm==u)
        printf("%d",arm);
}
printf("are armstrong numbers\n\n\n");
break;

    case 3:
for(u=a;u<=b;u++)
{
    s=0;
    for(p=u;p!=0;p=p/10)
    {
        r2=p%10;
        f=1;
        for(j=r2;j!=0;j--)
            f=f*j;
    }
}

```

```

        s=s+f;
    }
    if(s==u)
        printf("%d",u);
    }
    printf("are strong numbers\n");
    break;
default:printf("error\n");
break;
}
}

```

Output 1:

```

enter two numbers
10
20
enter 1 for prime,2 for armstrong, 3 for strong numbers
1
11,13,17,19,are prime numbers

```

[Program finished]

Output 2:

```

enter two numbers
140
160
enter 1 for prime,2 for armstrong, 3 for strong numbers
2
153,are armstrong numbers

```

[Program finished]

Output 3:

```

enter two numbers
1
150
enter 1 for prime,2 for armstrong, 3 for strong numbers
3
1,2,145,are strong numbers

```

[Program finished]

e. **Convert the given rupees into text**
Ex: 123 One Hundred and Twenty-Three Only

Program:-

```
#include <stdio.h>
void main()
{
    int n, r[5], i = 0;
    printf("enter rupees\n");
    scanf("%d", &n);
    while (n != 0)
    {
        i++;
        r[i] = n % 10;
        n = n / 10;
    }
    if (i == 0)
        printf("zero\n");
    else if (i >= 5)
        printf("enter number up to 4 digits only\n");
    else
    {
        switch (r[4])
        {
            case 1:
                printf("One");
                break;
            case 2:
                printf("Two");
                break;
            case 3:
                printf("Three");
                break;
            case 4:
                printf("Four");
                break;
            case 5:
                printf("Five");
                break;
            case 6:
                printf("Six");
                break;
            case 7:
                printf("Seven");
                break;
            case 8:
```

```

        printf("Eight");
        break;
case 9:
    printf("Nine");
    break;
}

if (i == 4)
    printf(" Thousand");
switch (r[3])
{
case 1:
    printf(" One");
    break;
case 2:
    printf(" Two");
    break;
case 3:
    printf(" Three");
    break;
case 4:
    printf(" Four");
    break;
case 5:
    printf(" Five");
    break;
case 6:
    printf(" Six");
    break;
case 7:
    printf(" Seven");
    break;
case 8:
    printf(" Eight");
    break;
case 9:
    printf(" Nine");
    break;
}

if (i >= 3 && r[3] != 0)
    printf(" Hundred and");
switch (r[2])
{
case 1:
    switch (r[1])
    {
        case 0:

```

```
        printf(" Ten");
        break;
case 1:
    printf(" Eleven");
    break;
case 2:
    printf(" Twelve");
    break;
case 3:
    printf(" Thirteen");
    break;
case 4:
    printf(" Fourteen");
    break;
case 5:
    printf(" Fifteen");
    break;
case 6:
    printf(" Sixteen");
    break;
case 7:
    printf(" Seventeen");
    break;
case 8:
    printf(" Eighteen");
    break;
case 9:
    printf(" Nineteen");
    break;
}
```

```
    break;
case 2:
    printf(" Twenty");
    break;
case 3:
    printf(" Thirty");
    break;
case 4:
    printf(" Forty");
    break;
case 5:
    printf(" Fifty");
    break;
case 6:
    printf(" Sixty");
    break;
case 7:
```

```

        printf(" Seventy");
        break;
    case 8:
        printf(" Eighty");
        break;
    case 9:
        printf(" Ninty");
        break;
    }
    if (r[2] != 1)
    {
        switch (r[1])
        {
            case 1:
                printf(" One");
                break;
            case 2:
                printf(" Two");
                break;
            case 3:
                printf(" Three");
                break;
            case 4:
                printf(" Four");
                break;
            case 5:
                printf(" Five");
                break;
            case 6:
                printf(" Six");
                break;
            case 7:
                printf(" Seven");
                break;
            case 8:
                printf(" Eight");
                break;
            case 9:
                printf(" Nine");
                break;
        }
    }
    printf(" Rupees Only");
}
}

```

Output 1:
enter rupees

150

One Hundred and Fifty Rupees Only
[Program finished]

Output 2:

enter rupees

4025

Four Thousand Twenty Five Rupees Only
[Program finished]

Lab4 1D Array manipulation.

- a. Write a C program for the following operations on a 1d array**
- i. Insert at a position**

Program::-

```
#include <stdio.h>
void main()
{
    int a[10],n,p,e,i;
    printf("enter the size of array\n");
    scanf("%d",&n);
    printf("enter the elements in array\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    printf("enter element you want to insert\n");
    scanf("%d",&e);
    printf("enter the index position\n");
    scanf("%d",&p);
    for(i=n-1;i>=p;i--)
        a[i+1]=a[i];
    a[p]=e;
    for(i=0;i<=n;i++)
        printf("%d\n",a[i]);
}
```

Output 1;:

enter the size of array

5

enter the elements in array

20

45

80

35

15

enter element you want to insert

2

enter the index position

2
20
45
2
80
35
15

[Program finished]

Output 2::

enter the size of array

6

enter the elements in array

34

56

7

5

8

75

enter element you want to insert

444

enter the index position

5

34

56

7

5

8

444

75

[Program finished]

ii. Delete an element

Program:-

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

```
void main()
```

```
{
```

```
    int i,a[10],n,pos;
```

```
    printf("enter the size of array\n");
```

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

```
    printf("enter the elements in array\n");
```

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

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

```

        printf("enter the index position of element you want to delete\n");
        scanf("%d",&pos);
        for(i=pos;i<n-1;i++)
            a[i]=a[i+1];
        n--;
        for(i=0;i<n;i++)
            printf("%d\n",a[i]);
    }

```

Output 1:

```

enter the size of array
5
enter the elements in array
55
45
30
80
90
enter the index position of element you want to delete
3
55
45
30
90

```

[Program finished]

Output 2:

```

enter the size of array
3
enter the elements in array
45
67
89
enter the index position of element you want to delete
0
67
89

```

[Program finished]

iii. Find the sum of all elements

Program:-

```

#include <stdio.h>
void main()
{

```

```

int i,a[10],sum=0,n;
printf("enter size of array\n");
scanf("%d",&n);
printf("enter the elements of array\n");
for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
    sum=sum+a[i];
}
printf("sum of array elements is %d\n",sum);
}

```

Output 1:

```

enter size of array
5
enter the elements of array
15
20
10
8
5
sum of array elements is 58

```

[Program finished]

Output 2:

```

enter size of array
4
enter the elements of array
-60
40
30
20
sum of array elements is 30

```

[Program finished]

iv. **Count of positive, negative, odd and even numbers**

Program:-

```

#include <stdio.h>
void main()
{
    int i, n, k, a[10], c = 0, g = 0, s1 = 0, s2 = 0;
    printf("enter the size of array\n");
    scanf("%d", &n);
    printf("enter array elements\n");
}

```

```

        for (i = 0; i < n; i++)
            scanf("%d", &a[i]);
        printf("enter 1 to get count of positive,negative numbers , 2 to het count of odd and
even count\n");
        scanf("%d", &k);
        switch (k)
        {
        case 1:
            for (i = 0; i < n; i++)
            {
                if (a[i] > 0)
                    c++;
                else
                    g++;
            }
            printf("No. of positive numbers=%d\n", c);
            printf("No. of negative numbers=%d\n", g);
            break;
        case 2:
            for (i = 0; i < n; i++)
            {
                if (a[i] % 2 == 0)
                    s1++;
                else
                    s2++;
            }
            printf("No. of even numbers is %d\n", s1);
            printf("No. of odd numbers is %d\n", s2);
            break;
        default:
            printf("enter 1 or 2 \n");
            break;
        }
    }
}

```

Output 1:

enter the size of array

4

enter array elements

4

-4

5

30

enter 1 to get count of positive,negative numbers , 2 to het count of odd and even count

1

No. of positive numbers=3

No. of negative numbers=1

[Program finished]

Output 2:

enter the size of array

4

enter array elements

4

5

8

3

enter 1 to get count of positive,negative numbers , 2 to get count of odd and even count

2

No. of even numbers is 2

No. of odd numbers is 2

[Program finished]

v. **Search for an element**

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

```
void main()
```

```
{
```

```
    int a[10],n,ele,low,high,mid,pos,i,flag;
```

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

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

```
    printf("enter elements in ascending order\n");
```

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

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

```
    printf("enter the element to be searched\n");
```

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

```
    low=0;high=n-1;flag=0;
```

```
    while(low<=high)
```

```
    {
```

```
        mid=(low+high)/2;
```

```
        if(ele==a[mid])
```

```
        {
```

```
            flag=1;
```

```
            pos=mid +1;
```

```
            break;
```

```
        }
```

```
        else if(ele>a[mid])
```

```
            low=mid +1;
```

```
        else
```

```
            high=mid-1;
```

```
    }
```

```
    if(flag==1)
```

```
        printf("The element is found at %d position\n",pos);
```

```
    else
```

```
        printf("The element is not found\n");
    }
```

Output 1:

Enter the size of array

5

enter elements in ascending order

5

15

20

25

80

enter the element to be searched

25

The element is found at 4 position

[Program finished]

Output 2:

Enter the size of array

5

enter elements in ascending order

23

45

68

76

100

enter the element to be searched

25

The element is not found

[Program finished]

vi. **Interchange two position values**

Program::-

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

```
void main()
```

```
{
```

```
    int n,i,a[10],pos1,pos2,t;
```

```
    printf("enter the size of array\n");
```

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

```
    printf("enter elements of array\n");
```

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

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

```
    printf("enter two index positions to interchange their values\n");
```

```
    scanf("%d%d",&pos1,&pos2);
```

```
t=a[pos1],a[pos1]=a[pos2],a[pos2]=t;
    printf("after inter changing\n");
    for(i=0;i<n;i++)
        printf("%d\n",a[i]);
}
```

Output 1:

enter the size of array

5

enter elements of array

4

5

55

6

8

enter two index positions to interchange their values

1

3

after inter changing

4

6

55

5

8

[Program finished]

Output 2:

enter the size of array

5

enter elements of array

44

65

30

66

89

enter two index positions to interchange their values

0

3

after inter changing

66

65

30

44

89

[Program finished]

Lab5 Problems on 2D arrays and Strings.

- a. Write a C program to
 - i. Read a Matrix, and find the transpose of the matrix

Program::-

```
#include <stdio.h>
void main()
{
    int a[10][10],b[10][10],i,j,m,n;
    printf("Enter the order of matrix\n");
    scanf("%d%d",&m,&n);
    printf("Enter matrix elements\n");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    for(i=0;i<n;i++)
        for(j=0;j<m;j++)
            b[i][j]=a[j][i];
    printf("\nThe transpose of matrix A is \n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
            printf("%d\t",b[i][j]);
        printf("\n");
    }
}
```

Output 1:

Enter the order of matrix

3

2

Enter matrix elements

5

6

7

2

3

4

The transpose of matrix A is

5 7 3

6 2 4

[Program finished]

Output 2:

Enter the order of matrix

3

5

Enter matrix elements

3

5

6

7

8

7

5

44

3

2

1

6

7

8

9

The transpose of matrix A is

3 7 1

5 5 6

6 44 7

7 3 8

8 2 9

[Program finished]

ii. **Addition of two matrices**

Program:-

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

```
void main()
```

```
{
```

```
    int a[10][10],b[10][10],c[10][10],i,j,m,n,p,q;
```

```
    printf("Enter the order of first matrix\n");
```

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

```
    printf("Enter first matrix elements\n");
```

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

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

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

```
    printf("Enter the order of second matrix\n");
```

```
    scanf("%d%d",&p,&q);
```

```
    printf("Enter second matrix elements\n");
```

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

```

        for(j=0;j<q;j++)
        scanf("%d",&b[i][j]);
        if(m==p&& n==q)
        {
            for(i=0;i<m;i++)
            for(j=0;j<n;j++)
            c[i][j]=a[i][j]+b[i][j];
            printf("\nThe resultant matrix after addition\n");
            for(i=0;i<m;i++)
            {
                for(j=0;j<n;j++)
                printf("%d\t",c[i][j]);printf("\n");
            }
        }
        else
        printf("Matrix addition is not possible.Order should be same for both matrices\n");
    }
}

```

Output 1:

Enter the order of first matrix

2

2

Enter first matrix elements

4

5

1

2

Enter the order of second matrix

2

2

Enter second matrix elements

2

2

3

4

The resultant matrix after addition

6 7

4 6

[Program finished]

Output 2:

Enter the order of first matrix

3

4

Enter first matrix elements

2

5
6
4
7
8
12
4
3
6
7
8
Enter the order of second matrix
3
4
Enter second matrix elements
12
3
3
34
5
6
6
8
9
7
6
5

The resultant matrix after addition

14	8	9	38
12	14	18	12
12	13	13	13

[Program finished]

iii. **Multiplication of two matrices with prompting of rules**

Program:-

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

```
void main()
```

```
{
```

```
    int a[10][10],b[10][10],c[10][10],i,j,k,m,n,p,q;
```

```
    printf("enter the first matrix order\n");
```

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

```
    printf("enter the first matrix elements\n");
```

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

```

        for(j=0;j<n;j++)
        scanf("%d",&a[i][j]);
        printf("enter second matrix order\n");
        scanf("%d%d",&p,&q);
        printf("enter second matrix elements\n");
        for(i=0;i<p;i++)
        for(j=0;j<q;j++)
        scanf("%d",&b[i][j]);
        if(n==p)
        {
            /*Resultant matrix will be of order (no. of rows in 1st matrix × no. of columns in
second matrix)*/
            for(i=0;i<m;i++)
            for(j=0;j<q;j++)
            {
                c[i][j]=0;
                for(k=0;k<n;k++)
                /*k<n or k<p as n=p*/
                c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
            }
            printf("\nResultant matrix after matrix multiplication\n");
            for(i=0;i<m;i++)
            {
                for(j=0;j<q;j++)
                printf("%d\t",c[i][j]);
                printf("\n");
            }
        }
        else
            printf("\nFor matrix multiplication,The column order of first matrix should be
equal to row order of second matrix\n");
    }

```

Output 1::

```

enter the first matrix order
2
2
enter the first matrix elements
2
3
4
6
enter second matrix order
2
2
enter second matrix elements
3
2

```

1
1

Resultant matrix after matrix multiplication

9 7
18 14

[Program finished]

Output 2:

enter the first matrix order

3
3

enter the first matrix elements

6
5
4
3
2
1
3
5
7

enter second matrix order

3
2

enter second matrix elements

3
5
2
5
6
1

Resultant matrix after matrix multiplication

52 59
19 26
61 47

[Program finished]

- b. **Write a C program that uses functions to perform the following operations:**
i. **To insert a sub string into a given main string from a given position.**

Program::-

#include <stdio.h>

```

void insert (char[]);
void main()
{
    char s1[100];
    printf("Enter the main string\n");
    scanf("%s",s1);
    insert(s1);
}
void insert(char ch[100])
{
    char s2[20];
    int i,j,p,l1,l2;
    printf("Enter the substring\n");
    scanf("%s",s2);
    printf("Enter position in main string you want to insert\n");
    scanf("%d",&p);
    for(l1=0;ch[l1]!='\0';l1++);
    for(l2=0;s2[l2]!='\0';l2++);
    ch[l1+l2]='\0';
    for(i=l1+l2-1,j=l1-1;j>=p-1;i--,j--)
        ch[i]=ch[j];
    for(i=p-1,j=0;j<l2;i++,j++)
        ch[i]=s2[j];
    printf("\nAfter inserting,the main string is\n\n");
    printf("%s\n",ch);
}

```

Output 1:

```

Enter the main string
kohli is cricketer
Enter the substring
great
Enter position in main string you want to insert
10

```

After inserting,the main string is

kohli is great cricketer

[Program finished]

Output 2:

```

Enter the main string
wnderful
Enter the substring
o
Enter position in main string you want to insert
2

```

After inserting,the main string is

wonderful

[Program finished]

ii. **To delete n characters from a given position in a given string.**

Program::-

```
#include <stdio.h>
void insert (char[]);
void main()
{
    char s1[100];
    printf("Enter the string\n");
    scanf("%s",s1);
    insert(s1);
}
void insert(char ch[100])
{
    int i,j,p,n;
    printf("Enter position and no. of characters you want to delete\n");
    scanf("%d%d",&p,&n);
    for(i=p-1,j=p+n-1;i<p+n;i++,j++)
        ch[i]=ch[j];
    ch[j]='\0';
    printf("\nAfter deleting,the string is\n");
    printf("%s",ch);
}
```

Output 1:

Enter the string

hello hai

Enter position and no. of characters you want to delete

2

4

After deleting,the string is

h hai

[Program finished]

Output 2:

Enter the string

rajeshkumar

Enter position and no. of characters you want to delete

6

3

After deleting,the string is

rajesmar

[Program finished]

c. **Write a C program to find a string within a sentence and replace it with another string.**

Program:-

```
#include <stdio.h>
#include <string.h>
void rs(char []);
void main()
{
    char z[100];
    printf("Enter a sentence\n");
    scanf("%[^\n]",z);
    rs(z);
}
void rs(char ch[100])
{
    char s[20],r[20];
    int l1,l2,l3,i,j,k,p,q,t,flag=0;
    printf("Enter the substring you want to remove \n");
    scanf("%s",s);
    l1=strlen(ch),l2=strlen(s);
    for(i=0;ch[i]!='\0';i++)
    {
        k=0;
        for(j=i;j<i+l2;j++)
        {
            if(ch[j]==s[k])
                k++;
        }
        if(k==l2)
        {
            p=t=i,q=j;
            flag=1;
            break;
        }
    }
    if(flag==0)
```

```

printf("Entered substring not found\n");
else
{
while(q<=l1)
{
    ch[p]=ch[q];
    p++;
    q++;
}
printf("Enter string you want to replace in\n");
scanf("%s",r);
l3=strlen(r);
while(p>=t)
{
ch[p+l3]=ch[p];
p--;
}
for(q=0;r[q]!='\0';q++,t++)
ch[t]=r[q];
printf("\nThe resultant string is\n\n");
puts(ch);
}
}

```

Output 1:

Enter a sentence

Rahul is a bad boy

Enter the substring you want to remove

bad

Enter string you want to replace in

good

The resultant string is

Rahul is a good boy

[Program finished]

Output 2:

Enter a sentence

Rohan is excellent

Enter the substring you want to remove

Rohan

Enter string you want to replace in

Rohini

The resultant string is

Rohini is excellent

[Program finished]

d. **Write a C program that reads a line of text and counts all occurrence of a particular word.**

Program::-

```
#include <stdio.h>
#include <string.h>
void cs(char []);
void main()
{
    char z[100];
    printf("Enter a sentence\n");
    scanf("%s",z);
    cs(z);
}
void cs(char ch[100])
{
    char s[20];
    int l1,l2,i,j,k,c=0;
    printf("Enter the word\n");
    scanf("%s",s);
    l1=strlen(ch),l2=strlen(s);
    for(i=0;ch[i]!='\0';i++)
    {
        k=0;
        for(j=i;j<i+l2;j++)
        {
            if(ch[j]==s[k])
                k++;
        }
        if(k==l2)
        {
            c++;
        }
    }
    if(c==0)
        printf("\nEntered word not found\n");
    else
    {
        printf("\nThe given word occurred %d times\n",c);
    }
}
```

Output 1:

Enter a sentence
johny johny yes papa
Enter the word
johny

The given word occurred 2 times

[Program finished]

Output 2:
Enter a sentence
He studies well. He is topper as well.
Enter the word
well

The given word occurred 2 times

[Program finished]

e. **Write a C program for the following:**

Ram wanted to increase his typing speed to participate in programming contests. His friend suggested that type the sentence “The quick brown fox jumps over the lazy dog” repeatedly. This sentence is known as a pangram because it contains every letter of the alphabet. After typing the sentence several times, Ram became bored with it so he started to look for other pangrams. For this task, read a sentence from the user and store it in a character array ‘s’ Hint: Allocate memory for the string using dynamic memory allocation and determine whether the given string is a pangram(or not. Ignore upper or lower cases

Program:-

```
#include <stdio.h>
#include<stdlib.h>
#include <ctype.h>
int main ()
{
    char *s,j;
    int x,y,k;
    s=(char*)malloc(sizeof(char));
    printf("Enter sentence\n");
    scanf("%[^\n]",s);
    x=0;
    for(j=65;j<=90;j++)
    {
        y=0;
        for(k=0;s[k]!='\0';k++)
        {
```

```

                if(j==toupper(s[k]))
                    y++;
            }
            if(y>0)
                x++;
        }
        if(x==26)
            printf("\nThe given sentence is pangram\n");
        else
            printf("\nThe given sentence is not pangram\n");
    }

```

Output 1:
Enter sentence
qwerty

The given sentence is not pangram

[Program finished]

Output 2:
Enter sentence
The quick brown fox jumps over the lazy dog

The given sentence is pangram

[Program finished]

Lab6 Function calling mechanisms (Call by value).

a. Write a C program to find prime Fibonacci numbers using functions.

Program:-

```

#include<stdio.h>
void pf(int);
void main()
{
    int n;
    printf("Enter no. of prime number elements for in fibonacci series\n");
    scanf("%d",&n);
    pf(n);
}
void pf(int b)
{
    int a[50],i,c,j,d=1;
    a[0]=0,a[1]=1;
    for(i=2;i<2*(b+1);i++)//Out of 2(b+1) numbers,b prime numbers will be printed
        a[i]=a[i-1]+a[i-2];
}

```

```

printf("\nThe prime fibonacci series is \n");
for(i=0;i<2*(b+1);i++)
{
    c=0;
    for(j=1;j<=a[i];j++)
    {
        if(a[i]%j==0)
            c++;
    }
    if(c==2&&d<=b)
    {
        printf("%d,",a[i]);
        d++;
    }
}
}

```

Output 1:

Enter no. of prime number elements for in fibonacci series

3

The prime fibonacci series is

2,3,5,

[Program finished]

Output 2:

Enter no. of prime number elements for in fibonacci series

7

The prime fibonacci series is

2,3,5,13,89,233,

[Program finished]

b. Write a C program to find npr, ncr values by writing factorial function.

Program::

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

```
void npcr(int,int);
```

```
void main()
```

```
{
```

```
    int n,r;
```

```
    printf("Enter n and r values\n");
```

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

```
    npcr(n,r);
```

```
}
```

```
void npcr(int m,int p)
```

```
{
```

```

    int i;
    float f1,f2,f3;
    f1=f2=f3=1;
    for(i=1;i<=m;i++)
        f1=f1*i;
    for(i=1;i<=p;i++)
        f3=f3*i;
    for(i=1;i<=m-p;i++)
        f2=f2*i;
    printf("The values of npr and ncr are %f and %f\n",f1/f2,f1/(f2*f3));
}

```

Output 1:

Enter n and r values

5

2

The values of npr and ncr are 20.000000 and 10.000000

[Program finished]

Output 2::

Enter n and r values

6

3

The values of npr and ncr are 120.000000 and 20.000000

[Program finished]

Lab7 Function calling mechanisms (Call by reference).

- a. **Write a C program to perform all operations on strings using functions**
- i. **Find the length of a string**

Program::-

```

#include <stdio.h>
void str(char*);
void main()
{
    char s[20];
    printf("Enter a word\n");
    scanf("%s",s);
    str(s);
}
void str(char *c)
{
    int l=0;
    while(*c!='\0')

```

```

{
    l++;
    c++;
}
printf("The length of string is %d\n",l);
}

```

Output 1:

Enter a word

success

The length of string is 7

[Program finished]

Output 2:

Enter a word

optimistic

The length of string is 10

[Program finished]

ii. Find the substring when position and length is given

Program::-

```

#include<stdio.h>
void str(char*);
void main()
{
    char s[100];
    printf("Enter a string\n");
    scanf("%s",s);
    str(s);
}
void str(char *c)
{
    int p,l,i;
    printf("Enter position and length of substring\n");
    scanf("%d%d",&p,&l);
    printf("\nRequired substring is ");
    for(i=p-1;i<p-1+l;i++)
        printf("%c",c[i]);
}

```

Output 1::-

Enter a string

My name is sunil

Enter position and length of substring

4
4

Required substring is name
[Program finished]

Output 2:
Enter a string
Beautiful
Enter position and length of substring
7
3

Required substring is ful
[Program finished]

Lab8 Recursive functions.

- a. Write C programs that use recursive function
- i. To find the factorial of a given integer.

Program::-

```
#include <stdio.h>
int fact(int);
int main()
{
    int n,f;
    printf("Enter the number\n");
    scanf("%d",&n);
    f=fact(n);
    printf("\nThe factorial of given number is %d\n",f);
}

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

Output 1:
Enter the number
5

The factorial of given number is 120

[Program finished]

Output 2:

Enter the number

8

The factorial of given number is 40320

[Program finished]

ii. **To find the greatest common divisor of two given integers.**

Program:-

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

```
int gcd(int,int);
```

```
int main()
```

```
{
```

```
    int a,b,c;
```

```
    printf("Enter two numbers\n");
```

```
    scanf("%d%d",&a,&b);
```

```
    c=gcd(a,b);
```

```
    printf("\nThe greatest common divisor of two numbers is %d\n",c);
```

```
}
```

```
int gcd(int a,int b)
```

```
{
```

```
    int c;
```

```
    if(b==0)
```

```
        c=a;
```

```
    else
```

```
        c=gcd(b,a%b);
```

```
    return c;
```

```
}
```

Output 1:

Enter two numbers

7

2

The greatest common divisor of two numbers is 1

[Program finished]

Output 2:

Enter two numbers

8

4

The greatest common divisor of two numbers is 4

[Program finished]

iii. To print Fibonacci series.

Program:-

```
#include <stdio.h>
int fib(int);
int main()
{
    int n, i;
    printf("Enter how many numbers\n");
    scanf("%d", &n);
    printf("\nThe fibonacci numbers are\n");
    printf("%d\t%d", 0, 1);

    for (i = 1; i < n-1; i++)
        printf("\t%d", fib(i));
}
int fib(int n)
{
    int c;
    if (n == 0 || n == 1)
        c = 1;
    else
        c = fib(n - 1) + fib(n - 2);
    return c;
}
```

Output 1::

Enter how many numbers

8

The fibonacci numbers are

0 1 1 2 3 5 8 13

[Program finished]

Lab9 Dynamic memory allocation.

a. Write a C program to concatenate two strings using pointers.

Program:-

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

```

int main()
{
    char *c,*s;
    int i,j;
    c=(char*)malloc(10*sizeof(char));
    s=(char*)calloc(1,sizeof(char));
    printf("Enter 1st string\n");
    scanf("%[^\n]",c);
    printf("Enter 2nd string\n");
    scanf("%s",s);
    for(i=0;c[i]!='\0';i++);
    for(j=0;s[j]!='\0';j++,i++)
        c[i]=s[j];
    printf("\nAfter concatenating two strings,\n\n");
    printf("%s",c);
}

```

Output 1:
Enter 1st string
RVRJCCE College
Enter 2nd string
Guntur

After concatenating two strings,

RVRJCCE CollegeGuntur
[Program finished]

Output 2:
Enter 1st string
computer
Enter 2nd string
program

After concatenating two strings,

computer program
[Program finished]

b. Write a C program to find the length of string using pointers.

Program::-
#include <stdio.h>
#include <stdlib.h>
void main()
{
 char *s;

```

s=(char*)malloc(sizeof(char));
printf("Enter string\n");
scanf("%[^\\n]",s);
int l=0;
while(*s!='\\0')
{
    l++;
    s++;
}
printf("The length of string is %d\\n",l);
}

```

Output 1:

```

Enter string
RVRJCCE College,Guntur
The length of string is 22

```

[Program finished]

Output 2;:

```

Enter string
Computer Science
The length of string is 16

```

[Program finished]

c. **Write a C program to compare two strings using pointers.**

Program::-

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    char *c,*s;
    int k=0;
    c=(char*)malloc(10*sizeof(char));
    s=(char*)calloc(1,sizeof(char));
    printf("Enter 1st string\\n");
    scanf("%s",c);
    printf("Enter 2nd string\\n");
    scanf("%s",s);
    while(*c!='\\0' || *s!='\\0')
    {
        if(*c!=*s)
        {
            k=1;
            break;

```

```

        }
        else
        {
            c++;
            s++;
        }
    }

    if(k==0)
        printf("\nStrings are same\n");
    else
        printf("\nstrings are not same\n");
}

```

Output 1:
Enter 1st string
college
Enter 2nd string
college

Strings are same

[Program finished]

Output 2:
Enter 1st string
Mango
Enter 2nd string
Mahi

strings are not same

[Program finished]

d. **Write a C program to copy a string from source to destination using pointers.**

Program::-

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    char *s,*d;
    int i,j;

    s=(char*)malloc(10*sizeof(char));
    d=(char*)calloc(1,sizeof(char));
    printf("Enter string in source\n");
    scanf("%s",s);
}

```

```

        for(i=0,j=0;s[i]!='\0';i++,j++)
            d[j]=s[i];
        printf("source text is %s\n",s);
        printf("destination text is %s\n",d);
    }

```

Output 1:
Enter string in source
Programming for problem solving
source text is Programming for problem solving
destination text is Programming for problem solving

[Program finished]

Lab10 Structures and unions.

a. Create a Book structure containing book_id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details.

Program:-

```

#include <stdio.h>
    struct book
    {
        int id,price;
        char title[20],author[20];
    };
    void printb(struct book a);
    int main()
    {
        struct book a;
        printf("Enter the book id\n");
        scanf("%d",&a.id);
        printf("Enter the price of book\n");
        scanf("%d",&a.price);
        printf("Enter title of book\n");
        scanf("%s",&a.title);
        printf("Enter author name\n");
        scanf("%s",&a.author);

        printb(a);
    }
void printb(struct book b)
{
    printf("\nThe book id number is %d\n",b.id);
    printf("The price of the book is %d\n",b.price);
    printf("The title of the book is %s\n",b.title);
    printf("The author name is %s\n",b.author);
}

```

Output 1:

Enter the book id

12345577

Enter the price of book

1200

Enter title of book

PROGRAMMING

Enter author name

S.KUMAR

The book id number is 12345577

The price of the book is 1200

The title of the book is PROGRAMMING

The author name is S.KUMAR

[Program finished]

b. Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.

Program:-

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

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

```
union adr
```

```
{
```

```
    char name[30],hoa[20],hsa[20],city[15],state[15];int zip;
```

```
};
```

```
int main()
```

```
{
```

```
    union adr a;
```

```
    printf("My address is\n\n");
```

```
    strcpy(a.name,"D.K.Sunil Kumar Reddy");
```

```
    printf("Name: %s\n",a.name);
```

```
    strcpy(a.hoa,"Giddalur");
```

```
    printf("Home address: %s\n",a.hoa);
```

```
    strcpy(a.hsa,"Guntur");
```

```
    printf("Hostel address: %s\n",a.hsa);
```

```
    strcpy(a.city,"Ongole");
```

```
    printf("City: %s\n",a.city);
```

```
    strcpy(a.state,"Andhra Pradesh");
```

```
    printf("State: %s\n",a.state);
```

```
    a.zip=123456;
```

```
    printf("Zipcode: %d\n",a.zip);
```

```
}
```

Output:

My address is

Name: D.K.Sunil Kumar Reddy
Home address: Giddalur
Hostel address: Guntur
City: Ongole
State: Andhra Pradesh
Zipcode: 123456

[Program finished]

c. **Write a C program to define a structure named DOB, which contains name, day, month and year. Using the concept of nested structures display your name and date of birth.**

Program:-

```
#include <stdio.h>
struct name
{
    char name[30];
};
struct DOB
{
    struct name c;
    int date,month,year;
}a;
int main()
{
    printf("Enter your name\n");
    scanf("%s",a.c.name);
    printf("Enter date,month no.,year of your birthday\n");
    scanf("%d%d%d",&a.date,&a.month,&a.year);
    printf("\nName: %s\nDate of birthday: %d/%d/%d\n",a.c.name,a.date,a.month,a.year);
}
```

Output;

Enter your name
santosh
Enter date,month no.,year of your birthday
25
12
2008

Name: santosh
Date of birthday: 25/12/2008

[Program finished]

Lab11 File operations.

a. Write a C program to display the contents of a file.

Program:-

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fp;
    char ch;
    char filename[15];
    printf("Enter the filename to be opened\n");
    scanf("%s",filename);
    fp=fopen(filename,"a");
    if(fp==NULL)
    {
        printf("Unable to open file\n");
        exit(0);
    }
    printf("Enter some text to add\n");
    ch=getchar();
    while(ch!=EOF)
    {
        fputc(ch,fp);
        ch=getchar();
    }
    fclose(fp);
    fp=fopen(filename,"r");
    printf("\nThe contents of file are\n");
    ch=fgetc(fp);
    while(ch!=EOF)
    {
        printf("%c",ch);
        ch=fgetc(fp);
    }
    fclose(fp);
}
```

Output:

Enter the filename to be opened

qw.txt

Enter some text to add

My name is sunil

The contents of file are

My name is sunil

[Program finished]

b. **Write a C program to copy the contents of one file to another.**

Program:-

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char s[15],d[15],ch;
    FILE *fp1,*fp2;
    printf("Enter the source filename\n");
    scanf("%s",s);
    printf("Enter the filename where to be copied\n");
    scanf("%s",d);
    fp1=fopen(s,"r");
    if(fp1==NULL)
    {
        printf("Unable to open file\n");
        exit(0);
    }
    fp2=fopen(d,"a");
    if(fp2==NULL)
    {
        printf("Unable to open file\n");
        exit(0);
    }
    ch=fgetc(fp1);
    while(ch!=EOF)
    {
        fputc(ch,fp2);
        ch=fgetc(fp1);
    }
    fclose(fp1);
    fclose(fp2);
}
```

Output:

Enter the source filename

a.txt

Enter the filename where to be copied

b.txt

[Program finished]

c. Write a C program to reverse the first n characters in a file, where n is given by the user.

Program:-

```
#include <stdio.h>
#include<stdlib.h>
void main()
{
    FILE *fp1,*fp2;
    char ch;
    int n;
    fp1=fopen("a.txt","w+");
    if(fp1==NULL)
    {
        printf("Unable to open file\n");
        exit(0);
    }
    printf("Enter text into file\n");
    ch=getchar();
    while(ch!=EOF)
    {
        fputc(ch,fp1);
        ch=getchar();
    }
    printf("Enter no. of characters to be replaced\n");
    scanf("%d",&n);
    fp2=fopen("r.txt","w");
    fseek(fp1,n+1,0);
    while(fseek(fp1,-2,1)==0)
    {
        ch=fgetc(fp1);
        fputc(ch,fp2);
    }
    fseek(fp1,n,0);
    ch=fgetc(fp1);
    while(ch!=EOF)
    {
        fputc(ch,fp2);
        ch=fgetc(fp1);
    }
    fclose(fp1);
    fclose(fp2);
}
```

Output 1:

Enter text into file
good morning
Enter no. of characters to be replaced
6

[Program finished]

Contents in r.txt when we opened file,
m doogorning

d. Two files DATA1 and DATA2 contain sorted lists of integers. Write a C program to merge the contents of two files into a third file DATA i.e., the contents of the first file followed by those of the second are put in the third file

Program:-

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    FILE *fp,*fp2;
    int i;
    char ch;
    fp2=fopen("DATA.txt","a");
    for(i=1;i<argc;i++)
    {
        fp=fopen(argv[i],"r");
        ch=fgetc(fp);
        while(ch!=EOF)
        {
            fputc(ch,fp2);
            ch=fgetc(fp);
        }
    }
    fclose(fp);
    fclose(fp2);
}
```

Output:

\$/a.out DATA1.txt DATA2.txt

//Open DATA.txt to check whether merged or not.Before writing program, create files DATA1 and DATA2 which contain sorted integers//

e. Write a C program to count the no. of characters present in the file.

```

#include <stdio.h>
#include<stdlib.h>
void main()
{
    char filename[15];
    char ch;
    FILE *fp;
    int c,s;
    printf("Enter an file name\n");
    scanf("%s",filename);
    fp=fopen(filename,"w+");
    printf("Enter some text to file\n");
    ch=getchar();
    while(ch!=EOF)
    {
        fputc(ch,fp);
        ch=getchar();
    }
    fseek(fp,2,0);
    c=0,s=0;
    ch=fgetc(fp);
    while(ch!=EOF)
    {
        c++;
        if(ch==' ')
            s++;
        ch=fgetc(fp);
    }
    fclose(fp);
    printf("\nNo. of characters in given file=%d\n",c);
    printf("No. of characters without including spaces=%d\n",c-s);
}

```

Output:

Enter an file name

c.txt

Enter some text to file

C programming for problem solving

No. of characters in given file=33

No. of characters without including spaces=29

[Program finished]

f. Write a C program to find whether the given word is present in the given file or not. Example: The content of the file is “Computer programming. Computer can do computations”. Input: Computer Output: ‘Computer’ is found at two locations

Program::-

```
#include<stdio.h>
#include<string.h>
int main()
{
    FILE *fp;
    char t[15],w[15],file[15];
    int c=0;
    char ch;
    printf("Enter file name\n");
    scanf("%s",file);
    fp=fopen(file,"w");
    printf("\n Enter the text\n");
    ch=getchar();
    while(ch!=EOF)
    {
        fputc(ch,fp);
        ch=getchar();
    }
    printf("\nEnter the word to be searched\n");
    scanf("%s",w);
    fclose(fp);
    fp=fopen(file,"r");
    while(!feof(fp))
    {
        fscanf(fp,"%s",t);
        if(strcmp(t,w)==0)
            c++;
    }
    fclose(fp);
    if(c==0)
        printf("Given word not found\n");
    else
        printf("\n The word %s is occurred %d times",w,c);
}
```

Output:

Enter file name
e.txt

Enter the text
He studies well. He is topper as well

Enter the word to be searched
well

The word well is occurred 2 times

[Program finished]

Lab12 Command line arguments.

a. Write a C program to print all arguments given through command line.

Program:-

```
#include<stdio.h>
#include<stdlib.h>
int main(int argc,char *argv[])
{
printf("\n Number of arguments supplied=%d\n",argc);
if(argc<1)
{
printf("\n Insufficient number of arguments");
exit(0);
}
printf("\n The arguments supplied are\n");
for(i=0;i<argc;i++)
printf("%s ",argv[i]);
}
```

Output:

```
$cc com.c
$./a.out apple red healthy good
Number of arguments supplied=5
The arguments supplied are
a.out apple red healthy good
```

b. Write a C program to find the sum of N integer numbers using command line arguments.

Program:-

```
#include<stdio.h>
#include<stdlib.h>
main(int argc,char *argv[])
{
int sum,i;
for(i=1,sum=0;i<argc;i++)
sum=sum+atoi(argv[i]); //atoi() function converts a string to number
printf("\n sum=%d",sum);
}
```

Output :

```
$cc sumc.c
$./a.out 12 45 68 45
sum=160
```


