Important C Programs Notes - Believer 01

Made with love and thanks for your tremendous support

Creator : Koluboyina Varaprasad (Bobby)

Location : Kakinada , East Godavari , 533001, Andhra Pradesh, India.

Contact us : believer01.official@gmail.com

Channel Name : Believer 01

If you want to support my work, **you can donate any amount of your choice even 1rupee is valuable**

Details :

Upi id : believer01.official@okhdfcbank

Content:

13. Maximum Number of Array
14. Half Pyramid of *
15. Half Pyramid of Numbers
16. Inverted half pyramid of *
17. full pyramid of *
18. full pyramid of numbers
19. inverted full pyramid of *
20. Pascal's Triangle
21. Floyd's Triangle
22. Find the 2's complement of a binary number
23. C Program to find the roots of Quadratic Equation
24. Addition and multiplication of complex numbers
25. C program to find the number of characters in a given string including and excluding spaces
26. Given string is palindrome or not
27. Reverse a string using pointers
28. write a c program to implement student details using structures
29. Matrix Addition
30. Matrix Multiplication
31. write a c program which copies one file to another
32. write a c program to count the number of characters and number of lines in a file
33. write a c program to merge two files into a third file
34. AP Leader Ellection Project in C language (c project)
35. Eamcet Exam Management C language (c project)

1. Fibonacci Series

Question :

The Fibonacci numbers are the numbers in the following integer sequence.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

$$Fn = Fn-1 + Fn-2$$

The series starts of with 0 and 1 , our aim is to build a series of numbers with above equation , like below

0, 1 , 0+1 = 1 , 1+1 = 2 , 1+2 = 3, 2+3 = 5, 3+5 = 8 , ... => 0,1,1,2,3,5,8 .....

`Input / output :`

```
Input  : n = 9
Output : 34
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int fib(int n)
{
    if(n==0) return -1;
    if(n==1) return 0;
    if(n==2) return 1;
    int a=0,b=1,c,i;
    for(i=2;i<n;i++)
    {
        c=a+b;
        a=b;
        b=c;
    }
    return c;
}
int main()
{
    int n;
    printf("Enter N value :");
    scanf("%d",&n);
    int ans = fib(n);
    printf("Answer : %d\n",ans);
    return 0;
}
```

2. Find the Sum and Product of Individual Digits of a Positive Number

Question :

Finding the Sum and Product of Individual Digits of a Positive Number

For example

Number 123 =>

Sum = 1 + 2 + 3 = 6

Product = 1 * 2 * 3 = 6

Input / output :

```
Input  : n = 153
Output :
Sum : 9
Product : 15
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n,r;
    printf("Enter N value:");
    scanf("%d",&n);
    int sum =0, product = 1;
    while(n>0)
    {
        r = n%10;
        sum = sum + r;
        product = product * r;
        n = n/10;
    }
    printf("Sum : %d\n",sum);
    printf("Product : %d\n",product);
    return 0;
}
```

3. Factorial of a given Integer

Question :

Finding Factorial of a given Integer

Factorial of a Number is calculated as multiplying each number from 1 to itself

For example

Factorial of 5 = 1 * 2 * 3 * 4 * 5 = 120

=> 5! = 120

Factorial of a given number is a classic Example for understanding Loops concept in any programming language that includes C programming.

Input / output :

```
Input  : n = 6

Output :

Factorial : 720
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int factorial(int n)
{
    if(n==0)
        return 1;
    else
    {
        int i,result=1;
        for(int i=1;i<=n;i++)
            result = result * i;
        return result;
    }
}
int main()
{
    int n;
    printf("Enter N value : ");
    scanf("%d",&n);
    int ans = factorial(n);
    printf("Factorial : %d",ans);
    return 0;
}
```

4.Given Number is Palindrome or Not

Question :

Checking whether given number is Palindrome or Not is again example to understand Loops concept in any programming language

Palindrome means even if we reverse certain number it should look exactly same those numbers are called Palindromes

For example

121 is a palindrome

Reverse of 121 = 121

Input / output :

```
Input  : n = 15551
Output :
15551 is a Palindrome
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int palindrome(int n)
{
    int r,sum=0,temp = n;
    while(n>0)
    {
        r = n%10;
        sum =(sum*10)+r;
        n = n/10;
    }
    if(sum==temp)
        return 1;
    else
        return 0;
}
int main()
{
    int n;
    printf("Enter N value : ");
    scanf("%d",&n);
    int ans = palindrome(n);
    if(ans==1)
        printf("%d is a Palindrome!",n);
    else
        printf("%d is not a Palindrome!",n);
    return 0;
}
```

5 . Armstrong Number or Not

Question :

Checking whether given number is Armstrong or Not

One of popular program to learn Loops working in any programming language

Armstrong number :

The number is equal to the sum of cube roots of its digits is called as Armstrong Number

For example

153 = > 1^3 + 5^3 + 3^3 = 1 + 125+27 = 153

Input / output :

```
Input  : n = 153

Output :

15551 is an Armstrong Number
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int armstrong(int n)
{
    int r,sum=0,temp = n;
    while(n>0)
    {
        r = n%10;
        sum = sum + r*r*r;
        n = n/10;
    }
    if(sum==temp)
        return 1;
    else
        return 0;
}
int main()
{
    int n;
    printf("Enter N value : ");
    scanf("%d",&n);
    int ans = armstrong(n);
    if(ans==1)
        printf("%d is an Armstrong!",n);
    else
        printf("%d is not an Armstrong!",n);
    return 0;
}
```

6. Simple Calculator using switch Statement

Question :

Simple Calculator Using Switch Case Statement is pretty straight forward and Very easy to write, this calculator we are going to perform 4 operations namely

Addition, Subtraction , Multiplication, Division

Input / output :

Input :

```
6 3
```

Output :

```
Addition : 9
Subtraction : 3
Multiplication : 18
Division : 2
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int a,b,option;
    printf("Enter a,b values: ");
    scanf("%d%d",&a,&b);
    while(1)
    {
        printf("Choose option\n");
        printf("1)Addition\n");
        printf("2)Subtraction\n");
        printf("3)Multiplication\n");
        printf("4)Division\n");
        printf("Enter Option:");
        scanf("%d",&option);
        switch(option)
        {
        case 1:
            printf("Addition: %d \n",a+b);
            break;
        case 2:
            printf("Subtraction: %d \n",a-b);
            break;
        case 3:
            printf("Multiplication: %d \n",a*b);
            break;
        case 4:
            printf("Division: %.2f \n",(float)a/b);
            break;
        default:
            printf("Invalid Option! \n");
        }
    }
    return 0;
}
```

7. Reverse a Number

Question :

Reversing a given number is one of important program to understand loops concept , it involve simple mathematics concept to write the program

Reversing number is easy for example

Reverse of 153 => 351

`Input / output :`

```
Input  : n = 153

Output :

Reversed Number : 351
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n;
    printf("Enter Any Number:");
    scanf("%d",&n);
    int reverse_number=0,r,temp =n;
    while(temp>0)
    {
        r = temp%10;
        reverse_number = reverse_number * 10 + r;
        temp = temp/10;
    }
    printf("Before Reverse : %d \n",n);
    printf("After Reverse : %d\n",reverse_number);
    return 0;
}

```

8. Prime Number or Not

Question :

Check whether the given number is prime number or not

Prime number is defined as the number which is only divisible by 1 and itself

All prime numbers are odd numbers except 2

Examples : 2,3,5,7,11,13,17,19 .....

Input / output :

```
Input  : n = 97

Output :

97 is a Prime Number
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n;
    printf("Enter Any Number : ");
    scanf("%d",&n);
    // now checking given number is prime or not
    int counter = 0,i;
    for(i=1;i<=n;i++)
    {
        if(n%i==0)
            counter++;
    }
    if(count==2)
        printf("%d is a Prime Number!",n);
    else
        printf("%d is not a Prime Number!",n);
    return 0;
}
```

9. Multiplication table of a Given Number

Question :

Write a program to find the multiplication table of a given number

Mathematical table is one of the classic example to learn loops and it is easy to understand.

Input / output :

```
Input  : n = 9
Output :
9 X 1 = 9
9 X 2 = 18
9 X 3 = 27
9 X 4 = 36 ……..
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n;
    printf("Enter Table Number:");
    scanf("%d",&n);
    int i;
    for(i=1;i<=10;i++)
    {
        printf("%d X %d = %d\n",n,i,n*i);
    }
    return 0;
}
```

10. Write a c program to swap two numbers without using temporary Variable

Question :

Swapping to two numbers means replace both numbers by each other

We can easily do the swapping using third variable, but we have to do it without variables

Example:

a=10 , b=20

=> a=20 , a=10.

Input / output :

```
Input  : a =10 , b = 20

Output :

a = 20

b = 10
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int a,b;
    printf("Enter a value:");
    scanf("%d",&a);
    printf("Enter b value:");
    scanf("%d",&b);
    // swapping without third variable
    printf("before swapping\n a = %d \n b= %d\n",a,b);
    a = a+b;
    b = a-b;
    a = a-b;
    printf("after swapping\n a = %d \n b= %d\n",a,b);
    return 0;
}

```

11. GCD , Greatest common Divisor of two numbers

Question :

Greatest common Divisors of two numbers is again classic example of learning loops and if else statements

Basically GCD of two numbers is finding greatest common number that will divide both the two numbers

Example:

a=18 , b=27

Both 18 and 27 can be divisible by 9, and 9 is the greatest common divisor for both numbers

`Input / output :`

```
Input  : a = 33 , b = 44

Output :

GCD : 11
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int gcd(int a,int b)
{
    int i,min_value,ans=1;
    if(a<b)
        min_value = a;
    else
        min_value = b;
    for(i=2;i<=min_value;i++)
    {
        if(a%i==0 && b%i==0)
            ans = i;
    }
    return ans;
}
int main()
{
    int a,b;
    printf("Enter a value:");
    scanf("%d",&a);
    printf("Enter b value:");
    scanf("%d",&b);
    int ans = gcd(a,b);
    printf("GCD : %d",ans);
    return 0;
}
```

12. Check Leap Year

Question :

To check whether the given year is leap year not we have certain conditions those are

All years that are divisible by 4, except the years whose value end with 00 and not divisible by 400 like 1800,1700,1900 are not leap years

Example:

year : 1900 , is not a leap year because it is divisible by 100 and not by 400.

`Input / output :`

```
Input  : year : 2001
Output :
2001 is not a leap year
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int year;
    printf("Enter Year: ");
    scanf("%d",&year);
    //now checking the leap year
    // if flag is 0, not leap year
    // if flag is 1, it is a leap year
    int flag;
    if(year%4==0)
    {
        if(year%100 == 0 && year%400 != 0)
            flag = 0;
        else
            flag = 1;
    }
    else
        flag = 0;
    // now printing
    if(flag == 0)
        printf("%d is not a leap year!",year);
    else
        printf("%d is a leap year!",year);
    return 0;
}
```

13.Maximum Number of Array

Question :

Finding Maximum Number of an Array of positive numbers is very important

Problem asked in almost all interviews, it requires loops and if else concepts

Arr = [23,56,75,43,78]

Max Value : 78

Input / output :

```
Input  : Enter array size : 5
Enter 5 values : 54 34 89 54 21
Output :
Max value : 89
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n,i;
    printf("Enter array size:");
    scanf("%d",&n);
    int *arr = (int*)malloc(n*sizeof(int*));
    printf("Enter %d values (+ve numbers):",n);
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    int max_value= -1;
    for(i=0;i<n;i++)
    {
        if(max_value<arr[i])
            max_value = arr[i];
    }
    printf("Max value : %d",max_value);
    return 0;
}
```

14.Half Pyramid of *

Question :

Pattern printing is one of the important and effective way to learn loops

Half Pyramid of * is one way to learn.

Pattern :

```
*
* *
* * *
* * * *
* * * * *
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n;
    printf("Enter size of pyramid: ");
    scanf("%d",&n);
    int i,j;
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

15. Half Pyramid of Numbers

Question :

Pattern printing is one of the important and effective way to learn loops

Half Pyramid of Number is one way to learn.

Pattern :

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n;
    printf("Enter size of pyramid: ");
    scanf("%d",&n);
    int i,j,k;
    for(i=1;i<=n;i++)
    {
        k=1;
        for(j=1;j<=i;j++)
        {
            printf("%d ",k);
            k++;
        }
        printf("\n");
    }
    return 0;
}
```

16.Inverted half pyramid of *

Question :

Pattern printing is one of the important and effective way to learn loops

Inverted Half Pyramid of Number is one way to learn.

Pattern :

```
* * * * *
* * * *
* * *
* *
*
```

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n;
    printf("Enter size of pyramid: ");
    scanf("%d",&n);
    int i,j;
    for(i=n;i>=1;i--)
    {
        for(j=1;j<=i;j++)
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

17.Full pyramid of *

Question :

Pattern printing is one of the important and effective way to learn loops

Full Pyramid of Number is one way to learn.

Pattern :

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

Observation:

For Row 2 => 3 * should be there => 2 x i -1 => 2 x 2 -1 = 3

Formula = 2*i-1 , where i = number of row

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n,i,spaces,j;
    printf("Enter Number of Rows : ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        for(spaces=0;spaces<n-i;spaces++)
            printf(" ");
        for(j=0;j<2*i-1;j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

18. Full pyramid of numbers

Question :

Pattern printing is one of the important and effective way to learn loops

Full Pyramid of Number is one way to learn.

Pattern :

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

Observation :

For each row we should have 2 x i - 1 , elements, where “i” is number of row
I have split the each row to two parts like ,
size = 2 x i-1
first = (size/2)+1, second = size-first;, more details in the videos linked to this.

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n,i,spaces,j,k;
    printf("Enter Number of Rows : ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        for(spaces=0;spaces<n-i;spaces++)
            printf("  ");
        int size = 2*i-1;
        int first = (size/2) + 1;
        int second = size-first;
        k= i;
        for(j=0;j<first;j++)
        {
            printf("%2d",k);
            k++;
        }
        k = k-2;
        for(j=0;j<second;j++)
        {
            printf("%2d",k);
            k--;
        }
        printf("\n");
    }
    return 0;
}
```

19.Inverted Full pyramid of *

Question :

Pattern printing is one of the important and effective way to learn loops

Inverted Full Pyramid of * is one way to learn.

Pattern :

****

****

*****

***

*

Observation:

Number of elements in a row , Formula = 2*i-1 ,where i =N - row number + 1

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n,i,spaces,j;
    printf("Enter Number of Rows : ");
    scanf("%d",&n);
    for(i=n;i>=1;i--)
    {
        for(spaces=0;spaces<n-i;spaces++)
            printf(" ");
        for(j=0;j<2*i-1;j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

20. Pascal's Triangle

Question :

Pattern printing is one of the important and effective way to learn loops

Pascal's Triangle is one way to learn.

Pattern :

```
        1
      1   1
    1   2   1
  1   3   3   1
1   4   6   4   1
1  5  10  10  5  1
```

Observation :

Formula for Pascal's Triangle => coef = coef * (i-j+1)/j

Pascal's triangle is soo famous and lot of mathematical series represent this.

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n,i,spaces,j,coef;
    printf("Enter Number of Rows : ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        for(spaces=1;spaces<n-i;spaces++)
            printf("  ");
        for(j=0;j<=i;j++)
        {
            if(j==0)
                coef = 1;
            else
                coef = coef * (i-j+1)/j;
            printf("%4d",coef);
        }
        printf("\n");
    }
    return 0;
}
```

21. Floyd's Triangle

Question :

Pattern printing is one of the important and effective way to learn loops

Floyd's Triangle is one way to learn.

Pattern :

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

Observation :

Floyd Traingle is one of easy pattern, it is just like a half pyramid with numbers in increasing order.

Code : (source code link :)

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n,i,j,k=1;
    printf("Enter Number of Rows : ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        for(j=0;j<=i;j++)
        {
            printf("%d ",k);
            k++;
        }
        printf("\n");
    }
    return 0;
}
```

22. Find the 2's complement of a binary number

Question :

Binary Number is one of the way of representing numbers which include 0's and 1's.

Before Finding 2's complement we have to find 1's complement , and 1's complement can be find by flipping the binary number and later we can easily find the 2's complement, for more details watch the linked video and practice

`Input / output :`

```
Input  : 0111

Output :
1's complement : 1000
2's complement : 1001
```

Code : (source code link :)

```c
#include<stdio.h>
#include<string.h>
int main()
{
    char s[64];
    int n,i,cnt=0;
    printf("Enter Binary Number : ");
    scanf("%s",s);
    n = strlen(s);
    for(i=0;i<n;i++)
        s[i] = s[i] == '0'? '1' : '0';
    for(i=n-1;i>=0;i--)
    {
        if(s[i]=='1')
        {
            s[i]='0';
            cnt++;
        }
        else
        {
            s[i] = '1';
            break;
        }
    }
    if(cnt == n)
        printf("2s Complement : %s",strcat("1",s));
    else
        printf("2s Complement : %s",s);
    return 0;
}
```

23. C Program to find the roots of Quadratic Equation

Question :

Finding roots of a quadratic equation is one the classic example to learn mathematical operators in programming

We have

Roots = (-b/2a)±√(b^2-4ac)/2a where a,b,c are the coefficients

By using above formula we will find the roots and watch the video for brief explanation

Input / output :

```
Input  : Enter Coefficients a,b,c : 1 -2 1

Output :

Roots : 1 1
```

Code : (source code link :)

```c
#include<stdio.h>
#include<math.h>
int main()
{
    double a,b,c,discriminant,root1,root2;
    printf("Enter Coefficients (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("root1 = %.2lf and root2 = %.2lf\n", root1, root2);
    }
    else if(discriminant==0)
    {
        root1 = root2 = -b / (2 * a);
        printf("root1 = %.2lf and root2 = %.2lf\n", root1,root2);
    }
    else
    {
        double real,img;
        real = -b / (2 * a);
        img = sqrt(-discriminant) / (2 * a);
        printf("root1 = %.2lf+%.2lfi and root2 = %.2lf-%.2lfi\n",real,img,real,img);
    }
    return 0;
}
```

24. Addition and multiplication of complex numbers

Question :

Any number in the form of (a+bi) is called as complex number , addition and multiplication of complex number helps us to learn structure **concept in c programming ,** let **a+bi and c+di** are two complex numbers

We have

Addition = (a+c) + (b+d)i

Multiplication = (ac-bd) + (ad+bc)i

`Input / output :`

```
Input  : First Number(a+bi) : 2 3
Second Number (c+di) : 4 5
Output :
Addition : 6 + 8i
Multiplication :  -7 + 22i
```

Code : (source code link :):

```c
#include<stdio.h>
#include<stdlib.h>
struct complex
{
    int real;
    int imaginary;
};
int main()
{
    struct complex c1,c2,sum,mul;
    printf("Enter First Complex Number a,b of(a+bi) :");
    scanf("%d %d",&c1.real,&c1.imaginary);
    printf("Enter Second Complex Number c,d of(c+di) :");
    scanf("%d %d",&c2.real,&c2.imaginary);
    sum.real = (c1.real+c2.real);
    sum.imaginary = (c1.imaginary+c2.imaginary);
    mul.real = (c1.real*c2.real-c1.imaginary*c2.imaginary);
    mul.imaginary = (c1.real*c2.imaginary+c1.imaginary*c2.real);
    printf("Addition : %d + %di\n",sum.real,sum.imaginary);
    printf("Multiplication : %d + %di\n",mul.real,mul.imaginary);
    return 0;
}
```

25. C program to find the number of characters in a given string including and excluding spaces

Question :

Finding the number of characters of a string with and without spaces is one good example to learn about strings and loops

'\0' is called null character which represents end of the string

Ascii value for space is 32 , based on this we will calculate the number of characters of a string with and without space

Input / output :

```
Input  : String : Consistency is Hardwork
Second Number (c+di) : 4 5
Output :
Number of Characters with Spaces : 23
Number of Characters without Spaces : 21
```

Code : (source code link :):

```c
#include<stdio.h>
#include<conio.h>
int main()
{
    char input[100];
    int countWithSpaces=0,countWithoutSpaces=0,i=0;
    printf("Enter Any English Sentence :");
    gets(input);
    while(input[i]!='\0')
    {
       if(input[i]== 32)
       {
           countWithSpaces++;
       }
       else
       {
           countWithSpaces++;
           countWithoutSpaces++;
       }
    }
    printf("Number of Characters with Spaces : %d \n",countWithSpaces);
    printf("Number of Characters without Spaces : %d \n",countWithoutSpaces);
    return 0;
}
```

26. Given string is palindrome or not

Question :

Palindrome means if a string is the same in both the directions i.e both from left to right and from right to left , then that string is called palindrome.

We will maintain two variables one from start and other from the end, and will check both the points are same as they move towards each other through loop

`Input / output :`

```
Input  : String : ABBCBBA

Output :

ABBCBBA is a palindrome!
```

Code : (source code link :):

```c
#include<stdio.h>
#include<string.h>
int main()
{
    char input[100];
    int low,high,flag=1;
    printf("Enter Any String Value :");
    scanf("%s",input);
    low = 0;
    high = strlen(input)-1;
    while(low<high)
    {
        if(input[low]!=input[high])
        {
            flag = 0;
            break;
        }
        low++;
        high--;
    }
    if(flag == 0)
        printf("%s is not a palindrome!",input);
    else
        printf("%s is a palindrome!",input);
    return 0;
}
```

27. Reverse a string using pointers

Question :

Reversing a String include Loops and swapping operations, by using this program you can learn about loops and maintaining two variables in loops

Reversing operations include swapping operations, try to watch the linked video completely and understand and enjoy the coding show.

We are going to use pointers concept to do Reversal operation on Strings

Input / output :

```
Input  : String : Consistency

Output :

Before Reversal : Consistency

After Reversal : ycnetsisnoC
```

Code : (source code link :):

```c
#include<stdio.h>
#include<string.h>
void swap(char *first,char *second)
{
    char temp = *first;
    *first = *second;
    *second = temp;
}
int main()
{
    char input[100];
    char *low,*high;
    int i,l;
    printf("Enter Any String Value :");
    scanf("%s",input);
    printf("Before Reversal : %s\n",input);
    l= strlen(input);
    low = input;
    high = input;
    for(i=0;i<l-1;i++)
        high++;
    for(i=0;i<l/2;i++)
    {
        swap(low,high);
        low++;
        high--;
    }
    printf("After Reversal : %s\n",input);
    return 0;
}
```

28. write a c program to implement student details using structures

Question :

Implementing Structures is important, as it involves combining of other data types in one place and creating our required version of data type for example in this case student structure,

In this program we are going to read input and then print the received input to have hands-on practice on structure concept.

Input / output :

```
Input  : Enter number of student : 1

Name : Bobby , age : 22 , roll no : 10

Output :

Student 1 data :

Name : Bobby, age : 22 , roll no : 10
```

Code : (source code link :):

```c
#include<stdio.h>
#include<stdlib.h>
struct student {
    char name[40];
    int roll_number;
    int age;
};
int main()
{
    int n,i;
    printf("Enter Number of Students :");
    scanf("%d",&n);
    struct student *s = (struct student*)malloc(n*sizeof(struct student));
    for(i=0;i<n;i++)
    {
       printf("Enter Student - %d Data: \n",i+1);
       printf("Name :");
       scanf("%s",s[i].name);
       printf("Roll Number:");
       scanf("%d",&s[i].roll_number);
       printf("Age :");
       scanf("%d",&s[i].age);
    }
    printf("Received Data : \n");
    for(i=0;i<n;i++)
    {
        printf("Student no : %d\n",i+1);
        printf("Name: %s \n",s[i].name);
        printf("Roll Number: %d \n",s[i].roll_number);
        printf("Age: %d \n",s[i].age);
    }
    return 0;
}
```

29. Matrix Addition

Question :

Addition of Two Matrix can be performed on same Dimension matrices i.e matrices with same number of rows and columns, in this program we are going to perform Addition Operation by using **Dynamic Memory Allocation concept**

We are going to create 2D arrays dynamically and perform the matrix operation by taking values through input and printing the result.

Input / output :

```
Input  : Enter matrix dimensions (m x n): 2 2
Enter matrix 1: 1 2 3 4
Enter matrix 2: 5 6 7 8
Output : result :
6 8
10 12
```

Code : (source code link :):

```c
#include<stdio.h>
#include<stdlib.h>
void print_matrix(int **matrix,int r,int c)
{
    int i,j;
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            printf("%d ",matrix[i][j]);
        printf("\n");
    }
}
int** create_matrix(int r,int c)
{
    int i;
    int **matrix=(int **)malloc(r*sizeof(int **));
    for(i=0;i<r;i++)
        matrix[i]=(int*)malloc(c*sizeof(int *));
    return matrix;
}
int main()
{
    int m,n,i,j,k;
    printf("Enter Matrix Dimensions (m x n) :");
    scanf("%d %d",&m,&n);
    int **A = create_matrix(m,n);
    printf("Enter 1st matrix data :\n");
```

Continuation code ..

```
28      for(i=0;i<m;i++)
29      {
30          for(j=0;j<n;j++)
31              scanf("%d",&A[i][j]);
32      }
33      // Second Matrix
34      int **B = create_matrix(2,3);
35      printf("Enter 2nd matrix data :\n");
36      for(i=0;i<m;i++)
37      {
38          for(j=0;j<n;j++)
39              scanf("%d",&B[i][j]);
40      }
41      printf("Matrix A :\n");
42      print_matrix(A,m,n);
43      printf("Matrix B :\n");
44      print_matrix(B,m,n);
45      int **C =  create_matrix(m,n);
46      for(i=0;i<m;i++)
47      {
48          for(j=0;j<n;j++)
49          {
50              C[i][j] = A[i][j] + B[i][j];
51          }
52      }
53      printf("Addition Matrix :\n");
54      print_matrix(C,m,n);
55
56      return 0;
57  }
```

30. Matrix Multiplication

Question :

Multiplication of Two matrices is a little complex operation; we need to check the dimensions of two matrices before performing multiplication.

Matrix -1 : A (r1 x c1) and Matrix -2: B (r2 x c2)

Here c1 === r2 , then only matrix multiplication will be possible

The result matrix will be of (r1 x c2) dimension, more details in the video

Again we are going to use **Dynamic Memory Allocation concept** for arrays

Input / output :

```
Input  : Enter matrix -1 dimensions (m x n): 2 2
Enter matrix 1: 1 2 3 4
Enter matrix -2 dimensions (m x n): 2 2
Enter matrix 2: 5 6 7 8
Output : result :
19 22
43 50
```

Code : (source code link :):

```c
#include<stdio.h>
#include<stdlib.h>
void print_matrix(int **matrix,int r,int c)
{
    int i,j;
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            printf("%d ",matrix[i][j]);
        printf("\n");
    }
}
int** create_matrix(int r,int c)
{
    int i;
    int **matrix=(int **)malloc(r*sizeof(int **));
    for(i=0;i<r;i++)
        matrix[i]=(int*)malloc(c*sizeof(int *));
    return matrix;
}
```

Continuation code ...

```c
int main()
{
    int a_row,a_column,b_row,b_column,i,j,k;
    // First Matrix
    printf("Enter 1st matrix dimensions (m x n) :");
    scanf("%d %d",&a_row,&a_column);
    int **A = create_matrix(2,2);
    printf("Enter 1st matrix data :\n");
    for(i=0;i<a_row;i++)
    {
        for(j=0;j<a_column;j++)
            scanf("%d",&A[i][j]);
    }
    // Second Matrix
    printf("Enter 2nd matrix dimensions (m x n) :");
    scanf("%d %d",&b_row,&b_column);
    int **B = create_matrix(2,3);
    printf("Enter 1st matrix data :\n");
    for(i=0;i<b_row;i++)
    {
        for(j=0;j<b_column;j++)
            scanf("%d",&B[i][j]);
    }
    // Now Printing the data and calculating
    printf("Matrix A :\n");
    print_matrix(A,a_row,a_column);
    printf("Matrix B :\n");
    print_matrix(B,b_row,b_column);
    if(a_column != b_row)
        printf("Multiplication not Possible!\n");
    else
    {
        int **C =  create_matrix(a_row,b_column);
        for(i=0;i<a_row;i++)
        {
            for(j=0;j<b_column;j++)
            {
                int sum=0;
                for(k=0;k<a_column;k++)
                {
                    sum = sum + A[i][k]*B[k][j];
                }
                C[i][j] = sum;
            }
        }
        printf("Result :\n");
        print_matrix(C,a_row,b_column);
    }
    return 0;
}
```

31. write a c program which copies one file to another

Question :

File Handling is one of the important concept in C programming and in this program we are going to copy the data from file to another file

We are going to open an already existing file **input.txt** and take data from it and will create a new text file called **output.txt** and copy the data into it.

Input.txt (existing file):

```
Hello world!
Believer 01 youtube channel please support us!
```

Output.txt (new file):

```
Hello world!
Believer 01 youtube channel please support us!
```

Code : (source code link :):

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fptr1,*fptr2;
    //opening the input.txt file
    fptr1 = fopen("input.txt","r");
    if(fptr1==NULL)
        printf("Failed to load the existing file!\n");
    else
    {
        fptr2 = fopen("output.txt","w");
        if(fptr2==NULL)
            printf("Error while creating new File!\n");
        else
        {
            char c = fgetc(fptr1);
            while(c!=EOF)
            {
                fputc(c,fptr2);
                c = fgetc(fptr1);
            }
            printf("Data from the file is copied successfully!\n");
        }
    }
    fclose(fptr1);
    fclose(fptr2);
    return 0;
}
```

32. write a c program to count the number of characters and number of lines in a file

Question :

File Handling is one of the important concept in C programming and in this program we are going to calculate number of characters and lines in the file

We are going to calculate the number of lines and number of characters present in the file. Lets see the existing file name “input.txt” and lets calculate data

Input.txt (existing file):

```
Hello world!
Believer 01 youtube channel please support us!
```

Output.txt (new file):

```
Number of Characters : 58
Number of Lines in the File : 2
```

Code : (source code link :):

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fptr1;
    //opening the input.txt file
    fptr1 = fopen("input.txt","r");
    if(fptr1==NULL)
        printf("Failed to load the existing file!\n");
    else
    {
            char c = fgetc(fptr1);
            int lineCount=0,characterCount=0;
            while(c!=EOF)
            {
               if(c=='\n')
                lineCount++;
               characterCount++;
               c = fgetc(fptr1);
            }
            printf("Number of Lines in the File : %d\n",lineCount+1);
            printf("Number of Characters in the File : %d\n",characterCount);
    }
    fclose(fptr1);
    return 0;
}
```

33. write a c program to merge two files into a third file

Question :

File Handling is one of the important concept in C programming and in this program we are going to merge two files data into the third file

We have two files **first.txt** and **second.txt** and we are going to merge data into the **third.txt** file

First.txt (existing file):

```
First file data!
```

Second.txt (existing file):

```
Second file data!
```

Third.txt (new file):

```
First file data!

Second file data!
```

Code : (source code link :):

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fptr1,*fptr2,*fptr3;
    fptr1 = fopen("first.txt","r");
    fptr2 = fopen("second.txt","r");
    if(fptr1!=NULL && fptr2!= NULL)
    {
        fptr3 = fopen("third.txt","w");
        else
        {
            char c=fgetc(fptr1);
            while(c!=EOF)
            {
                fputc(c,fptr3);
                c = fgetc(fptr1);
            }
            c = fgetc(fptr2);
            while(c!=EOF)
            {
                fputc(c,fptr3);
                c = fgetc(fptr2);
            }
            printf("Two files merged successfully into the third File!\n");
            fclose(fptr1); fclose(fptr2); fclose(fptr3);
        }
    }
    else
        printf("Failed to open existing files!\n");
    return 0;
}
```