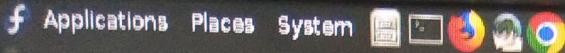


FCS lab assignment:

1. Write a C program to check whether a given year is a leap year or not.
2. Write a C program to check whether a given character is an alphabet or a digit.
3. Write a C program to find the maximum among three integer numbers using conditional statement.
4. Write a C program to find the maximum among three integer numbers using conditional operator.
5. Write a C program to check the grade of the students based on marks.

If marks <50 then Grade is F  
if marks >=50 <60 then Grade is D  
if marks >=60 <70 then Grade is C  
if marks >=70 <80 then Grade is B  
if marks >=80 <90 then Grade is A  
if marks >=90 then Grade is A+



pro1.c (~/soumojit\_ECE\_101) · Pluma

File Edit View Search Tools Documents Help



pro1.c pro2.c pro3.c pro4.c pro5.c pro6.c

```
1 #include<stdio.h>
2 int main(){
3     int year;
4     printf("Enter the year : ");
5     scanf("%d", &year);
6     if ((year%4==0) && (year%100!=0) || (year%400==0))
7         printf("%d is a leap year", year);
8
9     else
10        printf("%d is not a leap year", year);
11
12
13
14    return 0;
15 }
16
17 |
```

I



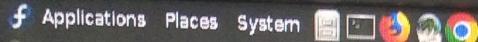
pro2.c (~/soumojit\_ECE\_101) - Pluma

File Edit View Search Tools Documents Help



pro1.c pro2.c pro3.c pro4.c pro5.c pro6.c

```
1 #include<stdio.h>
2 int main(){
3     char ch;
4     printf("Enter number or char : ");
5     scanf("%c", &ch);
6
7     if((ch>= 'a' && ch<='z') || (ch>= 'A' && ch<='Z'))
8
9
10    printf("%c is an alphabet \n", ch);
11 else if((ch>= '0' && ch<='9'))
12
13    printf("%c is a digit \n", ch);
14
15 else
16    printf("Pls enter valid input \n");
17
18
19
20
21 return 0;
```



File Edit View Search Tools Documents Help

pro3.c (~/sourmojit\_ECE\_101) • Pluma



```
1 #include<stdio.h>
2 int main(){
3     int a,b,c,max;
4     printf("Enter 3 numbers : ");
5     scanf("%d %d %d", &a, &b, &c);
6
7     if(a>=b && a>=c)
8         max = a;
9
10    else if(b>=a && b>=c)
11        max = b;
12
13    else if(c>=a && c>=b)
14        max = c;
15
16    else
17        printf("Enter valid input");
18
19
20    printf("Maxima is %d", max);
21
22
23    return 0;
24 }
```

Mate Terminal

sourmojit\_ECE\_101

pro3.c (~/sourmojit\_ECE\_101)

Applications Places System



pro4.c (~/soumojit\_ECE\_101) · Pluma

File Edit View Search Tools Documents Help



pro1.c pro2.c pro3.c pro4.c pro5.c pro6.c

```
1 #include<stdio.h>
2 int main(){
3     int a,b,c,max;
4     printf("Enter 3 numbers : ");
5     scanf("%d %d %d", &a, &b, &c);
6
7     max = a>b ? (a>c ? a:c):(b>c? b:c);
8
9
10    printf("Maxima is %d", max);
11
12
13    return 0;
14 }
```

Applications Places System



pro5.c (~/soumojit\_ECE\_101) · Pluma

File Edit View Search Tools Documents Help



Open



Save



Undo



Redo



Cut



Copy



Paste



Find



Search

pro1.c pro2.c pro3.c pro4.c pro5.c pro6.c

```
1 #include<stdio.h>
2 int main(){
3     int a,b,c,d,e,total;
4     float avg;
5     printf("Enter 5 marks : ");
6     scanf("%d %d %d %d %d", &a,&b,&c,&d,&e);
7     total = a+b+c+d+e;
8     printf("Total marks is %d \n", total);
9     avg = total/5;
10    printf("Average avg is %0.2f \n", avg);
11
12    if(avg>=90)
13        printf("Grade A+");
14    else if(avg>=80 && avg<90)
15        printf("Grade A");
16    else if(avg>=70 && avg<80)
17        printf("Grade B");
18    else if(avg>=60 && avg<70)
19        printf("Grade C");
20    else if(avg>=50 && avg<60)
21        printf("Grade D");
22    else if(avg<50)
23        printf("Grade F");
24
25 }
```

Mate Terminal

soumojit\_ECE\_101

pro5.c (~/soumojit\_ECE\_101)



Applications Places System



pro6.c (~/soumojit\_ECE\_101) · Pluma

File Edit View Search Tools Documents Help



Open



Save



Undo



Redo



Find



Replace



Search

pro1.c pro2.c pro3.c pro4.c pro5.c pro6.c

```
1 #include<stdio.h>
2 int main(){
3     char ch;
4     printf("Enter a char : ");
5     scanf("%c", &ch);
6     switch (ch){
7
8         case 'a':
9         case 'e':
10        case 'i':
11        case 'o':
12        case 'u':
13        case 'A':
14        case 'E':
15        case 'I':
16        case 'O':
17        case 'U':
18            printf("%c is a vowel \n", ch);
19            break;
20        default:
21            printf("%c is a consonant \n", ch);
22        }
23
24     return 0;
25 }
```

Mate Terminal

soumojit\_ECE\_101

pro6.c (~/soumojit\_ECE...

**Lab Assignment**

**Subject:** Programming for problem solving  
**Subject Code:** ES-CS291

**Discipline:** B – Tech (All)  
**Semester:** 2<sup>nd</sup>

**Assignment - 1**

The objective of this assignment is to learn how to write C program using Input/output Function and conditional statements using:

- (a) 'if else' condition
- (b) 'if else if' condition and how to use logical operators.
- (c) 'switch case' statement

**Assignment:**

The objective of this assignment is to learn how to write C program using Input/output Function and conditional statements using:

- a. 'if else' condition
- b. 'if else if' condition and how to use logical operators.
- (c) 'switch case' statement

Assignment:

1. Write a C program to swap two numbers. (with and without third variable)

[Hints: Swapping is used in various programs like sorting the array. It is mainly used in the area when we want to store old values without using much space.]

**Algorithm:****Using Third variable:**

STEP 1: Declare a variable a,b and c as integer;  
 STEP 2: Read two numbers a and b;  
 STEP 3: c=a;  
 STEP 4: a=b;  
 STEP 5: b=a;  
 STEP 6: Print a and b

**Without using third variable:**

STEP 1: START  
 STEP 2: ENTER A, B  
 STEP 3: PRINT A, B  
 STEP 4: A = A + B  
 STEP 5: B= A - B  
 STEP 6: A =A - B  
 STEP 7: PRINT A, B  
 STEP 8:  
 END ]

2. Write a C program to check whether a number is even or odd using if-else statement.

[Hint:

**Algorithm:**

Step 1: Start  
 Step 2: Taking input of a number say n  
 Step 3: Read the number n  
 Step 4: Check if((n%2)==1), then  
     Print n is an odd number  
     Else  
         Print n is an even number  
 Step 5: End

3. Write a C program to check whether a year is a leap year or not.

[Hint:

In the Gregorian calendar, a normal year consists of 365 days. Because the actual length of a sidereal year (the time required for the Earth to revolve once about the Sun) is actually

**Lab Assignment**

**Subject:** Programming for problem solving  
**Subject Code:** ES-CS291

**Discipline:** B – Tech (All)  
**Semester:** 2<sup>nd</sup>

365.2425 days, a "leap year" of 366 days is used once every four years to eliminate the error caused by three normal (but short) years. Any year that is evenly divisible by 4 is a leap year: for example, 1988, 1992, and 1996 are leap years.

However, there is still a small error that must be accounted for. To eliminate this error, the Gregorian calendar stipulates that a year that is evenly divisible by 100 (for example, 1900) is a leap year only if it is also evenly divisible by 400.

To determine whether a year is a leap year, follow these steps:

1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
  2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
  3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
  4. The year is a leap year (it has 366 days).
  5. The year is not a leap year (it has 365 days). ]
4. Write a C program to find all roots of a quadratic equation  $ax^2+bx+c=0$  for all possible combinations of a, b and c. A quadratic equation will have two roots which are obtained using the following expression  $X=(-b \pm \sqrt{b^2-4ac})/2a$  where  $b^2-4ac$  is called discriminant.

Note: When,  $b^2-4ac > 0$  roots are real and unequal.

$b^2-4ac=0$  roots are real and equal i.e. $x=-b/2a$ .

$b^2-4ac < 0$  roots are imaginary i.e. $x=-b/2a \pm (\sqrt{b^2-4ac})/2a \cdot i$ .

[Hint: Quadratic equations are the polynomial equations of degree 2 in one variable of type:  
 $f(x) = ax^2 + bx + c$  where  $a, b, c, \in R$  and  $a \neq 0$ . It is the general form of a quadratic equation where 'a' is called the leading coefficient and 'c' is called the absolute term of  $f(x)$ .

A quadratic equation will always have two roots. The nature of roots may be either real or imaginary.

The roots of a quadratic equation are given by the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Step 1. Start

Step 2. Read the coefficients of the equation, a, b and c from the user.

Step 3. Calculate discriminant =  $(b * b) - (4 * a * c)$

Step 4. If discriminant > 0:

- 4.1: Calculate root1 =  $(-b + \sqrt{\text{discriminant}}) / (2 * a)$
- 4.2: Calculate root2 =  $(-b - \sqrt{\text{discriminant}}) / (2 * a)$
- 4.3: Display "Roots are real and different"
- 4.4: Display root1 and root2 Step

Step 5: Else if discriminant = 0:

- 5.1: Calculate root1 =  $-b / (2 * a)$
- 5.2: root2 = root1
- 5.3: Display "Root are real and equal"
- 5.4: Display root1 and root2

Step 6. Else:

- 6.1: Calculate real =  $-b / (2 * a)$
- 6.2: Calculate imaginary =  $\sqrt{-\text{discriminant}} / (2 * a)$
- 6.3: Display "Roots are imaginary"
- 6.4: Display real, "±", imaginary, "i"

Step 7. Stop ]

## Lab Assignment

**Subject: Programming for problem solving**  
**Subject Code: ES-CS291**

**Discipline: B – Tech (All)**  
**Semester: 2<sup>nd</sup>**

5. Mr. Sayan Ghosh is an employee of a Private Firm. His Basic is Rs. 5500/- . Now the dearness allowance is 74% of his basic salary and house rent allowance is 15% of basic salary. Write a program to calculate his gross salary. [Though his basic salary is given, do this program where basic is taken through keyboard].

[Hint: Gross salary = Basic + DA + HRA; where Basic is given through keyboard.

DA will be calculated from basic. DA= 74% of Basic

HRA= 15% of Basic ]

6. Write a C program to calculate and print electricity bill for consumer @Rs.3.75 per unit, given the following information: previous meter reading and current reading.

[ Hint:

Step 1: Two meter reading will be given through the keyboard.

Step 2: Total consumption of electricity can be calculated by the difference of current and previous electricity reading.

Step 3: To calculate electricity bill amount multiply Calculated difference with 3.75.

Step 4: Print the calculated electricity bill. ]

```
#include<stdio.h>
int main(){
    //Write a C program to swap two numbers. Using Third variable:
    int a,b,c;
    a = 10;
    b = 6;
    printf("Value of a before swap %d \n", a);
    printf("Value of b before swap %d \n", b);

    c = a;
    a = b;
    b = c;

    printf("Value of a after swap %d \n", a);
    printf("Value of b after swap %d \n", b);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    // Write a C program to swap two numbers .Without using third variable.

    int a,b;
    a = 10;
    b = 6;
    printf("Value of a before swap %d \n", a);
    printf("Value of b before swap %d \n", b);

    a = a+b;
    b = a-b;
    a = a-b;

    printf("Value of a after swap %d \n", a);
    printf("Value of b after swap %d \n", b);
    return 0;
}
```

```
#include<stdio.h>
int main(){

    //Write a C program to check whether a number is even or odd using if-else statement.
    int num;
    printf("Enter a number :");
    scanf("%d", &num);

    if((num%2)==1)
        printf("%d is an odd number", num);
    else
        printf("%d is an even number", num);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    //Write a C program to check whether a year is a leap year or not.

    int year;
    printf("Enter the year : ");
    scanf("%d", &year);

    if ((year%4==0) && (year%100!=0) || (year%400==0)){
        printf("%d is leap year \n", year);
    }
    else
        printf("%d is not leap year \n", year);

    return 0;
}
```

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

int main(){
    //Write a C program to find all roots of a quadratic equation
    double discriminant,a,b,c, root1, root2, real, imaginary;
    printf("Enter a b c :");
    scanf("%lf %lf %lf", &a,&b,&c);
    discriminant = ((b*b)-(4*a*c));

    if (discriminant > 0){
        root1 = (( -b + sqrt(discriminant)) / (2*a));
        root2 = (( -b - sqrt(discriminant)) / (2*a));
        printf("Roots are real and different Root 1 is %lf Root 2 is %lf \n", root1, root2);
    }
    else if (discriminant == 0){
        root1 = -b / (2 *a);
        printf("Root are real and equal Root 1 and Root 2 is %lf", root1);
    }
    else{
        real = -b / (2 * a);
        imaginary = sqrt(discriminant) / (2 * a);
        printf("Roots are imaginary %lf, ± ,%lf,i", real,imaginary );
    }
    return 0;
}
```

```
#include<stdio.h>
int main(){
    int basic;
    float da,hra;

    printf("Enter basis salary :");
    scanf("%d", &basic);

    da = (0.74*basic);
    hra = (0.15*basic);

    printf("Gross salary = %0.2f", basic + da + hra);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    // Write a C program to calculate and print electricity bill for consumer

    int current, previous,total;
    float bill;

    printf("Enter current meter readings :");
    scanf("%d", &current);

    printf("Enter previous meter readings :");
    scanf("%d", &previous);

    total = current - previous ;

    bill = 3.75*total;

    printf("Bill is %0.2f \n", bill);

    return 0;
}
```

**Lab Assignment**

**Subject:** Programming for problem solving  
**Subject Code:** ES-CS291

**Discipline:** B – Tech (All)  
**Semester:** 2<sup>nd</sup>

**Assignment - 2**

The objective of this assignment is to learn how to write C program using Input/output Function and conditional statements using:

- (a) 'if else' condition
- (b) 'if else if' condition and how to use logical operators.
- (c) 'switch case' statement

**Assignment:**

1. Write a C program to check whether an alphabet is vowel or consonant using switch case.

[Hint

**Algorithm:**

Step 1: Input an alphabet from user. Store it in some variable say 'ch'

Step 2: Switch the value of 'ch'

Step 3: For 'ch', there are 10 possibilities for vowel we need to check i.e. a, e, i, o, u, A, E, I, O and U.

Step 4: Write all 10 possible cases for vowels and print "Vowel" for each case.

Step 5: If alphabet is not vowel then add a default case and print "Consonant". ]

2. Write a C program to calculate the value of Y using:

$$\begin{aligned}
 Y(x,n) &= \begin{cases} 1+x^2 & \text{when } n=1 \\ 1+x/n & \text{when } n=2 \\ 1+2x & \text{when } n=3 \\ 1+nx & \text{when } n>3 \text{ or } n<1 \end{cases}
 \end{aligned}$$

Where x and n are user inputs. Do this program using if-else-if statement and switch case statement.

3. Write a C Program for following using if-else statements.

An electric distribution companies arranges its domestic consumer as follows:

Consumption in Units Rate of charge

0 - 200 Rs. 0.50 per unit

201 - 400 Rs. 100 plus Rs. 0.65 per unit excess to 200

400 - 600 Rs. 250 plus Rs. 0.80 per unit excess to 400

Above 600 Rs. 425 plus Rs. 1.25 per unit excess to 600

Print the amount to be paid by the consumer.

4. Write a C program to check whether a number is even or odd using switch-case statement

[Hints:

**Algorithm:**

Step 1: Input number from user. Store it in some variable say num.

Step 2: Switch the even number checking expression i.e. switch(num % 2).

Step 3: The expression (num % 2) can have two possible values 0 and 1. Hence write two cases case 0 and case 1.

Step 4: For case 0 i.e. the even case print "Even number".

Step 5: For case 1 i.e. the odd case print "Odd number". ]

## Lab Assignment

**Subject:** Programming for problem solving  
**Subject Code:** ES-CS291

**Discipline:** B – Tech (All)  
**Semester:** 2<sup>nd</sup>

5. Write a C program to find maximum between two numbers using switch case.

[**Hints:**

**Algorithm:**

Step 1: Input two numbers from user. Store it in some variable say num1 and num2.

Step 2: Switch expression switch(num1 > num2).

Step 3: For the expression (num1 > num2), there can be two possible values 0 and 1.

Step 4: Write case 0 and print num2 is maximum.

Step 5: Write case 1 and print num1 is maximum. ]

6. A student's grade is calculated in a subject according to the following rules:

Number Obtained Grade

>=90 and <=100	O
>=80 and <90	E
>=70 and <80	A
>=60 and <70	B
>=50 and <60	C
>=40 and <50	D
<40 and >=0	F(FAILED)
Others No.	INVALID

Write a C Program which accept a student's marks as an input and then determine the grade of the students in that subject. Do this program using 'if-else-if' statement and 'switch-case' statement.

```
#include<stdio.h>
int main(){

    char ch;
    printf("Enter an alphabet : ");
    scanf("%c",&ch);

    switch(ch){
        case 'a':
        case 'A':
        case 'e':
        case 'E':
        case 'i':
        case 'I':
        case 'o':
        case 'O':
        case 'u':
        case 'U':
            printf("%c is vowel \n",ch);
            break;

        default:
            printf("%c consonant \n",ch);
            break;
    }

    return 0;
}
```

```
#include<stdio.h>
int main(){

    int x,n,y;
    printf("Enter x and n : ");
    scanf("%d %d", &x,&n);

    if (n==1)
        y = (1+ (x*x));

    else if (n==2)
        y = (1+ (x/n));

    else if(n==3)
        y = (1 + (2*x));

    else if(n>3 || n<1)
        y = (1 + (n*x));

    printf("value of Y = %d \n",y);

    return 0;
}
```

```
#include<stdio.h>
int main(){

    int x,n,y;
    printf("Enter x and n : ");
    scanf("%d %d", &x,&n);

    switch (n){
        case 1:
            y = (1+ (x*x));
            printf("value of Y = %d \n",y);
            break;

        case 2:
            y = (1+ (x/n));
            printf("value of Y = %d \n",y);
            break;

        case 3:
            y = (1 + (2*x));
            printf("value of Y = %d \n",y);
            break;

        default:
            y = (1 + (n*x));
            printf("value of Y = %d \n",y);
            break;

    }

    return 0;
}
```

```
#include<stdio.h>
int main(){

    float consumption, total_bill, per_unit_factor, extra ;

    printf("Enter electric consumption : ");
    scanf("%f", &consumption);

    if (consumption >= 0 && consumption <=200){
        per_unit_factor = 0.50;

        total_bill = per_unit_factor*consumption;
    }

    else if (consumption >= 201 && consumption <=400){
        per_unit_factor = 0.65;
        extra = 100;

        total_bill = extra + (per_unit_factor*(consumption-200));
    }

    else if (consumption > 400 && consumption <=600){
        per_unit_factor = 0.80;
        extra = 250;

        total_bill = extra + (per_unit_factor*(consumption-400));
    }

    else if (consumption > 600){
        per_unit_factor = 1.25;
        extra = 425;

        total_bill = extra + (per_unit_factor*(consumption-600));
    }

    printf("The amount paid by the consumer = %f ",total_bill);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int num, a;
    printf("Enter a num : ");
    scanf("%d", &num);

    a = num%2;

    switch(a){
        case 1:
            printf("%d is odd number", num);
            break;

        case 0:
            printf("%d is even number", num);
            break;
    }

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int num1, num2;
    printf("enter 2 number : ");
    scanf("%d %d", &num1, &num2);

    switch(num1>num2){
        case 0:
            printf("%d is maximum", num2);
            break;

        case 1:
            printf("%d is maximum", num1);
            break;
    }

    return 0;
}
```

```
#include<stdio.h>
int main(){

    int marks;

    printf("Enter marks :");
    scanf("%d", &marks);

    if(marks >= 90 && marks <= 100)
        printf("Grade O");

    else if(marks >= 80 && marks <90)
        printf("Grade E");

    else if(marks >= 70 && marks <80)
        printf("Grade A");

    else if(marks >= 60 && marks <70)
        printf("Grade B");

    else if(marks >= 50 && marks <60)
        printf("Grade C");

    else if(marks >= 40 && marks <50)
        printf("Grade D");

    else if(marks <40 && marks >=0)
        printf("Grade F(FAILED)");

    else
        printf("INVALID");

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int marks;
    printf("Enter marks :");
    scanf("%d", &marks);

    switch (marks/10){
        case 10:
        case 9:
            printf("Grade 0");
            break;

        case 8:
            printf("Grade E");
            break;

        case 7:
            printf("Grade A");
            break;

        case 6:
            printf("Grade B");
            break;

        case 5:
            printf("Grade C");
            break;

        case 4:
            printf("Grade D");
            break;

        case 3:
        case 2:
        case 1:
        case 0:
            printf("Grade F(FAILED)");
            break;

        default:
            printf("INVALID");
            break;
    }

    return 0;
}
```

}

### **1. Write a C program to check whether a number is palindrome or not**

An integer is a **palindrome** when it reads the same backward as forward. For example, 121 is a palindrome while 123 is not.

Step1: Read the number n and copy the value in another variable

i.e. number=n

Step2: reverse the number n//if n=235 so reverse will be 532

number = n;

reverse=0;

while( n!=0 )

begin

    remainder = n%10;

    reverse = reverse\*10 + remainder;

    n /= 10;

end

Step 3:if (number == reverse) then n is palindrome else it is not palindrome.

### **2. Write a C program to find all factors of a number.**

A number b is factor of a number a if a is divisible by b,i.e. if( a%b==0) then b is the factor of a.

Step1: Read the number n

Step2: For i in 1 to n

    Begin

    If a %i equal to zero then print i is factor of n

    End.

### **3. Write a C program to calculate factorial of a number**

Product of all consecutive Integer numbers up to n is called Factorial of a Number and is denoted by n! For Example, the value of 5! is 120.

Mathematically it is written as,

$n! = 1 * 2 * 3 * 4 * \dots * (n-1) * n$

For example, the factorial of 5 is,

$5! = 1 * 2 * 3 * 4 * 5 = 120$

Step 1: Start

Step 2: Declare Variable n, fact, i

Step 3: Read number from User

Step 4: Initialize Variable fact=1 and i=1

Step 5: while i<=number

    5.1 fact=fact\*i

    5.2 i=i+1

Step 6: Print fact

Step 7: Stop

### **4. Write a C program to find HCF (GCD) of two numbers.**

GCD stands for Greatest Common Divisor. So GCD of 2 numbers is nothing but the largest number that divides both of them.

Example: Lets say 2 numbers are 36 and 60. Then

$$36 = 2 \cdot 2 \cdot 3 \cdot 3$$

$$60 = 2 \cdot 2 \cdot 3 \cdot 5$$

$$\text{GCD} = 2 \cdot 2 \cdot 3$$

i.e GCD=12

GCD is also known as HCF (Highest Common Factor)

Algorithm for Finding GCD of 2 numbers:

Step 1: Start

Step 2: Declare variable n1, n2, gcd=1, i=1

Step 3: Input n1 and n2

Step 4: Repeat until i<=n1 and i<=n2

    Step 4.1: If n1%i==0 && n2%i==0:

        Step 4.2: gcd = i

Step 5: Print gcd

Step 6: Stop

### **5. Write a C program to check whether a number is Prime number or not.**

A number that's only divisible by 1 and itself is named a Prime Number. For Example, 3, 5, 7, 11 are Prime Numbers.

Note: 2 is the only even prime number.

Step 1: Start

Step 2: Initialize variables num,flag=1, j=2

Step 3: Read num from user

Step 4: If num<=1 // Any number less than 1 is not a prime number

    Display "num is not a prime number"

    Goto step 7

Step 5: Repeat the steps until j<[(n/2)+1]

    5.1 If remainder of number divide j equals to 0,

        Set flag=0

        Goto step 6

    5.2 j=j+1

Step 6: If flag==0,

    Display num+" is not prime number"

    Else

        Display num+" n is prime number"

Step 7: Stop

### **6. Write a C program to check whether a number is Armstrong number or not.**

What is an Armstrong number?

An Integer number in which the sum of the power of number of digits is same as the number is called as Armstrong Number

for a three digit number the sum of the cubes of its digits is equal to the number itself For example, 153 is an Armstrong number since  $1^{**}3 + 5^{**}3 + 3^{**}3 = 153$ .

Algorithm to check whether a 3-digit number is Armstrong or not

Step 1: Start

Step 2: Declare Variable sum, temp, num

Step 3: Read num from User

Step 4: Initialize Variable sum=0 and temp=num

Step 5: Repeat Until num>=0

    5.1 sum=sum + cube of last digit i.e  $[(num \% 10) * (num \% 10) * (num \% 10)]$

    5.2 num=num/10

Step 6: IF sum==temp

        Print "Armstrong Number"

    ELSE

        Print "Not Armstrong Number"

Step 7: Stop

## 7. Write a C program to check whether a number is Perfect number or not.

What is Perfect Number?

Perfect number is a positive integer equal to the sum of its proper divisors. Sum of its proper divisor excludes the Number itself. Ex. For number 6, the divisors are 1, 2, 3 and 6. Now if we take sum of 1, 2, 3 and exclude the number itself (i.e. 6), the sum is 6. Hence, 6 is a perfect number. 6 is the smallest Perfect Number.

1. Start

2. Read n

3. Initialize s=0

4. for i=1 to n do

    a. if( $n \% i == 0$ ), then

        b.  $s = s + i$

5. if  $s == n$

        then Print "Given Number is Perfect Number". Goto Step 7

6. Print "Given Number is Not a Perfect Number"

7. Stop

Strong number is a number whose sum of all digits' factorial is equal to the number 'n'.

Factorial implies when we find the product of all the numbers below that number including that number and is denoted by ! (Exclamation sign), For example:  $4! = 4 \times 3 \times 2 \times 1 = 24$ .

So, to find a number whether its strong number, we have to pick every digit of the number like the number is 145 then we have to pick 1, 4 and 5 now we will find factorial of each number i.e,  $1! = 1$ ,  $4! = 24$ ,  $5! = 120$ .

Now we will sum up  $1 + 24 + 120$  so we get 145, that is exactly same as the input given, So we can say that the number is strong number.

START

In Function int factorial(int r)

Step1 -> Initialize int fact and set as 1

Step2-> Loop while  $r > 1$

    Set fact as fact \* r

    Decrement r by 1

End Loop

Step 3-> Return fact

End Function factorial

In Function int check(int n)

Step 1-> Initialize int temp, rem and result, set result as 0

Step 2-> Set temp as n

Step 3-> Loop while temp

    Set rem as temp % 10

    Set result as result + factorial(rem)

    Set temp as temp/10

End loop

Step 4-> If result == n then,

    Return 1

Step 5-> Else

    Return 0

End function check

In main(int argc, char const \*argv[])

Step 1-> Initialise and set n as 145

Step 2->If check(n) is valid then,

    Print "Yes it is a strong number"

Step 3-> Else

    Print "no it is not a strong number"

STOP

## 8. Write a C program to print Fibonacci series up to n terms.

A series of numbers in which each number is the sum of the two preceding or previous numbers is called Fibonacci Series.

For example, Fibonacci series upto 7 numbers is 1, 1, 2, 3, 5, 8, 13.

In above example, first 2 numbers (1, 1) are printed directly as there are no preceding numbers. But after that, i.e the 3rd number (2) is the sum of 1st and 2nd number ( $1+1=2$ ). Similarly to get 4th number, we add 2nd and 3rd number. (i.e.,  $1+2=3$ ). You can use this pattern to find fibonacci series upto any number.

Mathematical expression to find Fibonacci number is :

$F_n = F_{n-1} + F_{n-2}$

i.e. To get nth position number, you should add (n-2) and (n-1) position number.

Step 1: Start

Step 2: Declare variable a, b, c, n, i

Step 3: Initialize variable a=0, b=1 and i=2

Step 4: Read n from user

Step 5: Print a and b

Step 6: Repeat until  $i \leq n$  :

    Step 6.1:  $c = a + b$

    Step 6.2: print c

    Step 6.3:  $a = b$ ,  $b = c$

    Step 6.4:  $i = i + 1$

Step 7: Stop

```
#include<stdio.h>
int main(){

    int n,number,reverse,remainder;
    printf("enter a number : ");
    scanf("%d", &n);

    number = n;
    reverse = 0;
    while( n!=0 ){

        remainder = n%10;
        reverse = reverse*10 + remainder;
        n = n/10;
    }

    if (number == reverse)
        printf("%d is palindrome");

    else
        printf("%d is not palindrome");

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int n,i;
    printf("enter a number : ");
    scanf("%d", &n);

    for (i=1;i<n;i++)
        if(n%i==0)
            printf("%d is factor of %d \n", i,n);

    return 0;
}
```

```
#include<stdio.h>
int main(){

    int n, fact, i;
    printf("enter a number : ");
    scanf("%d", &n);

    fact = 1;
    i = 1;

    while (i<=n){
        fact=fact*i;
        i=i+1;
    }

    printf("factorial of %d is : %d \n", n,fact);

    return 0;
}
```

```
#include<stdio.h>
int main(){

    int num1,num2,gcd=1, i;
    printf("enter number 1 : ");
    scanf("%d", &num1);

    printf("enter number 2 : ");
    scanf("%d", &num2);

    for(i=1;i<=num1 && i<=num2;i++){
        if(num1%i==0 && num2%i==0){
            gcd = i;
        }
    }

    printf("GCD of %d and %d is %d \n",num1,num2,gcd);

    return 0;
}
```

```
#include<stdio.h>
int main(){

    int num,flag=1, j=2;
    printf("enter a number : ");
    scanf("%d", &num);

    if (num<=1)
        printf("%d is not a prime number \n",num);

    for(j=2;j<(num/2+1);j++){
        if(num%j==0)
            flag=0;
    }

    if(flag==0)
        printf("%d is not prime number \n",num);

    else
        printf("%d is prime number \n",num);

    return 0;
}
```

```
#include<stdio.h>

int main(){
    int sum=0,temp,num,a;

    printf("Enter an integer: ");
    scanf("%d", &num);
    temp=num;

    while(num>=0){
        a = num%10;
        sum = (sum + (a*a*a));
        num=num/10;
    }

    if(sum == num)
        printf("%d is an Armstrong number\n", num);

    else
        printf("%d is not an Armstrong number\n", num);

    return 0;
}
```



Code Testing Soumojit - pro7\_1.c

```
1 #include<stdio.h>
2 int main(){
3     int n,i,s=0;
4     printf("Enter number : ");
5     scanf("%d", &n);
6
7     for(i=1;i<n;i++){
8         if(n%i==0)
9             s=s+i;
10
11    }
12    if(s==n)
13        printf("Given Number %d is Perfect Number",n);
14
15    else
16        printf("Given Number %d is not Perfect Number",n);
17
18    return 0;
19
20 }
21
```



Code Testing Soumojit - pro7\_2.c

```
1 #include<stdio.h>
2 int main(){
3     int n,i,fact,rem,sum=0,temp;
4     printf("Enter a number : ");
5     scanf("%d", &n);
6
7     temp =n;
8
9     while(n){
10         i =1,fact =1;
11         rem =n%10;
12         while(i<=rem){
13             fact = fact*i;
14             i++;
15         }
16         sum = sum+fact;
17         n =n/10;
18
19     }
20     if(sum==temp)
21         printf("%d is a strong number",temp);
22
23     else
24         printf("%d is not a strong number",temp);
25     return 0;
26
27 }
28
```



```
1 #include<stdio.h>
2 int main(){
3     int a=0,b=1,c,n,i=2;
4     printf("Enter a number : ");
5     scanf("%d", &n);
6
7     printf("%d, %d, ",a,b);
8
9     for(i=3;i<=n;i++){
10         c =a+b;
11         printf("%d, ", c);
12         a=b;
13         b=c;
14     }
15
16
17     return 0;
18
19 }
20
```

## Assignment-4

1. Write a C program to print the following asterisk graph:

```
*  
* *  
* * *  
* * * * (Up to n numbers of row)
```

Logic:

Step1: Input number of row.

Step2: for i=1 to number of row

- 2.a. for j=1 to i
  - 2.a.1 print “\*”
- 2.b move the control to new line

Step3. Stop.

2. Write a C program to print the following pattern:

```
    1  
   2   3  
  3   4   5  
4   5   6   7 (Up to n)
```

Logic:

Input number of rows

a=1

for i=1 to rows

begin

- //manage the space
- for k= 1 to rows-i
  - Printf(" ")
- For j= 1 to i // used to print the values in the row
  - Begin
  - Print a
  - Increment a

End//end of for j

Print a new line

End//end of for i

3. Write a C program to print digits in following pyramidal form:

```
    1  
      2   3   2  
     3   4   5   4   3  
   4   5   6   7   6   5   4
```

Logic: input number of rows

for (i = 1; i <= rows; ++i)

```

begin
for (space = 1; space <= rows - i; ++space) // mange the space at the beginning of each line
begin
    printf(" ");
    ++count;
End

while (k != 2 * i - 1)
begin
if (count <= rows - 1)
begin
    printf("%d ", i + k);
    ++count;
end
else
begin
    ++count1;
    printf("%d ", (i + k - 2 * count1));
end
++k;
End//end of while k
count1 = count = k = 0;
printf("\n");
end//end of for i

```

4. Write a C program to print the following pascal triangle

1	1	1	1
1	2	3	3
1	3	3	1
<b>1</b> (Up to n numbers of row)			

## Logic:

Input number of rows

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

begin

```

// mange the space at the beginning of line
for (space = 1; space <= rows - i; space++)
printf(" ");
//print the values in each row as coef
for (j = 0; j <= i; j++)
begin
    if (j == 0 || i == 0)
        coef = 1;
    else
        coef = coef * (i - j + 1);
    printf("%4d", coef);
end//end of for j
printf("\n");
end of for i

```



```
1 #include <stdio.h>
2
3 int main()
4 {
5     int rows, i, j, space, coef=1;
6     printf("Enter rows:");
7     scanf("%d", &rows);
8
9     for (i = 0; i < rows; i++)
10    {
11        for (space = 1; space <= rows - i; space++)
12        {
13            printf("  ");
14        }
15        for (j = 0; j <= i; j++)
16        {
17            if (j == 0 || i == 0)
18                coef = 1;
19
20            else
21            {
22                coef = coef * (i - j + 1) / j;
23            }
24            printf("M", coef);
25        }
26        printf("\n");
27    }
28    return 0;
29 }
```



## Code Testing Soumojit - Ass\_4\_Q1.c

```
1 #include<stdio.h>
2
3 int main(){
4     int i,j,row;
5     printf("Enter row:");
6     scanf("%d", &row);
7
8     for ( i = 0; i <=row; i++)
9     {
10         for ( j = 1; j <=i; j++)
11         {
12             printf("* ");
13         }
14         printf("\n");
15     }
16
17     return 0;
18 }
```



## Code Testing Soumojit - Ass\_4\_Q2.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n, i, j, num , k;
6     printf("Enter row:");
7     scanf("%d", &n);
8
9     for (i = 1; i <=n; i++)
10    {
11        for (k = 1; k <= n-i; k++)
12        {
13            printf("      ");
14        }
15        num = i;
16        for (j = 1; j<=i ; j++)
17        {
18            printf("%d      ", num);
19            num = num +1;
20        }
21        printf("\n");
22
23    }
24
25    return 0;
26 }
```



```
1 #include<stdio.h>
2
3 int main(){
4     int rows, i, space, count=0, count1=0, k=0;
5     printf("Enter rows:");
6     scanf("%d", &rows);
7
8     for ( i = 1; i <=rows; i++)
9     {
10         for (space = 1; space <= rows-i; ++space)
11         {
12             printf("      ");
13             ++count;
14         }
15
16         while (k!= 2*i -1)
17         {
18             if(count<=rows-1){
19                 printf("%d    ",i+k);
20                 ++count;
21             }
22             else{
23                 ++count1;
24                 printf("%d    ",(i+k-2*count1));
25
26             }
27             ++k;
28         }
29         count1=count=k=0;
30         printf("\n");
31
32     }
33
34
35     return 0;
36 }
```

Assignment4a - Series:

1. Find the sum  $s=1+2+3+\dots$  up to n terms
2. Find the sum  $s=1+3+5+\dots$  up to n terms
3. Find the sum  $s=2+4+6+\dots$  up to n terms
4. Find the sum  $s=1 + \frac{1}{2} + \frac{1}{3} + \dots$  up to n terms
5. Find the sum  $s=1+ x + x^2 + x^3 + \dots$  up to n terms
6. Find the sum  $s=1 + \frac{1}{2!} + \frac{1}{3!} + \dots$  up to n terms
7. Find the sum  $s=1+ x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$  up to n terms
8. Find the sum  $s=1+ x + \frac{2x}{2!} + \frac{3x}{3!} + \dots$  up to n terms

Practice:

1. Program to convert a decimal number to roman numerals.
2. Program to convert a decimal number to binary number.
3. Program to convert binary number to decimal number.
4. Find the sine series  $s= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$  up to n terms
5. Find the cosine series  $s= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$  up to n terms

```
#include<stdio.h>
int main(){
    int n,i,sum=0;
    printf("Enter the number of terms : ");
    scanf("%d", &n);

    for(i=1;i<=n;i++){
        sum = sum+i;
    }

    printf("Sum of %d terms : %d ",n,sum);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int n,i,sum=0;
    printf("Enter the number of terms : ");
    scanf("%d", &n);

    for(i=0;i<n;i++){
        sum = sum+((2*i)+1);
    }

    printf("Sum of %d terms : %d ",n,sum);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int n,i,sum=0;
    printf("Enter the number of terms : ");
    scanf("%d", &n);

    for(i=1;i<=n;i++){
        sum = sum+(2*i);
    }

    printf("Sum of %d terms : %d ",n,sum);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int n,i;
    float sum = 0;
    printf("Enter the number of terms : ");
    scanf("%d", &n);

    for(i=1;i<=n;i++){
        sum = sum +(1.0/i);
    }

    printf("Sum of %d terms : %f ",n,sum);

    return 0;
}
```

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

int main(){
    int n,i,sum = 1,x;
    printf("Enter the number of terms : ");
    scanf("%d", &n);

    printf("Enter a number X = ");
    scanf("%d", &x);

    for(i=1;i<=n;i++){
        sum = sum + pow(x,i);
    }

    printf("Sum of %d terms : %d ",n,sum);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int n,i,j;
    float sum=0,fact=1;
    printf("Enter the number of terms : ");
    scanf("%d", &n);

    for(i=1;i<=n;i++){
        fact=1;
        for(j=1;j<=i;j++){
            fact = fact*j;
        }
        sum = sum+(1.0/fact);
        printf("1/%d! + ",i);
    }

    printf("= %f ",sum);

    return 0;
}
```

```
#include<stdio.h>
#include<math.h>
int main(){
    int n,i,j,x;
    float p,sum=1,fact=1;
    printf("Enter the number of terms : ");
    scanf("%d", &n);
    printf("Enter a number X = ");
    scanf("%d", &x);

    printf("1+");
    for(i=1;i<=n;i++){
        fact=1;
        p=pow(x,i);
        for(j=1;j<=i;j++){
            fact = fact*j;
        }
        sum = sum+(p/fact);
        printf("x^%d/%d! + ",i,i);
    }

    printf("= %f ",sum);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int n,i,j,x;
    float p,sum=1,fact=1;
    printf("Enter the number of terms : ");
    scanf("%d", &n);

    printf("Enter a number X = ");
    scanf("%d", &x);

    printf("1+");
    for(i=1;i<=n;i++){
        fact=1;
        p=i*x;
        for(j=1;j<=i;j++){
            fact = fact*j;
        }
        sum = sum+(p/fact);
        printf("x^%d/%d! + ",i,i);
    }

    printf("= %f ",sum);

    return 0;
}
```

## **Assignment - 5**

**The objective of this assignment is to learn how to use array in the program**

### **1 - D array**

#### **Assignment:**

1. Write a C program to find the sum of all elements of an array. Also find average.
2. Write a C program to search the any elements from an array using linear search technique.
3. Write a C program to find maximum and minimum element in an array (without sorting).
4. Write a C program to check the frequency of a given element in an array.
5. Write a C program to find the second largest element from an array (without sorting).
6. Write a C program to find the second smallest element from an array (without sorting).
  
7. Write a C program to sort the all elements of an array in ascending order using bubble sort technique.
8. Write a C program to sort the all elements of an array in ascending order using selection sort technique.

#### **Practice:**

1. Write a C program to calculate the sum of all negative and positive elements in an array.
2. Write a C program to find the arithmetic mean, variance and standard deviation of any  $n$  values.

$$\text{Mean } (\mu) = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{Varianc}(\theta) = (\mu) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

$$\text{Standard deviation}(\sigma) = \sqrt{\theta}$$

3. Write a C program to search the any elements from an array using binary search technique.
4. Write a C program to sort the all elements of an array in ascending order using modified bubble sort technique.
5. Write a C program to sort the all elements of an array in ascending order using insertion sort technique.
6. Write a C program to delete all duplicate elements from an array.
7. Write a C program to left rotate and right rotate an array.

Q1. Write a C program to find the sum of all elements of an array. Also find average.

Algo:

```
// Here a is an array

//INPUT

Print "Enter number of elements in array"

Input n

for(index=0; index<n; index++)

begin

    Print "Enter data for a[index]:"

    Input a[index]

end

//sum calculation

sum=0

for(index=0;index<n;index++)

{

    sum=sum+a[index]

}

average=(float)sum / n // Type conversion is needed here

//Output

Print sum, average
```

Q2. Write a C program to search the any elements from an array using linear search technique.

Algo:

//Input

Input number of elements n and array a

Print "Enter element/item to be searched:"

Input item

Indexpos=-1;

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

begin

//compare the item with each element

if(a[i]==item)

{

    Printf "\nItem", item, "Found at position" , i+1

    Indexpos=i;

}

end of for

//Item Not present

if(Indexpos== -1)

    print "Item not present in array"

Q3. Write a C program to find maximum and minimum element in an array (without sorting).

Algo:

**Algo to find out minimum**

```
Input number of elements n and array a
min=a[0]
minPos=0 // Index position of minimum
for(i=1;i<n;i++)
begin

    //Update minimum
    if(a[i]<min){
        min=a[i]
        minPos=i
    }
End of for

//Output
Print min , minPos      // Print minimum and Index position of minimum

## Try finding out maximum by yourself ##
```

Q4. Write a C program to check the frequency of a given element in an array.

Algo:

```
// Input  
Input number of elements n and array a  
Print "Enter the element to be count:"  
Input item  
  
//Frequency count  
count=0  
for(i=0;i<n;i++)  
begin  
//compare the item with each element  
if(a[i]==item){  
    count = count + 1  
}  
End of for  
  
if(count==0) //Item Not present  
print "\nItem not present in array"  
else  
print "\nFrequency of item is ", count
```

Q5. Write a C program to find the second largest and 2<sup>nd</sup> smallest element from an array (without sorting).

```
// max1 stores 1st maximum, max2 stores 2nd maximum
```

```
// min1 stores 1st minimum, min2 stores 2nd minimum
```

```
// Input
```

```
Input number of elements n and array a
```

```
//Initialization
```

```
if(a[0]>a[1]) {
```

```
    max1=min2=a[0]
```

```
    max2=min1=a[1]
```

```
}
```

```
else {
```

```
    max1=min2=a[1]
```

```
    max2=min1=a[0]
```

```
}
```

```
// Finding out Largest, 2nd largest and smallest , 2nd smallest
```

```
for(i=2;i<n;i++)
```

```
begin
```

```
    //Largest and 2nd Largest
```

```
    if(a[i]>max1){
```

```
        max2=max1
```

```
        max1=a[i]
```

```
}
```

```
else if(a[i] > max2 && a[i] < max1)
    max2 = a[i]
```

```
//Smallest and 2nd smallest
if(a[i]<min1){
    min2=min1
    min1=a[i]
}
else if(a[i] < min2 && a[i] > min1)
{
    min2 = a[i]
}
```

End of for

```
//OUTPUT
Print max1,max2
Print min1,min2
```

Q6. Write a C program to sort the all elements of an array in ascending order using bubble sort technique.

Algo:

```
for(i=0;i<n-1;i++)  
begin  
    for(j=0;j<n-(i+1);j++)  
        begin  
            if(a[j]>a[j+1])  
            {  
                //swap a[j] with a[j+1]  
                temp=a[j]  
                a[j]=a[j+1]  
                a[j+1]=temp  
            }  
        End of for  
    End of for
```

```
//Output  
Print "After Ascending sort: "  
for(i=0;i<n;i++)  
    print a[i]
```

Q7. Write a C program to sort the all elements of an array in ascending order using selection sort technique.

Algo

for( $i=0; i \leq n-2; i++$ ) // ith pass

Begin

    minIndex= $i$

    for( $j=i+1; j < n; j++$ )

        Begin

            //finding index of minimum

            if( $a[minIndex] > a[j]$ )

                minIndex= $j$

    End of for

    //swap  $a[i]$  with  $a[minIndex]$

$t=a[i]$

$a[i]=a[minIndex]$

$a[minIndex]=t$

End of for

//Output

Print "After Ascending sort: "

for( $i=0; i < n; i++$ )  
    print  $a[i]$

```
#include<stdio.h>
int main(){
    // Write a C program to find the sum of all elements of an array. Also find average.
    int n,index,sum=0;
    float average;
    printf("Enter number of elements in array : ");
    scanf("%d",&n);
    int arr[n];

    for(index=0; index<n; index++){
        printf("Enter data for index %d: ",index);
        scanf("%d",&arr[index]);
    }

    for(index=0;index<n;index++)
    {
        sum=sum +arr[index];
    }
    average=(float)sum/n;

    printf("Sum = %d and Average = %0.2f\n",sum,average);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    // Write a C program to search the any elements from an array using linear search technique.
    int n,i,index,element,Indexpos=0;

    printf("Enter number of elements in array : ");
    scanf("%d",&n);
    int arr[n];

    for(index=0; index<n; index++){
        printf("Enter data for index %d: ",index);
        scanf("%d",&arr[index]);
    }

    printf("Enter element to be searched : ");
    scanf("%d",&element);

    for(i=0; i<n; i++){
        if(arr[i]==element){
            printf("%d Found at index %d\n",element, i);
            Indexpos=1;
        }
    }
    if(Indexpos==0)
        printf("Item not present in array");

    return 0;
}
```

```
#include<stdio.h>
int main(){
    // Write a C program to find maximum and minimum element in an array (without sorting).
    int n,i,max=0,index_max=0,index_min=0,status_max=0,status_min=0,min;
    printf("Enter number of elements in array : ");
    scanf("%d",&n);
    int arr[n];

    for(i=0; i<n; i++){
        printf("Enter data for index %d: ",i);
        scanf("%d",&arr[i]);
    }

    for(i=0; i<n; i++){
        if(arr[i]>max){
            max = arr[i];
            index_max = i;
            status_max=1;
        }
    }

    for(i=0; i<n; i++){
        if(arr[i]<min){
            min = arr[i];
            index_min = i;
            status_min=1;
        }
    }

    if(status_max==0)
        printf("Maxima not found");

    else
        printf("Maxima = %d at index %d\n",max,index_max);

    if(status_min==0)
        printf("Minima not found");

    else
        printf("Minima = %d at index %d\n",min,index_min);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    // Write a C program to check the frequency of a given element in an array
    int n,index,count=0,item,i;

    printf("Enter number of elements in array : ");
    scanf("%d",&n);
    int arr[n];

    for(index=0; index<n; index++){
        printf("Enter data for index %d: ",index);
        scanf("%d",&arr[index]);
    }

    printf("Enter the element to be count: ");
    scanf("%d",&item);

    for(i=0;i<n;i++){
        if(arr[i]==item){
            count = count + 1;
        }
    }
    if(count==0)
        printf("Item not present in array");
    else
        printf("Frequency of item %d is %d times\n",item, count);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    // Write a C program to find the second largest element from an array (without sorting).
    int n,index,max1,max2,min1,min2,i;

    printf("Enter number of elements in array : ");
    scanf("%d",&n);
    int arr[n];

    for(index=0; index<n; index++){
        printf("Enter data for index %d: ",index);
        scanf("%d",&arr[index]);
    }
    if(arr[0]>arr[1]){
        max1=min2=arr[0];
        max2=min1=arr[1];
    }
    else{
        max1=min2=arr[1];
        max2=min1=arr[0];
    }

    for(i=2;i<n;i++){
        //Largest and 2nd Largest
        if(arr[i]>max1){
            max2=max1;
            max1=arr[i];
        }
        else if(arr[i] > max2 && arr[i] < max1)
            max2 = arr[i];

        //Smallest and 2nd smallest
        if(arr[i]<min1){
            min2=min1;
            min1=arr[i];
        }
        else if(arr[i] < min2 && arr[i] > min1)
        {
            min2 = arr[i];
        }
    }

    printf("max1 = %d and max2 = %d\n",max1,max2);
    printf("min1 = %d and min2 = %d\n",min1,min2);

    return 0;
}
```

```
#include<stdio.h>
int main(){
    // Bubble sort
    int i,n,j,temp;
    printf("Enter the size of array: ");
    scanf("%d", &n);
    int arr[n];
    for(i=0;i<n;i++){
        printf("Enter element at index %d : ",i);
        scanf("%d",&arr[i]);
    }
    for(i=0;i<n-1;i++){
        for(j=0;j<n-i-1;j++){
            if(arr[j]>arr[j+1]){
                temp = arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
    printf("After sort : ");
    for(i=0;i<n;i++)
        printf("%d, ",arr[i]);
    return 0;
}
```

```
#include<stdio.h>
int main(){
    // selection sort
    int i,n,j,temp;
    printf("Enter the size of array: ");
    scanf("%d", &n);
    int arr[n];
    for(i=0;i<n;i++){
        printf("Enter element at index %d : ",i);
        scanf("%d",&arr[i]);
    }
    for(i=0;i<n-1;i++){
        int min_index = i;
        for(j=i+1;j<n;j++){
            if(arr[j]<arr[min_index])
                min_index = j;
        }
        if(min_index != i){
            temp = arr[min_index];
            arr[min_index]=arr[i];
            arr[i]=temp;
        }
    }
    printf("After sort : ");
    for(i=0;i<n;i++)
        printf("%d, ",arr[i]);
}

return 0;
}
```

## **Assignment - 6(Continue Assignment - 5)**

### **Assignment:**

1. Write a C program to add two matrices of order  $M \times N$

Algorithm:

a[10][10], b[10][10], sum[10][10] are arrays

print"Enter the number of rows (between 1 and 10): "

Input r

print"Enter the number of columns (between 1 and 10): "

Input c

print"\nEnter elements of 1st matrix:\n"

for (i = 0; i < r; ++i)

    for (j = 0; j < c; ++j) {

        print"Enter element a%d%d: ", i + 1, j + 1

        Input a[i][j]

}

print"Enter elements of 2nd matrix:\n"

for (i = 0; i < r; ++i)

    for (j = 0; j < c; ++j) {

        print "Enter element a ", i + 1, j + 1

        Input b[i][j]

}

// adding two matrices

for (i = 0; i < r; ++i)

    for (j = 0; j < c; ++j) {

        sum[i][j] = a[i][j] + b[i][j];

}

```

// printing the result
print"\nSum of two matrices: \n"
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        print ( sum[i][j])
    }
Print "\n"
}

```

2. Write a C program to multiply two matrices.

Algorithm:

a[10][10], b[10][10], multiply[10][10] are arrays

```

print "Enter number of rows and columns of a matrix"
Input m, n
print"Enter elements of a matrix\n"

for (i = 0; i < m; i++)
    for (j = 0; j < n; j++)
        Input a[i][j]

print"Enter number of rows and columns of b matrix\n"
Input p, q

if (n != p)

```

```
print "The multiplication isn't possible.\n"
else
{
    Print "Enter elements of b matrix\n"
    for (i = 0; i < p; i++)
        for (j = 0; j < q; j++)
            Input b[i][j]

//Multiplication
print"Product of the matrices:\n"
for (i = 0; i < m ; i++) {
    for (j = 0; j < q ; j++) {
        sum = 0
        for (k = 0; k < n ; k++) { // p=n
            sum = sum + a[i][k]*b[k][j]
        }
        multiply[i][j] = sum
        print multiply[i][j]
    }
    print"\n"
}

}// end of else
```

3. Write a C program to find transpose of a matrix.

Algorithm:

```
int a[10][10], transpose[10][10] are arrays
```

```
print "Enter rows and columns: "
```

```
Input r, c
```

```
// Assigning elements to the matrix
```

```
Print "Enter matrix elements:"
```

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

```
    for (j = 0; j < c; ++j) {
```

```
        Input a[i][j]
```

```
}
```

```
// Displaying the matrix a[][]
```

```
Print "Entered matrix: "
```

```
for (i = 0; i < r; ++i){
```

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

```
        print a[i][j]
```

```
    print"\n"
```

```
}
```

```
// Finding the transpose of matrix a
```

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

```

for (j = 0; j < c; ++j) {
    transpose[j][i] = a[i][j]
}

// Displaying the transpose of matrix a
Print "\nTranspose of the matrix:\n"
for (i = 0; i < c; ++i){
    for (j = 0; j < r; ++j) {
        print transpose[i][j]
    }
    Print "\n"
}

```

### **Practice:**

1. Write a C program to subtract two matrices.
2. Write a C program to perform Scalar matrix multiplication.
3. Write a C program to find sum of main diagonal and minor diagonal elements of a matrix.
4. Write a C program to find sum of each row and column of a matrix.
5. Write a C program to find upper triangular matrix and lower triangular matrix.
6. Write a C program to check Identity matrix.
7. Write a C program to check Symmetric matrix.
8. Write a C program to check Sparse matrix.

```
#include<stdio.h>
int main(){

    int i,j,r,c;
    printf("Enter number of rows : ");
    scanf("%d", &r);

    printf("Enter number of columns : ");
    scanf("%d", &c);

    int matrix_1[r][c],matrix_2[r][c],sum[r][c];

    printf("Enter elements of 1st matrix: \n");
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            printf("Enter element for Row: %d and Col: %d = ",i+1,j+1);
            scanf("%d", &matrix_1[i][j]);
        }
    }

    printf("Enter elements of 2nd matrix: \n");
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            printf("Enter element for Row: %d and Col: %d = ",i+1,j+1);
            scanf("%d", &matrix_2[i][j]);
        }
    }

    for(i=0;i<r;i++){
        for(j=0;j<c;j++)
            sum[i][j] = matrix_1[i][j] + matrix_2[i][j];
    }

    printf("\n Sum of 2 matrices : \n\n");
    for(i=0;i<r;i++){
        for(j=0;j<c;j++)
            printf("\t%d",sum[i][j]);
        printf("\n\n");
    }

    return 0;
}
```

```
#include<stdio.h>
int main(){

    int m,n,i,j,p,q;

    printf("Enter rows and columns of A matrix : ");
    scanf("%d %d",&m,&n);
    int a[m][n];

    printf("Enter elements for matrix A\n");
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            printf("Enter element for row %d and column %d : ",i+1,j+1);
            scanf("%d", &a[i][j]);
        }
    }

    printf("Enter rows and columns of B matrix : ");
    scanf("%d %d",&p,&q);
    int b[m][n];

    if(n!=p)
        printf("Multiplication is not possible");
    else{
        printf("Enter elements for matrix B\n");
        for(i=0;i<p;i++){
            for(j=0;j<q;j++){
                printf("Enter element for row %d and column %d : ",i+1,j+1);
                scanf("%d", &b[i][j]);
            }
        }

        // Multiplication
        int multiply[m][n],sum,k;
        printf("The product of matrices: \n");
        for(i=0;i<m;i++){
            for(j=0;j<q;j++){
                sum=0;
                for(k=0;k<n;k++){
                    sum = sum+a[i][k]*b[k][j];
                }
                multiply[i][j]=sum;
                printf("\t%d\t",multiply[i][j]);
            }
            printf("\n\n");
        }
    }

    return 0;
}
```

```
#include<stdio.h>
int main(){

    int r,c,i,j;

    printf("Enter rows and columns : ");
    scanf("%d %d",&r,&c);
    int a[r][c];

    printf("Enter elements\n");
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            printf("Enter element for row %d and column %d : ",i+1,j+1);
            scanf("%d", &a[i][j]);
        }
    }
    // Displaying the matrix

    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            printf("\t%d\t",a[i][j]);
        }
        printf("\n\n");
    }

    // Finding transpose
    int transpose[c][r];
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            transpose[j][i]=a[i][j];
        }
    }

    // Displaying the transpose matrix
    printf("\nTranspose of the matrix : \n");
    for(i=0;i<c;i++){
        for(j=0;j<r;j++){
            printf("\t%d\t",transpose[i][j]);
        }
        printf("\n\n");
    }

    return 0;
}
```

```
1 #include<stdio.h>
2 int main(){
3
4     int i,j,r,c;
5     printf("Enter number of rows : ");
6     scanf("%d", &r);
7
8     printf("Enter number of columns : ");
9     scanf("%d", &c);
10
11    int matrix_1[r][c],matrix_2[r][c],subtract[r][c];
12
13    printf("Enter elements of 1st matrix: \n");
14    for(i=0;i<r;i++){
15        for(j=0;j<c;j++){
16            printf("Enter element for Row: %d and Col: %d = ",i+1,j+1);
17            scanf("%d", &matrix_1[i][j]);
18        }
19    }
20
21    printf("Enter elements of 2nd matrix: \n");
22    for(i=0;i<r;i++){
23        for(j=0;j<c;j++){
24            printf("Enter element for Row: %d and Col: %d = ",i+1,j+1);
25            scanf("%d", &matrix_2[i][j]);
26        }
27    }
28
29
30    for(i=0;i<r;i++){
31        for(j=0;j<c;j++)
32            subtract[i][j] = matrix_1[i][j] - matrix_2[i][j];
33    }
34    printf("\n Subtract of 2 matrices : \n\n");
35    for(i=0;i<r;i++){
36        for(j=0;j<c;j++)
37            printf("\t%d",subtract[i][j]);
38
39        printf("\n\n");
40    }
41
42    return 0;
43
44 }
```

```
1 #include<stdio.h>
2 int main(){
3
4     int i,j,r,c;
5     printf("Enter number of rows : ");
6     scanf("%d", &r);
7
8     printf("Enter number of columns : ");
9     scanf("%d", &c);
10
11    int matrix[r][c],num;
12
13    printf("Enter elements of matrix: \n");
14    for(i=0;i<r;i++){
15        for(j=0;j<c;j++){
16            printf("Enter element for Row: %d and Col: %d = ",i+1,j+1);
17            scanf("%d", &matrix[i][j]);
18        }
19    }
20    printf("Enter number : ");
21    scanf("%d",&num);
22
23    for(i=0;i<r;i++){
24        for(j=0;j<c;j++)
25            matrix[i][j] = num*matrix[i][j];
26    }
27    printf("\n Matrix is: \n\n");
28    for(i=0;i<r;i++){
29        for(j=0;j<c;j++)
30            printf("\t%d",matrix[i][j]);
31
32        printf("\n\n");
33    }
34
35    return 0;
36
37 }
```

```
1 #include<stdio.h>
2 int main(){
3
4     int i,j,r,c;
5     printf("Enter number of rows : ");
6     scanf("%d", &r);
7
8     printf("Enter number of columns : ");
9     scanf("%d", &c);
10
11    int matrix[r][c],sum_main=0,sum_minor=0;
12
13    printf("Enter elements of matrix: \n");
14    for(i=0;i<r;i++){
15        for(j=0;j<c;j++){
16            printf("Enter element for Row: %d and Col: %d = ",i+1,j+1);
17            scanf("%d", &matrix[i][j]);
18        }
19    }
20
21    for(i=0;i<r;i++){
22        sum_main += matrix[i][i];
23        sum_minor += matrix[i][c-1-i];
24    }
25
26    printf("\n Matrix is: \n\n");
27    for(i=0;i<r;i++){
28        for(j=0;j<c;j++)
29            printf("\t%d",matrix[i][j]);
30
31        printf("\n\n");
32    }
33
34    printf("\n The sum of main diagonal is : %d \n\n",sum_main);
35    printf("\n The sum of minor diagonal is : %d \n\n",sum_minor);
36
37
38    return 0;
39
40 }
```

```
1 #include<stdio.h>
2 int main(){
3
4     int i,j,r,c;
5     printf("Enter number of rows : ");
6     scanf("%d", &r);
7
8     printf("Enter number of columns : ");
9     scanf("%d", &c);
10
11    int matrix[r][c],sum_row=0,sum_column=0;
12
13    printf("Enter elements of matrix: \n");
14    for(i=0;i<r;i++){
15        for(j=0;j<c;j++){
16            printf("Enter element for Row: %d and Col: %d = ",i+1,j+1);
17            scanf("%d", &matrix[i][j]);
18        }
19    }
20
21
22    printf("\n Matrix is: \n\n");
23    for(i=0;i<r;i++){
24        for(j=0;j<c;j++)
25            printf("\t%d",matrix[i][j]);
26
27        printf("\n\n");
28    }
29
30
31    for(i=0;i<r;i++){
32        for(j=0;j<c;j++){
33            sum_row += matrix[i][j];
34        }
35        printf("Sum of Row %d is : %d\n",i+1,sum_row);
36        sum_row=0;
37    }
38    printf("\n\n");
39
40    for(j=0;j<c;j++){
41        for(i=0;i<r;i++){
42            sum_column += matrix[i][j];
43        }
44        printf("Sum of Column %d is : %d\n",j+1,sum_column);
45        sum_column=0;
46    }
47
48
49    return 0;
50
51 }
```

```
1 #include<stdio.h>
2 int main(){
3
4     int i,j,r,c;
5     printf("Enter number of rows : ");
6     scanf("%d", &r);
7
8     printf("Enter number of columns : ");
9     scanf("%d", &c);
10
11    int matrix[r][c],sum_row=0,sum_column=0;
12
13    printf("Enter elements of matrix: \n");
14    for(i=0;i<r;i++){
15        for(j=0;j<c;j++){
16            printf("Enter element for Row: %d and Col: %d = ",i+1,j+1);
17            scanf("%d", &matrix[i][j]);
18        }
19    }
20
21
22    printf("\n Matrix is: \n\n");
23    for(i=0;i<r;i++){
24        for(j=0;j<c;j++)
25            printf("\t%d",matrix[i][j]);
26
27        printf("\n\n");
28    }
29
30
31    printf("\n The Lower Triangular Matrix is: \n\n");
32
33    for(i=0;i<r;i++){
34        for(j=0;j<c;j++){
35            if(i>=j){
36                printf("\t%d",matrix[i][j]);
37            }
38            else{
39                printf("\t0");
40            }
41        }
42        printf("\n\n");
43    }
44
45    printf("\n\n");
46
47    printf("\n The Upper Triangular Matrix is: \n\n");
48
49    for(i=0;i<r;i++){
50        for(j=0;j<c;j++){
51            if(i>j){
52                printf("\t0");
53
54            }
55            else{
56                printf("\t%d",matrix[i][j]);
57            }
58        }
59        printf("\n\n");
60    }
61    printf("\n\n");
62
63    }
64    printf("\n\n");
65
66
67    return 0;
```

```
68  
69  }
```

```
1 #include<stdio.h>
2 int main(){
3
4     int i,j,r,c,flag=1;
5     printf("Enter number of rows : ");
6     scanf("%d", &r);
7
8     printf("Enter number of columns : ");
9     scanf("%d", &c);
10
11    int matrix[r][c],sum_row=0,sum_column=0;
12
13    printf("Enter elements of matrix: \n");
14    for(i=0;i<r;i++){
15        for(j=0;j<c;j++){
16            printf("Enter element for Row: %d and Col: %d = ",i+1,j+1);
17            scanf("%d", &matrix[i][j]);
18        }
19    }
20
21    printf("\n Matrix is: \n\n");
22    for(i=0;i<r;i++){
23        for(j=0;j<c;j++)
24            printf("\t%d",matrix[i][j]);
25
26        printf("\n\n");
27    }
28
29    for(i=0;i<r;i++){
30        for(j=0;j<c;j++){
31            if(matrix[i][j]!=1 && matrix[j][i] !=0){
32                flag=0;
33                break;
34            }
35        }
36    }
37    if(flag==1)
38        printf("Identity Matrix\n");
39
40    else
41        printf("Not an Identity Matrix\n");
42
43
44    return 0;
45
46 }
```

**Lab Assignment**

**Subject:** Programming for problem solving  
**Subject Code:** ES-CS291

**Discipline:** B – Tech (All)  
**Semester:** 2<sup>nd</sup>

**Assignment – 7**

**The objective of this assignment is to learn how to use string in the program**

[Do the all programs using library function and without using library function]

1. Write a C program to find the length of a string.

[Hint: Use **strlen()** under <string.h> header file]

**Algorithm without library function:**

STEP 1: Take input in character array

STEP 2: Using a for loop, count the number of characters in the array from 0th position, until a null character ('\0') is found.

2. Write a C program to count the total number of vowels and consonants in a string.

**Algorithm:**

STEP 1: Take input in character array

STEP 2: Initialize two variables: vowel=0 & consonant=0

STEP 3: Find the length of the string

STEP 4: Run a loop from start till end of string

STEP 5: Check if the current character is a vowel, increment vowel variable by one, else increment consonant variable by one

3. Write a C program to concatenate two strings.

[Hint: Use **strcat()** under <string.h> header file]

**Algorithm without library function:**

STEP 1: Take 2 strings in 2 character arrays as input.

STEP 2: Find length of 1st string and store it in a variable say j.

STEP 3: Run a loop from 0 to the last character of the 2nd string.

[for(i=0;s2[i]!='\0';i++)]

STEP 4: Append the characters of the 2nd string at s1[j+i] position of the 1st string until there is no character available in the 2nd string. So 2nd string is added at the end of the 1st string.

STEP 5: Print the 1st string

4. Write a C program to find the reverse of a string.

[Hint: Use **strrev()** under <string.h> header file]

**Algorithm without library function:**

STEP 1: Take input in character array

STEP 2: Find the length of the string

STEP 3: Initialize a variable j with 1 less than the length of the input string. [As in the reverse array, we will store the value from the 0th position.]

STEP 4: Run a loop from 0 to less than the length of input string. [for(i=0; i<count; i++)]

STEP 5: At ith position of the reverse array, store the jth positional value of the input string.

STEP 6: Decrement j by 1 inside the loop. [will point to the next element of input string.]

STEP 7: Print the reverse array.

**Lab Assignment**

**Subject:** Programming for problem solving  
**Subject Code:** ES-CS291

**Discipline:** B – Tech (All)  
**Semester:** 2<sup>nd</sup>

5. Write a C program to remove all extra blank spaces from a given string.

STEP 1: Input a string with space.

STEP 2: Find the length of the string

STEP 3: Start a loop [i] from 0 to length of string

STEP 4: Check if the current character is a space or not using [if str[i]== ‘ ’]

STEP 5: If it’s a space, start another loop [j] from i to length of string

STEP 6: Replace space with next element [str[j] = str[j+1] ]

STEP 7: Decrement the length by 1 inside if statement

STEP 8: Print the string without space.

6. Write a C program to count the frequency of each character in a string.

STEP 1: Input a string with space.

STEP 2: Find the length of the string

STEP 3: Start a for loop [i] from 0 to length of string

STEP 4: Initialize a counter=1.

STEP 5: If string is not null start a loop [j] from i+1 to length of string

STEP 6: If(str[i] == str[j]) increment the counter, and set the current position of j with null. [s[j]=’\0’]

[If the frequency of a character is already counted, then it will be replaced with null character to remove duplicity of character.]

STEP 7: Print the character and frequency of that character outside the inner for loop [j], inside the if statement.

7. Write a C program to convert lowercase string to uppercase string and vice versa.

**[Hint:** Use **toupper()** and **tolower()** under <string.h> header file]

8. Write a C program to compare two strings.

**[Hint:** Use **strcmp()** under <string.h> header file]

9. Write a C program to check whether a string is palindrome or not (using single string only).

10. Write a C program to find the total number of alphabets, digits or special characters in a string.

```
#include<stdio.h>

int main(){
    //1. Write a C program to find the length of a string.
    char ch[30];
    int len =0,i=0;

    printf("Enter string : ");
    scanf("%s",ch);

    while(ch[i]!='\0'){
        len++;
        i++;
    }

    printf("Length is %d\n",len);

    return 0;
}
```

```
#include<stdio.h>
#include<string.h>
int main(){
    //1. Write a C program to find the length of a string.using gets
    char ch[30];
    int len =0,i=0;

    printf("Enter string : ");
    gets(ch);

    while(ch[i]!='\0'){
        len++;
        i++;
    }

    printf("Length is %d\n",len);

    return 0;
}
```

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

int main(){
    // 2. Write a C program to count the total number of vowels and consonants in a string.
    char ch[30];
    int len =0,i=0,vowel=0,consonant=0;

    printf("Enter string : ");
    gets(ch);

    while(ch[i]!='\0'){
        len++;
        if(ch[i]=='A' || ch[i]=='E' || ch[i]=='I' || ch[i]=='O' || ch[i]=='U' || ch[i]=='a' || ch[i]=='e' || ch[i]=='i' || ch[i]=='o' || ch[i]=='u')
            vowel++;
        else if(ch[i]!=' ')
            consonant++;

        i++;
    }

    printf("Length is %d\n",len);
    printf("The total number of vowels = %d\n",vowel);
    printf("The total number of consonants = %d\n",consonant);

    return 0;
}
```

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

int main(){
    // 3. Write a C program to concatenate two strings.
    char ch1[30],ch2[30];
    int len =0,i=0,j=0;

    printf("Enter string 1 : ");
    gets(ch1);

    printf("Enter string 2 : ");
    gets(ch2);

    while(ch1[i]!='\0'){
        len++;
        i++;
    }

    while(ch2[j]!='\0'){
        ch1[i]=ch2[j];
        i++;
        j++;
    }

    printf("New string is : ");
    puts(ch1);

    return 0;
}
```

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

int main(){
    // 4. Write a C program to find the reverse of a string.
    char ch[30],new[30];
    int len =0,i=0,j=0;

    printf("Enter string : ");
    gets(ch);

    while(ch[i]!='\0'){
        len++;
        i++;
    }
    i--;
    while(i>=0){
        new[j]=ch[i];
        j++;
        i--;
    }

    new[j] = '\0';
    printf("Reverse array : ");
    puts(new);

    return 0;
}
```

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

int main(){
    // 5. Write a C program to remove all extra blank spaces from a given string.
    char ch[30];
    int len =0,i=0,j=0;

    printf("Enter string : ");
    gets(ch);

    while(ch[i]!='\0'){
        if(ch[i]==' '){
            j = i;
            while(ch[j]!='\0'){
                ch[j]=ch[j+1];
                j++;
            }
        }
        len++;
        i++;
    }

    printf("New string is : ");
    puts(ch);

    return 0;
}
```



Day 12 20\_4\_2023\_Soumojit\_Shome - ass7\_pro6\_2.c

```
1 #include <stdio.h>
2
3 // 6. Write a C program to count the frequency of each character in a string.
4
5 int main() {
6     char str[100];
7     int i, j, len, counter;
8
9     printf("Enter a string: ");
10    gets(str);
11
12    len = 0;
13    while(str[len] != '\0') {
14        len++;
15    }
16
17    for(i=0; i<len; i++) {
18        if(str[i] != '\0') {
19            counter = 1;
20            for(j=i+1; j<len; j++) {
21                if(str[i] == str[j]) {
22                    counter++;
23                    str[j] = '\0';
24                }
25            }
26            printf("%c appears %d times.\n", str[i], counter);
27        }
28    }
29
30    return 0;
31 }
32 }
```



Day 12 20\_4\_2023\_Soumojit\_Shome - ass7\_pro7\_2.c

```
1 #include <stdio.h>
2
3 // 7. Write a C program to convert lowercase string to uppercase string and vice versa.
4
5 int main() {
6     char str[100];
7     int i;
8
9     printf("Enter a string: ");
10    gets(str);
11
12    for(i=0; str[i]!='\0'; i++) {
13        if(str[i]>='a' && str[i]<='z') {
14            str[i] = str[i] - 32;
15        }
16    }
17    printf("Uppercase string: %s", str);
18
19    for(i=0; str[i]!='\0'; i++) {
20        if(str[i]>='A' && str[i]<='Z') {
21            str[i] = str[i] + 32;
22        }
23    }
24    printf("\nLowercase string: %s", str);
25
26    return 0;
27 }
28
```



```
1 #include <stdio.h>
2
3 // 8. Write a C program to compare two strings
4
5 int main() {
6     char str1[100], str2[100];
7     int i, flag = 0;
8
9     printf("Enter first string: ");
10    gets(str1);
11
12    printf("Enter second string: ");
13    gets(str2);
14    for(i=0; str1[i]!='\0' || str2[i]!='\0'; i++) {
15        if(str1[i] != str2[i]) {
16            flag = 1;
17            break;
18        }
19    }
20
21
22    if(flag == 0) {
23        printf("The two strings are equal.");
24    }
25    else {
26        printf("The two strings are not equal.");
27    }
28
29    return 0;
30 }
31
```



Day 12 20.4.2023\_Soumojit\_Shome - ass7\_pro9\_2.c

```
1 #include <stdio.h>
2
3 // 9. Write a C program to check whether a string is palindrome or not (using single string only)
4
5 int main() {
6     char str[100];
7     int i, length = 0, flag = 0;
8
9     printf("Enter a string: ");
10    gets(str);
11
12    while(str[length] != '\0') {
13        length++;
14    }
15
16
17    for(i=0; i<length/2; i++) {
18        if(str[i] != str[length-i-2]) {
19            flag = 1;
20            break;
21        }
22    }
23
24    if(flag == 0) {
25        printf("%s is a palindrome", str);
26    }
27    else {
28        printf("%s is not a palindrome", str);
29    }
30
31    return 0;
32 }
33
```

Day 12 20\_4\_2023\_Soumojit\_Shome - ass7\_pro10\_2.c

```
1 #include <stdio.h>
2
3 // 10. Write a C program to find the total number of alphabets, digits or special characters in a string
4
5 int main() {
6     char str[100];
7     int i, alphabets = 0, digits = 0, special_chars = 0;
8
9     printf("Enter a string: ");
10    gets(str);
11
12    for(i=0; str[i]!='\0'; i++) {
13        if((str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z')) {
14            alphabets++;
15        }
16        else if(str[i]>='0' && str[i]<='9') {
17            digits++;
18        }
19        else {
20            special_chars++;
21        }
22    }
23
24
25    printf("Alphabets = %d\n", alphabets);
26    printf("Digits = %d\n", digits);
27    printf("Special characters = %d\n", special_chars);
28
29    return 0;
30 }
31
```

## Assignment – 8

The objective of this assignment is to learn how to use user define function in the program Assignment:

- 1. Write a C program to find the factorial of a number and also find the value of  ${}^nC_r$  using this function.**

### **Factorial of a number using function:**

STEP 1: **Declare a function** ‘fact’ with input parameter (n) before main function

STEP 2: In main, take the number(n) as user input

STEP 3: **Call the function** and pass the input parameter (n)

[ factorial = fact (n) ]

STEP 4: In **function definition** initialize a variable ‘f’ with 1

STEP 5: Start a for loop from i=1 to the number(n)

STEP 6: Multiply the value of ‘f’ with ‘i’ in the loop

STEP 7: Return the final value of ‘f’ to the main function

STEP 8: Print the value of ‘factorial’ in the main function

### Find the value of ${}^nC_r$ using ‘factorial function’:

[**Hint:** Take another user input r. Call the ‘fact’ function using the following equation:

${}^nC_r = \text{fact}(n) / (\text{fact}(r) * \text{fact}(n-r))$  ]

- 2. Write a C Program to find the sum of two matrices using function.**

STEP 1: **Declare a function** ‘matrixadd’ with input parameters,

[ void matrixadd (int [][][10], int [][][10], int [][][10], int, int); ]

STEP 2: In main, define 3 matrices : matrix1, matrix2, matrix3 and the required variables

[ int matrix1[10][10], matrix2[10][10], matrix3[10][10], row, col, i, j; ]

STEP 3: Take number of rows and columns of the matrices as input

STEP 4: Take 2 user defined matrix inputs (matrix1, matrix2) in 2D array.

```
printf("\nEnter the elements of the 1st matrix: ");
for(i=0;i<row;i++)
{
    for(j=0;j<col;j++)
    {
        printf("\nmatrix1[%d][%d]=",i,j);
        scanf("%d",&matrix1[i][j]);
    }
}
```

[Similarly take the elements of the 2nd matrix.]

STEP 5: **Call the function** and pass the input parameters (matrix1, matrix2, matrix3)  
[ matrixadd(matrix1,matrix2,matrix3,row,col); ]

STEP 6: In **function definition** add the 2D arrays: matrix1 and matrix2 using 2 nested  
for loops i and j.

```
void matrixadd(int matrix1[][10],int matrix2[][10],int matrix3[][10],int m, int n)
{
    int i,j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            matrix3[i][j]=matrix1[i][j]+matrix2[i][j];
        }
    }
}
```

STEP 7: Print the 2D array matrix3 as output in the main function.

```
printf("\nThe sum of the two matrices:\n");
for(i=0;i<row;i++)
{
    for(j=0;j<col;j++)
    {
        printf("%3d",matrix3[i][j]);
    }
    printf("\n");
}
```

**3. Write a C program to find power of any number using recursive function.**

STEP 1: **Declare a recursive function** ‘power’ with input parameters (base and exponent)

STEP 2: Take ‘base’ and ‘exponent’ as user inputs in main.

STEP 3: **Call the recursive function** and pass the input parameters

[ result = power (base,exponent) ]

STEP 4: In **recursive function definition** check, If the value of exponent is 0 then  
return 1 to main

STEP 5: If the value of exponent is greater than 0

return (base \* power(base, exponent-1)) [calling the function ‘power’  
recursively]

STEP 6: If the value of exponent is less than 0,

```
return (1 / pow(base, -exponent)) [calling the function 'power'  
                                          Recursively]
```

STEP 7: Print the value of 'result' in main().

**4. Write a function in C program to sort all elements of an array in ascending order using bubble sort technique.**

[Bubble sort is a comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.]

STEP 1: **Declare a function 'bubble'** with input parameters - array and size of the array  
[void bubble (int [ ], int)]

STEP 2: Take the array size and array elements as inputs in the main function.

STEP 3: **Call the 'bubble' function** from main() and pass the input parameters (array name and size of the array)

STEP 4: In **function definition** run an outer for loop (say i) from 0 to less than (size-1)

STEP 5: Run an inner for loop (say j) from 0 to less than (size-1-i)

STEP 6: Inside the inner for loop, check **if jth element of the array is greater than (j+1)th element of the array, swap the elements.**

STEP 7: Print the array after completion of sorting

**5. Write a C program to sort all elements of an array in ascending order using merge sort technique.**

[Merge sort first divides the array into equal halves and then combines them in a sorted manner.]

STEP 1:

6. Write a C program to find the GCD of two numbers using a recursive function, and also find the GCD of three numbers using this function.

7. Write a C program to find the Fibonacci series up to n term using recursive function

Practice:

8. Write a C program to find maximum and minimum elements in an array using a recursive function.

9. Write a C program to sort all elements of an array in ascending order using quick sort technique.

10. Write a C Program to count the frequency of array elements in a 1-D array

11. Write a C program to solve Tower of Hanoi problem

Objective: Move n disks from 1 rod to another rod with the help of an intermediate rod.

[NOTE: Implement the previous assignments using function call]

```
1 //1. Write a C program to find the factorial of a number and also find the value of nCr using this
2 //function.
3
4 #include<stdio.h>
5 int fact(n){
6     int i,f=1;
7     for(i=1;i<=n;i++){
8         f*=i;
9     }
10    return f;
11 }
12 int main(){
13     int n,factorial;
14     printf("Enter any number : ");
15     scanf("%d",&n);
16
17     factorial = fact(n);
18
19     printf("The factorial of %d is %d",n,factorial);
20
21
22     return 0;
23 }
```

```
1 //1. Write a C program to find the factorial of a number and also find the value of nCr using this
2 //function.
3 //Find the value of nCr using 'factorial function:
4
5 #include<stdio.h>
6 int fact(n){
7     int i,f=1;
8     for(i=1;i<=n;i++){
9         f*=i;
10    }
11 }
12 int main(){
13     int n,r;
14     float ncr;
15     printf("Enter n : ");
16     scanf("%d",&n);
17     printf("Enter r : ");
18     scanf("%d",&r);
19
20
21     ncr=fact(n)/(fact(r)*fact(n-r));
22
23     printf("nCr = %f",ncr);
24
25
26
27     return 0;
28 }
```

```
1 //2. Write a C Program to find the sum of two matrices using function.
2
3 #include<stdio.h>
4 void matrixadd(int matrix1[][10], int matrix2[][10], int matrix3[][10], int m, int n){
5     int i,j,row,col;
6     for(i=0;i<m;i++){
7         for(j=0;j<n;j++){
8             matrix3[i][j]= matrix1[i][j]+matrix2[i][j];
9         }
10    }
11
12 }
13
14 int main(){
15     int matrix1[10][10], matrix2[10][10], matrix3[10][10], row, col, i, j;
16
17     printf("Enter number of rows : ");
18     scanf("%d",&row);
19
20     printf("Enter number of columns : ");
21     scanf("%d",&col);
22
23
24     printf("\nEnter the elements of the 1st matrix: ");
25     for(i=0;i<row;i++){
26         for(j=0;j<col;j++){
27             printf("\nmatrix1[%d][%d]=",i,j);
28             scanf("%d",&matrix1[i][j]);
29         }
30     }
31
32     printf("\nEnter the elements of the 2nd matrix: ");
33     for(i=0;i<row;i++){
34         for(j=0;j<col;j++){
35             printf("\nmatrix2[%d][%d]=",i,j);
36             scanf("%d",&matrix2[i][j]);
37         }
38     }
39
40     matrixadd(matrix1,matrix2,matrix3,row,col);
41
42     printf("\nThe sum of the two matrices:\n");
43     for(i=0;i<row;i++){
44         for(j=0;j<col;j++){
45             printf("%3d",matrix3[i][j]);
46         }
47         printf("\n");
48     }
49
50
51     return 0;
52 }
```

//3. Write a C program to find power of any number using recursive function.

```
#include<stdio.h>

float power(int base, int exponent){
    if(exponent==0)
        return 1;
    else if(exponent>0)
        return (base * power(base, exponent-1));
    else if(exponent<0)
        return (1 / (float)power(base, -exponent));
}

int main(){
    int base, exponent;
    float result;
    printf("Enter base : ");
    scanf("%d",&base);

    printf("Enter exponent : ");
    scanf("%d",&exponent);

    result = power(base,exponent);

    printf("The value of %d to the power %d = %f",base, exponent,result);

    return 0;
}
```

//4. Write a function in C program to sort all elements of an array in ascending order using bubble sort technique.

```
#include<stdio.h>

int bubble(int arr[ ], int n){
    int i,j,temp;
    for(i=0;i<n-1;i++){
        for(j=0;j<n-i-1;j++){
            if(arr[j]>arr[j+1]){
                temp = arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
}

int main(){
    int n,k;
    printf("Enter size of array : ");
    scanf("%d",&n);

    int arr[n];

    for(k=0;k<n;k++){
        printf("Enter element for index %d : ",k);
        scanf("%d",&arr[k]);
    }

    bubble(arr,n);

    printf("Sorted array is : ");
    for(k=0;k<n;k++)
        printf("%d, ",arr[k]);

    return 0;
}
```

//5. Write a C program to sort all elements of an array in ascending order using marge sort technique

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

```
int merge(int arr[], int p, int q, int r){  
    int n1 = q-p+1;  
    int n2 = r-q;  
  
    int L[n1],M[n2];  
  
    for (int i=0;i<n1;i++)  
        L[i]=arr[p+i];  
    for (int j=0;j<n2;j++)  
        M[j]=arr[q+1+j];  
  
    int i=0,j=0,k=p;  
  
    while(i<n1 && j<n2){  
        if(L[i]<=M[j]){

            arr[k]=L[i];
            i++;
        }
        else{
            arr[k]=M[j];
            j++;
        }
        k++;
    }
    while(i<n1){
        arr[k]=L[i];
        i++;
        k++;
    }
    while(j<n2){
        arr[k]=M[j];
        j++;
        k++;
    }
}  
  
int mergesort(int arr[], int l, int r){  
    if(l<r){  
        int m = l+(r-l)/2;  
  
        mergesort(arr,l,m);  
        mergesort(arr,m+1,r);  
  
        merge(arr,l,m,r);  
    }
}  
  
int main(){  
    int n,k;  
    printf("Enter size of array : ");  
    scanf("%d",&n);  
  
    int arr[n];  
  
    for(k=0;k<n;k++){  
        printf("Enter element for index %d : ",k);
        scanf("%d",&arr[k]);
    }
}
```

```
mergesort(arr,0,n-1);

printf("Sorted array is : ");
for(k=0;k<n;k++)
    printf("%d, ",arr[k]);

return 0;
}
```

## Assignment – 9

The objective of this assignment is to learn how to use pointer in the program

Assignment:

1. Write a C Program using pointer to find the bigger of two given numbers.
2. Write a C Program to swap two given numbers using call by reference.
3. Write a C program to copy one array to another using pointers.
4. Write a C program to reverse an array using pointers.
5. Write a C program to add two matrix using pointers.

Practice:

1. Write a C program to copy one string to another using pointers.
2. Write a C program to concatenate two strings using pointers.
3. Write a C program to sort an array using pointers (Any technique).
4. Write a C program to search an element in array using pointers.

## Assignment – 10

**The objective of this assignment is to learn how we can use Structures and Unions and also how-to dynamic memory allocation happen in C Program.**

1. Write a program to store and print the roll no., name, age and marks of a student using structures.
2. Write a program to store the roll no. (starting from 1), name and age of 5 students and then print the details of the student with roll no. 2.
3. Write a program to store and print the roll no., name, age, address and marks of 15 students using structure.
4. Write a program to add two distances in inch-feet using structure. The values of the distances is to be taken from the user.
5. Enter the marks of 5 students in Chemistry, Mathematics and Physics (each out of 100) using a structure named Marks having elements roll no., name, chem\_marks, maths\_marks and phy\_marks and then display the percentage of each student.
6. Write a program to add, subtract and multiply two complex numbers using structures to function.
7. Define a union with the following three members: roll no, name and total marks of student. Write a c program to read and display the details of a student.
8. C Program to Find Largest Number from an array. Array must be declared using Dynamic Memory Allocation.

**Practice:**

1. Write a structure to store the roll no., name, age (between 11 to 14) and address of students (more than 10). Store the information of the students.
  - 1-Write a function to print the names of all the students having age 14.
  - 2-Write another function to print the names of all the students having even roll no.
  - 3-Write another function to display the details of the student whose roll no is given (i.e. roll no. entered by the user).
2. Write a structure to store the name, account number and balance of customers (more than 10) and store their information.
  - 1- Write a function to print the names of all the customers having balance less than \$200.
  - 2- Write a function to add \$100 in the balance of all the customers having more than \$1000 in their balance and then print the incremented value of their balance.
3. Write a program to compare two dates entered by user. Make a structure named Date to store the elements day, month and year to store the dates. If the dates are equal, display “Dates are equal” otherwise display “Dates are not equal”.
4. Write a structure to store the names, salary and hours of work per day of 10 employees in a company. Write a program to increase the salary depending on the number of hours of work per day as follows and then print the name of all the employees along with their final salaries.

Hours of work per day	8	10	>=12
Increase in salary	\$50	\$100	\$150

5. Let us work on the menu of a library. Create a structure containing book information like accession number, name of author, book title and flag to know whether book is issued or not. Create a menu in which the following can be done.

- 1 - Display book information
- 2 - Add a new book
- 3 - Display all the books in the library of a particular author
- 4 - Display the number of books of a particular title
- 5 - Display the total number of books in the library
- 6 - Issue a book

(If we issue a book, then its number gets decreased by 1 and if we add a book, its number gets increased by 1)

GOOD LUCK  
FOR YOUR  
EXAM AND  
DO THE BEST