1. Write a program to find entered number is even or odd

#include <stdio.h>

void main()

{

    int no;

    printf("Enter Number: ");

    scanf("%d", &no);

    if (no % 2 == 0)

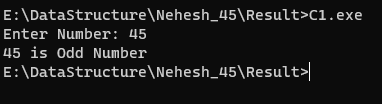
        printf("%d is Even Number", no);

    else

        printf("%d is Odd Number", no);

}

Output:



2. Write a program to find area of triangle

#include <stdio.h>

void main()

{

    int h, b;

    float ar;

    printf("Enter Base Length: ");

    scanf("%d", &b);

    printf("Enter Height: ");

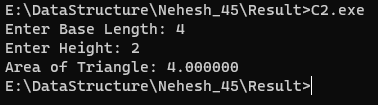
    scanf("%d", &h);

    ar = (0.5 \* b \* h);

    printf("Area of Triangle: %f", ar);

}

Output:



3. Write a program to find area of circle

#include <stdio.h>

void main()

{

    int r;

    float ar;

    printf("Enter Radius: ");

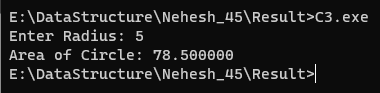
    scanf("%d", &r);

    ar = (3.14 \* r \* r);

    printf("Area of Circle: %f", ar);

}

Output:



4. Write a program to find the largest number among three entered number

#include <stdio.h>

void main()

{

    int n1, n2, n3;

    printf("Enter Three Number:\n");

    scanf("%d %d %d", &n1, &n2, &n3);

    if((n1 >= n2) && (n1 >= n3))

        printf("%d is the Greatest Number", n1);

    else if((n2 >= n1) && (n2 >= n3))

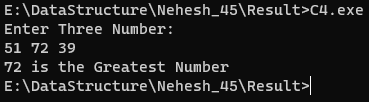
        printf("%d is the Greatest Number", n2);

    else if((n3 >= n2) && (n3 >= n1))

        printf("%d is the Greatest Number", n3);

}

Output:



5. Write a program to find the smallest number among three entered number

#include <stdio.h>

void main()

{

    int n1, n2, n3;

    printf("Enter Three Number:\n");

    scanf("%d %d %d", &n1, &n2, &n3);

    if((n1 <= n2) && (n1 <= n3))

        printf("%d is the Smallest Number", n1);

    else if((n2 <= n1) && (n2 <= n3))

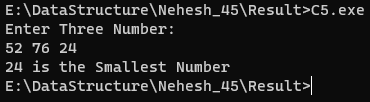
        printf("%d is the Smallest Number", n2);

    else if((n3 <= n2) && (n3 <= n1))

        printf("%d is the Smallest Number", n3);

}

Output:



6. Write a program to interchange the value of two variable using third variable

#include <stdio.h>

void main()

{

    int a, b, c;

    printf("Enter A: ");

    scanf("%d", &a);

    printf("Enter B: ");

    scanf("%d", &b);

    printf("Before Swap:\n");

    printf("A = %d \t B = %d\n", a, b);

    c = a;

    a = b;

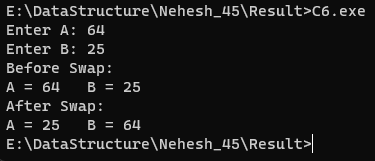
    b = c;

    printf("After Swap:\n");

    printf("A = %d \t B = %d", a, b);

}

Output:



7. Write a program to convert the temperature entered in Celsius to Fahrenheit ( (temp \*9/5)+32)

#include <stdio.h>

void main()

{

    float c, f;

    printf("Enter Temperature in Celcius: ");

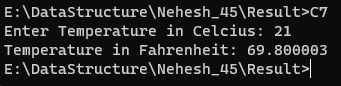
    scanf("%f", &c);

    f = (c \* 1.8) + 32;

    printf("Temperature in Fahrenheit: %f", f);

}

Output:



8. Write a program to interchange the value of two variable without using third variable

#include <stdio.h>

void main()

{

    int a, b;

    printf("Enter A: ");

    scanf("%d", &a);

    printf("Enter B: ");

    scanf("%d", &b);

    printf("Before Swap:\n");

    printf("A = %d \t B = %d\n", a, b);

    a = a + b;

    b = a - b;

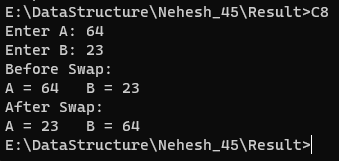
    a = a - b;

    printf("After Swap:\n");

    printf("A = %d \t B = %d", a, b);

}

Output:



9. Write a program to find series of even and odd numbers till the Nth element

#include <stdio.h>

void main()

{

    int no, i;

    printf("Enter Number: ");

    scanf("%d", &no);

    printf("Even Numbers \tOdd Numbers\n");

    for(i = 0; i <= no; i++)

    {

        if (i % 2 == 0)

            printf("%d\t\t", i);

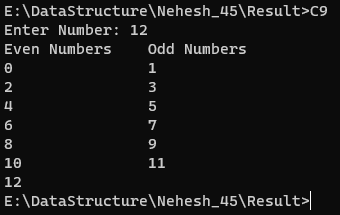
        else

            printf("%d\n", i);

    }

}

Output:



10. Write a program to find factorial of entered number

#include <stdio.h>

void main()

{

    int no, i, fact = 1;

    printf("Enter Number: ");

    scanf("%d", &no);

    for(i = 1; i <= no; i++)

    {

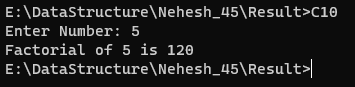
        fact = fact \* i;

    }

    printf("Factorial of %d is %d", no, fact);

}

Output:



11. Write a program to find series of factorial number till the Nth element

#include <stdio.h>

void main()

{

    int no, i, fact, j;

    printf("Enter Number: ");

    scanf("%d", &no);

    for(i = 0; i <= no; i++)

    {

        fact = 1;

        for(j = 1; j <= i; j++)

        {

            fact = fact \* j;

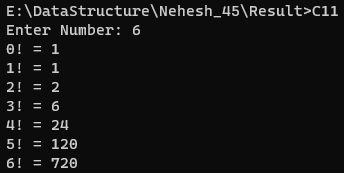
        }

        printf("%d! = %d\n", i, fact);

    }

}

Output:



12. Write a program to find entered number is prime or not

#include <stdio.h>

void main()

{

    int no, i;

    printf("Enter Number: ");

    scanf("%d", &no);

    for(i = 2; i < no; i++)

    {

        if (no % i == 0)

        {

            printf("%d is Composite Number", no);

            return;

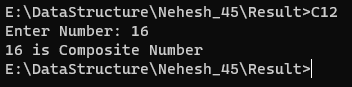
        }

    }

    printf("%d is Prime Number", no);

}

Output:



13. Write a program to find series of prime number till the Nth element

#include <stdio.h>

void main()

{

    int no, i, j, flag;

    printf("Enter Number: ");

    scanf("%d", &no);

    printf("List of Prime Numbers:\n");

    for(i = 2; i <= no; i++)

    {

    flag = 0;

    j = 2;

    do

    {

        if(i != 2)

        {

            if (i % j == 0)

            {

                flag++;

                break;

            }

        }

        j++;

    } while(j < i);

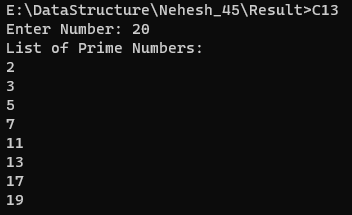
    if (flag == 0)

        printf("%d\n", i);

    }

}

Output:



14. Write a program to find entered number is Armstrong or not

#include <stdio.h>

void main()

{

    int no, tmp, rem, res = 0;

    printf("Enter Number: ");

    scanf("%d", &no);

    tmp = no;

    while(tmp != 0)

    {

        rem = tmp % 10;

        res = res + (rem \* rem \* rem);

        tmp = tmp / 10;

    }

    if (res == no)

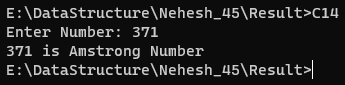
        printf("%d is Amstrong Number", no);

    else

        printf("%d is Not Amstrong Number", no);

}

Output:



15. Write a program to find series of Armstrong number till the Nth element

#include <stdio.h>

void main()

{

    int no, i, rem, tmp;

    long int res;

    printf("Enter Number: ");

    scanf("%d", &no);

    printf("List of Amstrong Numbers:\n");

    for (i = 0; i <= no; i++)

    {

        tmp = i;

        res = 0;

        while(tmp != 0)

        {

            rem = tmp % 10;

            res = res + (rem \* rem \* rem);

            tmp = tmp / 10;

        }

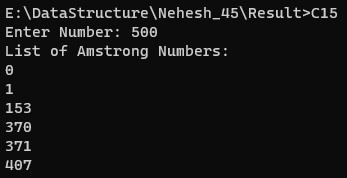
        if (res == i)

        printf("%d\n", i);

    }

}

Output:



16. Write a program to find sum of digit of entered number

#include <stdio.h>

void main()

{

    int no, res = 0, rem;

    printf("Enter Number: ");

    scanf("%d", &no);

    while(no != 0)

    {

        rem = no % 10;

        res = res + rem;

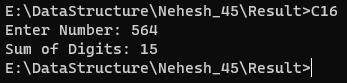
        no = no / 10;

    }

    printf("Sum of Digits: %d", res);

}

Output:



17. Write a program to find reveres of entered number

#include <stdio.h>

void main()

{

    int no, rev = 0, rem;

    printf("Enter Number: ");

    scanf("%d", &no);

    while(no != 0)

    {

        rem = no % 10;

        rev = rev \* 10 + rem;

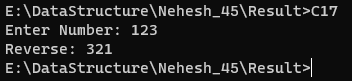
        no = no / 10;

    }

    printf("Reverse: %d", rev);

}

Output:



18. C Program to print Fibonacci series in a given range

#include <stdio.h>

void main()

{

    int i, n1 = 0, n2 = 1, n3, no;

    printf("Enter Number: ");

    scanf("%d", &no);

    for(i = 0; i < no; i++)

    {

        printf("%d\t", n1);

        n3 = n1 + n2;

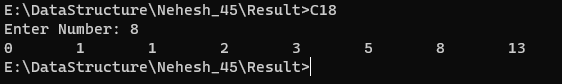
        n1 = n2;

        n2 = n3;

    }

}

Output:



19. C Program to check if given number is palindrome or not

#include <stdio.h>

void main()

{

    int no, rev = 0, rem, tmp;

    printf("Enter Number: ");

    scanf("%d", &no);

    tmp = no;

    while(no != 0)

    {

        rem = no % 10;

        rev = rev \* 10 + rem;

        no = no / 10;

    }

    if(tmp == rev)

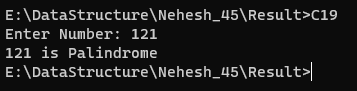
        printf("%d is Palindrome", tmp);

    else

        printf("%d is Not Palindrome", tmp);

}

Output:



20. Write a program to find series of palindrome till the Nth element

#include <stdio.h>

void chk\_palindrome(int x)

{

    int rev = 0, rem, tmp;

    tmp = x;

    while(x != 0)

    {

        rem = x % 10;

        rev = rev \* 10 + rem;

        x = x / 10;

    }

    if(tmp == rev)

        printf("%d\n", tmp);

}

void main()

{

    int no, i;

    printf("Enter Number: ");

    scanf("%d", &no);

    printf("List of Palindrome Elements:\n");

    for(i = 1; i <= no; i++)

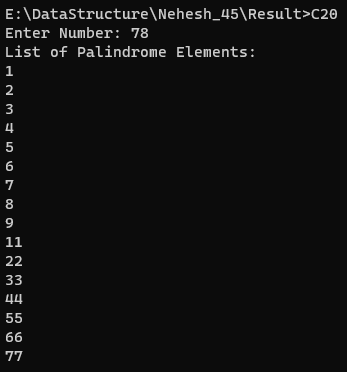
    {

        chk\_palindrome(i);

    }

}

Output:



21. Write A C Program to Print Multiplication Table

#include <stdio.h>

void main()

{

    int no, i;

    printf("Enter Number: ");

    scanf("%d", &no);

    for(i = 1; i <= 10; i++)

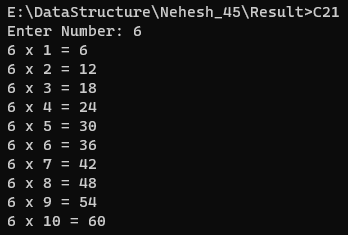
    {

        printf("%d x %d = %d\n", no, i, no\*i);

    }

}

Output:



22. C Program to Check whether an Alphabet is Vowel or Consonant

#include <stdio.h>

#include <ctype.h>

void main()

{

    char ch, c;

    printf("Enter Alphabet: ");

    scanf("%c", &ch);

    if(isalpha(ch))

    {

        c = tolower(ch);

        if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')

            printf("%c is a Vowel", ch);

        else

            printf("%c is a Consonant", ch);

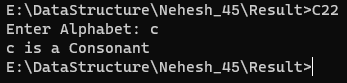
    }

    else

        printf("%c is not an Alphabet", ch);

}

Output:



23. Write a program to find GCD (Greatest common divisor)

#include <stdio.h>

void main()

{

    int n1, n2, i, gcd;

    printf("Enter Two Numbers:\n");

    scanf("%d%d", &n1, &n2);

    for(i = 1; i <= (n1>n2?n2:n1); i++)

    {

        if (n1 % i == 0 && n2 % i == 0)

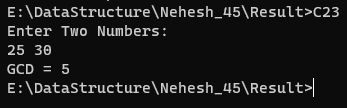
            gcd = i;

    }

    printf("GCD = %d", gcd);

}

Output:



24. Write a program to check the entered year is leap year or not

#include <stdio.h>

void main()

{

    int yr;

    printf("Enter Year: ");

    scanf("%d", &yr);

    if (yr % 400 == 0)

        printf("%d is Leap Year", yr);

    else if (yr % 100 == 0)

        printf("%d is Not Leap Year", yr);

    else if (yr % 4 == 0)

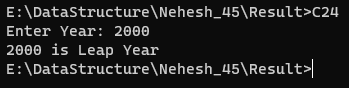
        printf("%d is Leap Year", yr);

    else

        printf("%d is Not Leap Year", yr);

}

Output:



25. C Program to find the Sum of First n Natural numbers

#include <stdio.h>

void main()

{

    int no, ans = 0;

    printf("Enter Number: ");

    scanf("%d", &no);

    while (no != 0)

    {

        ans += no;

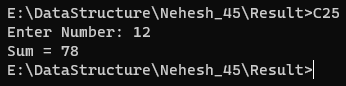
        no--;

    }

    printf("Sum = %d", ans);

}

Output:



26. Write a program to print the pyramid of stars

a. Mirrored Right Triangle Star Pattern

#include <stdio.h>

void main()

{

    int h, i, j;

    printf("Enter Height: ");

    scanf("%d", &h);

    for (i = 1; i <= h; i++)

    {

        for (j = 1; j <= h; j++)

        {

            if(j == 1 || j == i || i == h)

                printf("\*");

            else

                printf(" ");

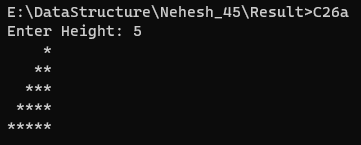
        }

        printf("\n");

    }

}

Output:



b. Hollow Right Triangle Star Pattern

#include <stdio.h>

void main()

{

    int h, i, j;

    printf("Enter Height: ");

    scanf("%d", &h);

    for (i = 0; i < h; i++)

    {

        for (j = 1; j <= h; j++)

        {

            if(j < (h-i))

                printf(" ");

            else

                printf("\*");

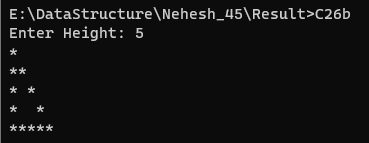
        }

        printf("\n");

    }

}

Output:



27. Write a program to print the following number pattern

#include <stdio.h>

void main()

{

    int h, i, j;

    printf("Enter Height: ");

    scanf("%d", &h);

    for (i = 1; i <= h; i++)

    {

        for (j = i; j <= h; j++)

        {

            printf("%d", j);

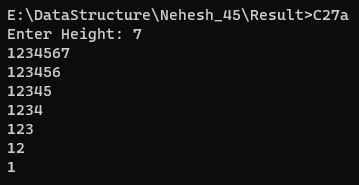
        }

        printf("\n");

    }

}

Output:



#include <stdio.h>

void main()

{

    int h, i, j;

    printf("Enter Height: ");

    scanf("%d", &h);

    for (i = h; i >= 0; i--)

    {

        for (j = 1; j <= i; j++)

        {

            printf("%d", j);

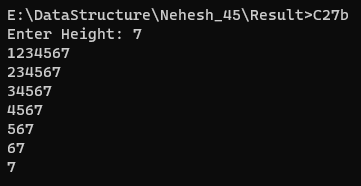
        }

        printf("\n");

    }

}

Output:



28. Write a program to find Net salary of employee. Basic salary entered by user. HRA is 40% basic, Da is 90% of basic, TA is 20 % of basic, PF is 25 % of basic.

#include <stdio.h>

void main()

{

    double salary, net, hra, da, ta, pf;

    printf("Enter Basic Salary:\n");

    scanf("%lf", &salary);

    hra = 40 \* salary / 100;

    da = 90 \* salary / 100;

    ta = 20 \* salary / 100;

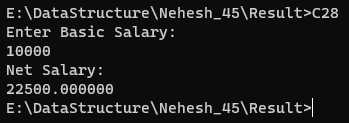
    pf = 25 \* salary / 100;

    net = (salary + hra + da + ta - pf);

    printf("Net Salary:\n%lf", net);

}

Output:



29. Write a menu driven program for 1. Check no is Even or odd number. 2.Check no is Prim or not 3. Check no is palindrome or not

#include <stdio.h>

void even\_odd()

{

    int no;

    printf("Enter Number: ");

    scanf("%d", &no);

    if (no % 2 == 0)

        printf("%d is Even Number\n", no);

    else

        printf("%d is Odd Number\n", no);

}

void chk\_prime()

{

    int i, no;

    printf("Enter Number: ");

    scanf("%d", &no);

    for(i = 2; i < no; i++)

    {

        if (no % i == 0)

        {

            printf("%d is Composite Number\n", no);

            return;

        }

    }

    printf("%d is Prime Number\n", no);

}

void palindrome()

{

    int no, rev = 0, rem, tmp;

    printf("Enter Number: ");

    scanf("%d", &no);

    tmp = no;

    while(no != 0)

    {

        rem = no % 10;

        rev = rev \* 10 + rem;

        no = no / 10;

    }

    if(tmp == rev)

        printf("%d is Palindrome\n", tmp);

    else

        printf("%d is Not Palindrome\n", tmp);

}

void main()

{

    char c;

    printf("Select Operation:\n 1. Check number is even or odd\n 2. Check number is prime or not\n 3. Check number is palindrome or not\n");

    printf("Operation: ");

    scanf("%c", &c);

    switch(c)

    {

        case '1':

            even\_odd();

            break;

        case '2':

            chk\_prime();

            break;

        case '3':

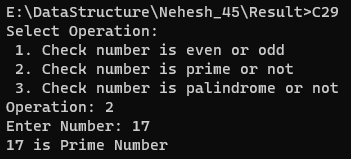
            palindrome();

            break;

    }

}

Output:



30. Write a menu driven program for 1. Print Pyramid of stars \* 2. Right triangle star pattern 3. Left or mirrored triangle star pattern 4. Hollow right triangle star pattern

#include <stdio.h>

void pyramid()

{

    int h, i, j;

    printf("Enter Height: ");

    scanf("%d", &h);

    for (i = 0; i < h; i++)

    {

        for (j = 1; j <= h; j++)

        {

            if(j < (h-i))

                printf(" ");

            else

                printf("\*");

        }

        for (j = 0; j < i; j++)

        {

            printf("\*");

        }

        printf("\n");

    }

}

void r\_triangle()

{

    int h, i, j;

    printf("Enter Height: ");

    scanf("%d", &h);

    for (i = 0; i <= h; i++)

    {

        for (j = 0; j < i; j++)

        {

            printf("\*");

        }

        printf("\n");

    }

}

void l\_triangle()

{

    int h, i, j;

    printf("Enter Height: ");

    scanf("%d", &h);

    for (i = 0; i < h; i++)

    {

        for (j = 1; j <= h; j++)

        {

            if(j < (h-i))

                printf(" ");

            else

                printf("\*");

        }

        printf("\n");

    }

}

void hr\_triangle()

{

    int h, i, j;

    printf("Enter Height: ");

    scanf("%d", &h);

    for (i = 1; i <= h; i++)

    {

        for (j = 1; j <= h; j++)

        {

            if(j == 1 || j == i || i == h)

                printf("\*");

            else

                printf(" ");

        }

        printf("\n");

    }

}

void main()

{

    char c;

    printf("Select Operation:\n 1. Print pyramid of start\n 2. Right triangle star pattern\n 3. Left or mirrored triangle star pattern\n 4. Hollow right triangle star pattern\n");

    printf("Operation: ");

    scanf("%c", &c);

    switch(c)

    {

        case '1':

            pyramid();

            break;

        case '2':

            r\_triangle();

            break;

        case '3':

            l\_triangle();

            break;

        case '4':

            hr\_triangle();

            break;

    }

}

                break;

            default:

                printf("Invalid choice! Please enter a valid option.\n");

                break;

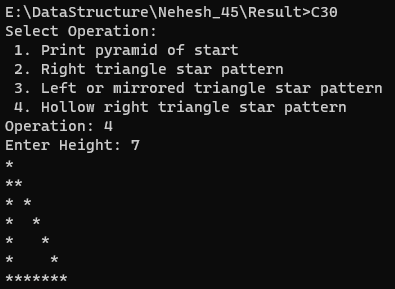
        }

    printf("\nSuyash Rusia Roll no.:58");

    return 0;

}

Output:



31.Write C program for to sort the array elements using bubble sort

#include <stdio.h>

#define MAX 5

void bubble\_sort(int students[], int len)

{

    int i, j, tmp;

    for(i = 0; i < (len - 1); i++)

    {

        for(j = 0; j < (len - i - 1); j++)

        {

            if(students[j] > students[j + 1])

            {

                tmp = students[j];

                students[j] = students[j + 1];

                students[j + 1] = tmp;

            }

        }

    }

}

int main()

{

    int record[MAX], i;

    printf("Enter 5 Elements to Sort:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

    }

    printf("\nInitial Array:\n");

    for (i = 0; i < MAX; i++)

        printf("%d\t", record[i]);

    printf("\n");

    bubble\_sort(record, MAX);

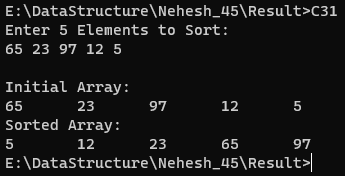
    printf("Sorted Array:\n");

    for (i = 0; i < MAX; i++)

        printf("%d\t", record[i]);

}

Output:



32. Write C program for to sort the array elements using selection sort

#include <stdio.h>

#define MAX 5

void selection\_sort(int students[], int len)

{

    int i, j, tmp;

    for(i = 0; i < len; i++)

    {

        for(j = i + 1; j < len; j++)

        {

            if(students[i] > students[j])

            {

                tmp = students[j];

                students[j] = students[i];

                students[i] = tmp;

            }

        }

    }

}

int main()

{

    int record[MAX], i;

    printf("Enter 5 Elements to Sort:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

    }

    printf("\nInitial Array:\n");

    for (i = 0; i < MAX; i++)

        printf("%d\t", record[i]);

    printf("\n");

    selection\_sort(record, MAX);

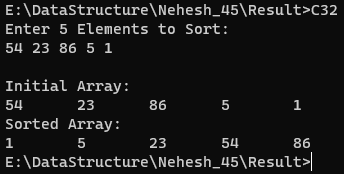
    printf("Sorted Array:\n");

    for (i = 0; i < MAX; i++)

        printf("%d\t", record[i]);

}

Output:



33. Write C program for to sort the array elements using insertion sort

#include <stdio.h>

#define MAX 5

void insertion\_sort(int students[], int len)

{

    int i, j, tmp;

    for(i = 1; i < len; i++)

    {

        for(j = 0; j < i; j++)

        {

            if(students[j] > students[i])

            {

                tmp = students[j];

                students[j] = students[i];

                students[i] = tmp;

            }

        }

    }

}

int main()

{

    int record[MAX], i;

    printf("Enter 5 Elements to Sort:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

    }

    printf("\nInitial Array:\n");

    for (i = 0; i < MAX; i++)

        printf("%d\t", record[i]);

    printf("\n");

    insertion\_sort(record, MAX);

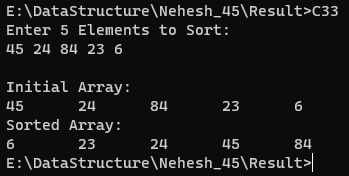
    printf("Sorted Array:\n");

    for (i = 0; i < MAX; i++)

        printf("%d\t", record[i]);

}

Output:



34. Write C program to find the largest element in the given array

#include <stdio.h>

#define MAX 5

int main()

{

    int record[MAX], i, greatest;

    printf("Enter 5 Elements:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

        if (i == 0)

            greatest = record[i];

        if (greatest < record[i])

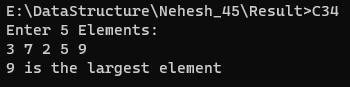
            greatest = record[i];

    }

    printf("%d is the largest element\n", greatest);

}

Output:



35. Write C program to find the smallest element in the given array

#include <stdio.h>

#define MAX 5

int main()

{

    int record[MAX], i, smallest;

    printf("Enter 5 Elements:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

        if (i == 0)

            smallest = record[i];

        if (smallest > record[i])

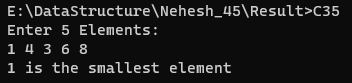
            smallest = record[i];

    }

    printf("%d is the smallest element\n", smallest);

}

Output:



36. Write C program to find the sum of all elements given in array

#include <stdio.h>

#define MAX 5

int main()

{

    int record[MAX], i, sum = 0;

    printf("Enter 5 Elements:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

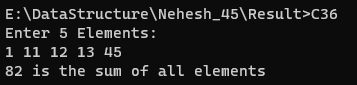
        sum += record[i];

    }

    printf("%d is the sum of all elements\n", sum);

}

Output:



37. Write C program to find number of elements are prime in given array

#include <stdio.h>

#define MAX 5

int chk\_prime(int no)

{

    int i;

    for(i = 2; i < no; i++)

    {

        if (no % i == 0)

            return -1;

    }

    return 0;

}

int main()

{

    int record[MAX], i, prime = 0;

    printf("Enter 5 Elements:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

        if(chk\_prime(record[i]) == 0)

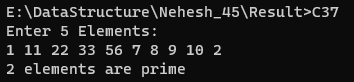
            prime++;

    }

    printf("%d elements are prime\n", prime);

}

Output:



38. Write C program to find number of elements are even and odd in the given array

#include <stdio.h>

#define MAX 10

int even\_odd(int no)

{

    if (no % 2 == 0)

        return 0;

    return 1;

}

int main()

{

    int record[MAX], i, even = 0, odd = 0;

    printf("Enter 10 Elements:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

        if(even\_odd(record[i]) == 0)

            even++;

        else if(even\_odd(record[i]) == 1)

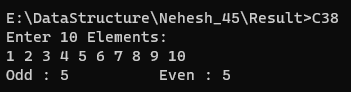
            odd++;

    }

    printf("Odd : %d \t Even : %d\n", odd, even);

}

Output:



39. Write C program to find number of elements are positive, negative and zero in the given array

#include <stdio.h>

#define MAX 10

int sign(int no)

{

    if (no > 0)

        return 1;

    else if (no < 0 )

        return -1;

    return 0;

}

int main()

{

    int record[MAX], i, positive = 0, negative = 0, zero = 0;

    printf("Enter 10 Elements:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

        if(sign(record[i]) == 0)

            zero++;

        else if(sign(record[i]) == 1)

            positive++;

        else if(sign(record[i]) == -1)

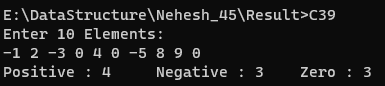
            negative++;

    }

    printf("Positive : %d \t Negative : %d\t Zero : %d\n", positive, negative, zero);

}

Output:



40. Write C program to find number of elements are palindrome in the given array

#include <stdio.h>

#define MAX 10

int palindrome(int no)

{

    int rev = 0, rem, tmp;

    tmp = no;

    while(no != 0)

    {

        rem = no % 10;

        rev = rev \* 10 + rem;

        no = no / 10;

    }

    if(tmp == rev)

        return 0;

    else

        return -1;

}

int main()

{

    int record[MAX], i, pal = 0;

    printf("Enter 10 Elements:\n");

    for(i = 0; i < MAX; i++)

    {

        scanf("%d", &record[i]);

        if(palindrome(record[i]) == 0)

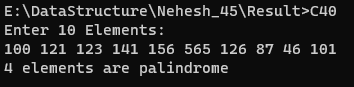
            pal++;

    }

    printf("%d elements are palindrome\n", pal);

}

Output:



41.Write C program to perform addition of two matrix

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j;

    int mat1[3][3], mat2[3][3], mat3[3][3];

    input(mat1);

    input(mat2);

    printf("Sum of Matrix:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

        {

            mat3[i][j] = mat2[i][j] + mat1[i][j];

            printf("%d\t", mat3[i][j]);

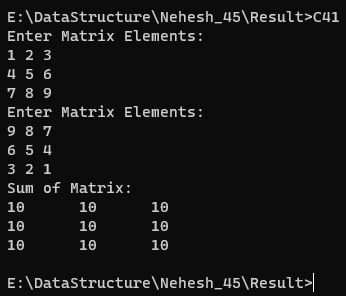
        }

        printf("\n");

    }

}

Output:



42.Write C program to perform subtraction of two matrix

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j;

    int mat1[3][3], mat2[3][3], mat3[3][3];

    input(mat1);

    input(mat2);

    printf("Difference of Matrix:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

        {

            mat3[i][j] = mat2[i][j] - mat1[i][j];

            printf("%d\t", mat3[i][j]);

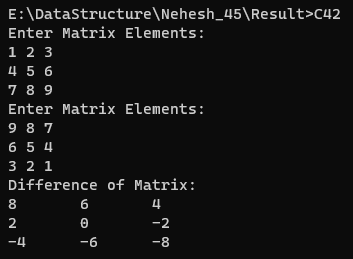
        }

        printf("\n");

    }

}

Output:



43.Write C program to perform multiplication of two matrix

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j, k;

    int mat1[3][3], mat2[3][3];

    int mat3[3][3] = {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}};

    input(mat1);

    input(mat2);

    printf("Product of Matrix:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

        {

            for(k = 0; k < 3; k++)

            {

                mat3[i][j] += mat1[i][k] \* mat2[k][j];

            }

            printf("%d\t", mat3[i][j]);

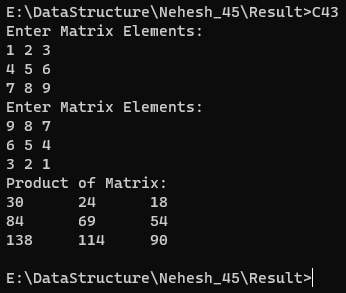
        }

        printf("\n");

    }

}

Output:



44. Write C program to find transpose of given matrix

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j;

    int mat1[3][3], mat2[3][3];

    input(mat1);

    printf("Transpose of Matrix:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

        {

            mat2[i][j] = mat1[j][i];

            printf("%d\t", mat2[i][j]);

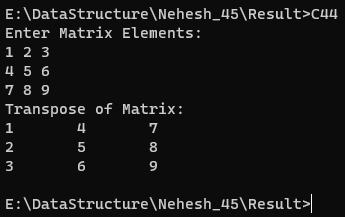
        }

        printf("\n");

    }

}

Output:



45.Write C program to print the diagonal elements of given matrix

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j;

    int mat1[3][3];

    input(mat1);

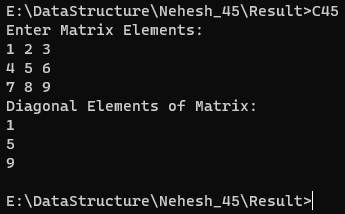
    printf("Diagonal Elements of Matrix:\n");

    for(i = 0; i < 3; i++)

        printf("%d\n", mat1[i][i]);

}

Output:



46.Write C program to print lower triangular of given matrix

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j, tmp;

    int mat1[3][3];

    input(mat1);

    printf("Lower Triangular Matrix:\n");

    for(i = 0; i < 3; i++)

    {

        for (j = 0; j < 3; j++)

        {

            if(j <= i)

                tmp = mat1[i][j];

            else

                tmp = 0;

            printf("%d\t", tmp);

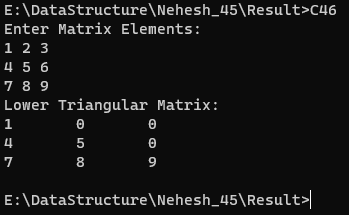
        }

        printf("\n");

    }

}

Output:



47.Write C program to print Upper triangular of given matrix

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j, tmp;

    int mat1[3][3];

    input(mat1);

    printf("Upper Triangular Matrix:\n");

    for(i = 0; i < 3; i++)

    {

        for (j = 0; j < 3; j++)

        {

            if(j >= i)

                tmp = mat1[i][j];

            else

                tmp = 0;

            printf("%d\t", tmp);

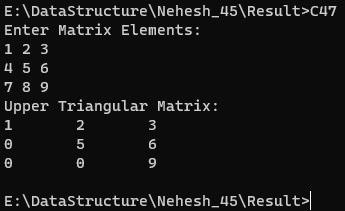
        }

        printf("\n");

    }

}

Output:



48.Write C program to find frequency of even number in the given matrix

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j, even = 0;

    int mat1[3][3];

    input(mat1);

    printf("Number of even elements: ");

    for(i = 0; i < 3; i++)

    {

        for (j = 0; j < 3; j++)

        {

            if(mat1[i][j] % 2 == 0)

                even++;

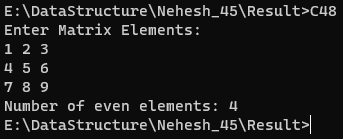
        }

    }

    printf("%d", even);

}

Output:



49. Write C program to find frequency of odd number in the given matrix

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j, odd = 0;

    int mat1[3][3];

    input(mat1);

    printf("Number of odd elements: ");

    for(i = 0; i < 3; i++)

    {

        for (j = 0; j < 3; j++)

        {

            if(mat1[i][j] % 2 != 0)

                odd++;

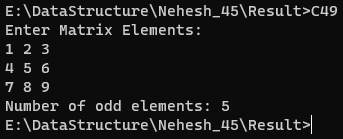
        }

    }

    printf("%d", odd);

}

Output:



50. Write C program to find the matrix are identical or not.

#include <stdio.h>

void input(int mat[3][3])

{

    int i, j;

    printf("Enter Matrix Elements:\n");

    for(i = 0; i < 3; i++)

    {

        for(j = 0; j < 3; j++)

            scanf("%d", &mat[i][j]);

    }

}

int main()

{

    int i, j;

    int mat1[3][3], mat2[3][3];

    input(mat1);

    input(mat2);

    for(i = 0; i < 3; i++)

    {

        for (j = 0; j < 3; j++)

        {

            if(mat1[i][j] != mat2[i][j])

            {

                printf("The matrix are not Identical");

                return 1;

            }

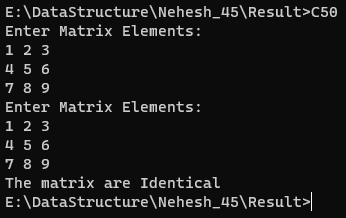
        }

    }

    printf("The matrix are Identical");

}

Output:



51. Write a C program to find length of a string

#include <stdio.h>

#include <string.h>

int main()

{

    char str[20];

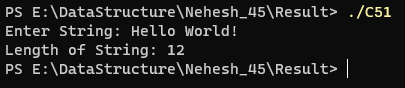
    printf("Enter String: ");

    gets(str);

    printf("Length of String: %d", strlen(str));

}

Output:



52. Write C program to copy one string to another string

#include <stdio.h>

#include <string.h>

int main()

{

    char str[20], res[20];

    printf("Enter String: ");

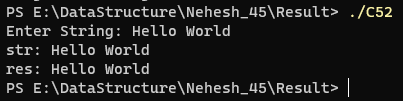
    gets(str);

    strcpy(res, str);

    printf("str: %s\nres: %s", str, res);

}

Output:



53. Write C program to concatenate two strings

#include <stdio.h>

#include <string.h>

int main()

{

    char str1[20], str2[20];

    printf("Enter String 1: ");

    gets(str1);

    printf("Enter String 2: ");

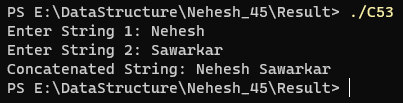
    gets(str2);

    strcat(str1, str2);

    printf("Concatenated String: %s", str1);

}

Output:



54. Write C program to compare two strings

#include <stdio.h>

#include <string.h>

int main()

{

    char str1[20], str2[20];

    int cmp;

    printf("Enter String 1: ");

    gets(str1);

    printf("Enter String 2: ");

    gets(str2);

    cmp = strcmp(str1, str2);

    if (cmp == 0)

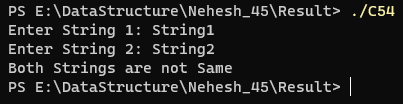
        printf("Both Strings are Same");

    else

        printf("Both Strings are not Same");

}

Output:



55. Write C program to convert lowercase string to uppercase

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20];

    int i;

    printf("Enter String: ");

    gets(str);

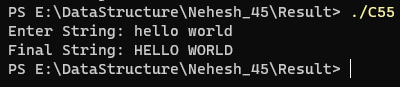
    for(i = 0; i < strlen(str); i++)

        str[i] = toupper(str[i]);

    printf("Final String: %s", str);

}

Output:



56. Write C program to convert uppercase string to lowercase

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20];

    int i;

    printf("Enter String: ");

    gets(str);

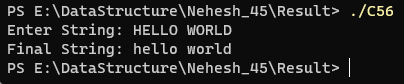
    for(i = 0; i < strlen(str); i++)

        str[i] = tolower(str[i]);

    printf("Final String: %s", str);

}

Output:



57. Write C program to toggle case of each character of a string

#include <stdio.h>

#include <string.h>

#include <ctype.h>

char toggle(char ch)

{

    if (isupper(ch))

        return(tolower(ch));

    else if(islower(ch))

        return(toupper(ch));

    else

        return ch;

}

int main()

{

    char str[20];

    int i;

    printf("Enter String: ");

    gets(str);

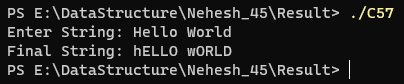
    for(i = 0; i < strlen(str); i++)

        str[i] = toggle(str[i]);

    printf("Final String: %s", str);

}

Output:



58. Write C program to find total number of alphabets, digits or special character in a string

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20];

    int i, digit = 0, alpha = 0, sc = 0;

    printf("Enter String: ");

    gets(str);

    for(i = 0; i < strlen(str); i++)

    {

        if(isalpha(str[i]))

            alpha++;

        else if(isalnum(str[i]))

            digit++;

        else

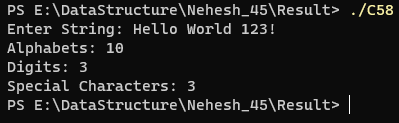
            sc++;

    }

    printf("Alphabets: %d\nDigits: %d\nSpecial Characters: %d", alpha, digit, sc);

}

Output:



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

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20];

    int i, vowels = 0, consonants = 0;

    printf("Enter String: ");

    gets(str);

    for(i = 0; i < strlen(str); i++)

    {

        str[i] = tolower(str[i]);

        if(isalpha(str[i]))

        {

            if(str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u')

                vowels++;

            else

                consonants++;

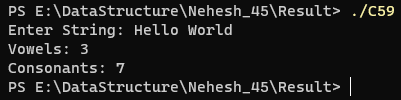
        }

    }

    printf("Vowels: %d\nConsonants: %d", vowels, consonants);

}

Output:



60. Write C program to count total number of words in a string

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20];

    int i, words = 0;

    printf("Enter String: ");

    gets(str);

    for(i = 0; i < strlen(str); i++)

    {

        if(str[i] == ' ')

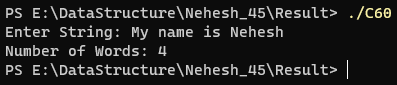
            words++;

    }

    printf("Number of Words: %d", words);

}

Output:



61. Write C program to find reverse of a string.

#include <stdio.h>

#include <string.h>

int main()

{

    char str[20];

    printf("Enter String: ");

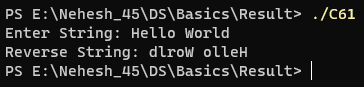
    gets(str);

    printf("Reverse String: ");

    puts(strrev(str));

}

Output:



62. Write C program to check whether a string is palindrome or not.

#include <stdio.h>

#include <string.h>

int main()

{

    char str[20], rev[20];

    printf("Enter String: ");

    gets(str);

    strcpy(rev, str);

    strrev(rev);

    if(strcmp(rev, str) == 0)

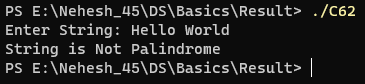
        printf("String is Palindrome");

    else

        printf("String is Not Palindrome");

}

Output:



63. Write C program to reverse order of words in a given string.

#include <stdio.h>

#include <string.h>

#define MAX 50

int main()

{

    char str[50], reverse[50];

    int len, i, index, start, end;

    printf("Enter string: ");

    gets(str);

    len   = strlen(str);

    index = 0;

    start = len - 1;

    end   = len - 1;

    while(start > 0)

    {

        if(str[start] == ' ')

        {

            for(i = start + 1; i <= end; i++)

            {

                reverse[index] = str[i];

                index++;

            }

            reverse[index++] = ' ';

            end = start - 1;

        }

        start--;

    }

    for(i = 0; i <= end; i++)

    {

        reverse[index] = str[i];

        index++;

    }

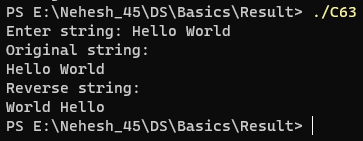
    reverse[index] = '\0';

    printf("Original string: \n%s\n", str);

    printf("Reverse string: \n%s", reverse);

}

Output:



64. Write C program to find first occurrence of a character in a given string.

#include <stdio.h>

#include <string.h>

int main()

{

    char ch, str[20];

    int i;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character: ");

    scanf("%c", &ch);

    for(i = 0; i < strlen(str); i++)

    {

        if(str[i] == ch)

        {

            printf("%c found at position %d", ch, i+1);

            return 0;

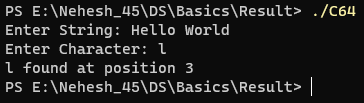
        }

    }

    printf("%c not found", ch);

}

Output:



65. Write C program to find last occurrence of a character in a given string.

#include <stdio.h>

#include <string.h>

int main()

{

    char ch, str[20];

    int i;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character: ");

    scanf("%c", &ch);

    for(i = strlen(str) - 1; i >= 0 ; i--)

    {

        if(str[i] == ch)

        {

            printf("%c found at position %d", ch, i+1);

            return 0;

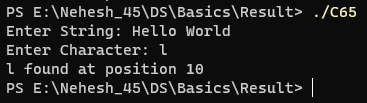
        }

    }

    printf("%c not found", ch);

}

Output:



66. Write C program to search all occurrences of a character in given string.

#include <stdio.h>

#include <string.h>

int main()

{

    char ch, str[20];

    int i, flag = 0;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character: ");

    scanf("%c", &ch);

    for(i = strlen(str) - 1; i >= 0 ; i--)

    {

        if(str[i] == ch)

        {

            printf("%c found at position %d\n", ch, i+1);

            flag = 1;

        }

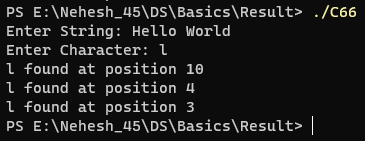
    }

    if(flag == 0)

        printf("%c not found", ch);

}

Output:



67. Write C program to count occurrences of a character in given string.

#include <stdio.h>

#include <string.h>

int main()

{

    char ch, str[20];

    int i, count = 0;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character: ");

    scanf("%c", &ch);

    for(i = strlen(str) - 1; i >= 0 ; i--)

    {

        if(str[i] == ch)

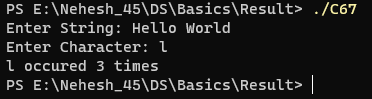
            count++;

    }

    printf("%c occured %d times", ch, count);

}

Output:



68. Write C program to find highest frequency character in a string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20];

    int i, alpha[26] = {0}, tmp, max;

    printf("Enter String: ");

    gets(str);

    for(i = 0; i <= strlen(str); i++)

    {

        tmp = tolower(str[i]);

        if(isalpha(str[i]))

            alpha[tmp - 97]++;

    }

    tmp = alpha[0];

    max = 0;

    for(i = 0; i < 26; i++)

    {

        if(alpha[i] > tmp)

        {

            tmp = alpha[i];

            max = i;

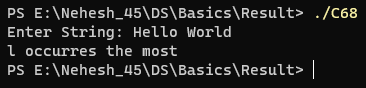
        }

    }

    printf("%c occurres the most", max+97);

}

Output:



69. Write C program to find lowest frequency character in a string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20];

    int i, alpha[26] = {0}, tmp, low;

    printf("Enter String: ");

    gets(str);

    for(i = 0; i <= strlen(str); i++)

    {

        tmp = tolower(str[i]);

        if(isalpha(str[i]))

            alpha[tmp - 97]++;

    }

    tmp = alpha[0];

    low = 0;

    for(i = 0; i < 26; i++)

    {

        if(alpha[i] < tmp && alpha[i] != 0)

        {

            tmp = alpha[i];

            low = i;

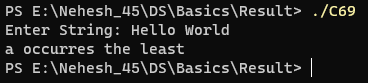
        }

    }

    printf("%c occurres the least", low+97);

}

Output:



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

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20];

    int i, alpha[26] = {0}, tmp;

    printf("Enter String: ");

    gets(str);

    for(i = 0; i <= strlen(str); i++)

    {

        tmp = tolower(str[i]);

        if(isalpha(str[i]))

            alpha[tmp - 97]++;

    }

    for(i = 0; i < 26; i++)

    {

        if(alpha[i] != 0)

        {

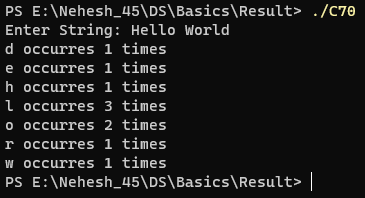
            printf("%c occurres %d times\n", i+97, alpha[i]);

        }

    }

}

Output:



71. Write C program to remove first occurrence of a character from string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20], ch;

    int i, flag = 0;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character: ");

    scanf("%c", &ch);

    for(i = 0; i <= strlen(str); i++)

    {

        if(tolower(str[i]) == tolower(ch) && flag == 0)

        {

            flag = 1;

            continue;

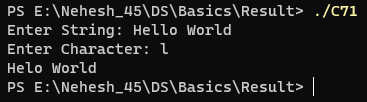
        }

        printf("%c", str[i]);

    }

}

Output:



72. Write C program to remove last occurrence of a character from string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20], ch, rev[20];

    int i, flag = 0;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character: ");

    scanf("%c", &ch);

    strcpy(rev, str);

    strrev(rev);

    for(i = 0; i <= strlen(rev); i++)

    {

        if(tolower(rev[i]) == tolower(ch) && flag == 0)

        {

            flag = strlen(str) - i - 1;

        }

    }

    for(i = 0; i <= strlen(str); i++)

    {

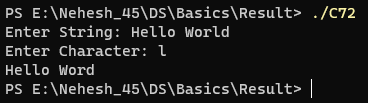
        if(i != flag)

            printf("%c", str[i]);

    }

}

Output:



73. Write C program to remove all occurrences of a character from string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20], ch;

    int i;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character: ");

    scanf("%c", &ch);

    for(i = 0; i <= strlen(str); i++)

    {

        if(tolower(str[i]) == tolower(ch))

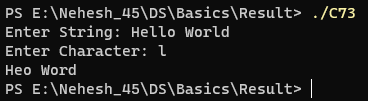
            continue;

        printf("%c", str[i]);

    }

}

Output:



74. Write C program to remove all repeated characters from a given string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20], cur, last;

    int i;

    printf("Enter String: ");

    gets(str);

    for(i = 0; i <= strlen(str); i++)

    {

        last = cur;

        cur = str[i];

        if(tolower(cur) == tolower(last))

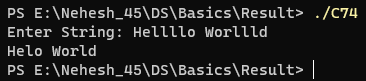
            continue;

        printf("%c", str[i]);

    }

}

Output:



75. Write C program to replace first occurrence of a character with another in a string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20], ch, rep;

    int i, flag = 0;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character and Replacement: ");

    scanf("%c %c", &ch, &rep);

    for(i = 0; i <= strlen(str); i++)

    {

        if(tolower(str[i]) == tolower(ch) && flag == 0)

        {

            flag = 1;

            str[i] = rep;

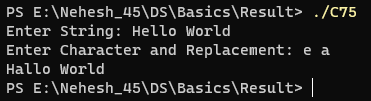
        }

        printf("%c", str[i]);

    }

}

Output:



76. Write C program to replace last occurrence of a character with another in a string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20], ch, rev[20], rep;

    int i, flag = 0;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character and Replacement: ");

    scanf("%c %c", &ch, &rep);

    strcpy(rev, str);

    strrev(rev);

    for(i = 0; i <= strlen(rev); i++)

    {

        if(tolower(rev[i]) == tolower(ch) && flag == 0)

        {

            flag = strlen(str) - i - 1;

            str[flag] = rep;

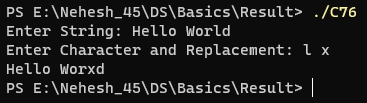
        }

    }

    puts(str);

}

Output:



77. Write C program to replace all occurrences of a character with another in a string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char str[20], ch, rep;

    int i;

    printf("Enter String: ");

    gets(str);

    printf("Enter Character and Replacement: ");

    scanf("%c %c", &ch, &rep);

    for(i = 0; i <= strlen(str); i++)

    {

        if(tolower(str[i]) == tolower(ch))

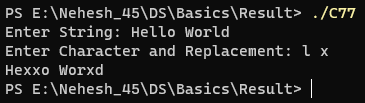
            str[i] = rep;

        printf("%c", str[i]);

    }

}

Output:



78. Write C program to find first occurrence of a word in a given string.

#include <stdio.h>

#include <string.h>

int find(const char \*str, const char \*word)

{

    int strLen = strlen(str);

    int wordLen = strlen(word);

    for (int i = 0; i <= strLen - wordLen; i++)

    {

        int j;

        for (j = 0; j < wordLen; j++)

        {

            if (str[i + j] != word[j])

                break;

        }

        if (j == wordLen)

            return i;

    }

    return -1;

}

int main()

{

    char str[50];

    char word[10];

    printf("Enter a string: ");

    gets(str);

    printf("Enter the word to find: ");

    scanf("%s", word);

    int result = find(str, word);

    if (result != -1)

        printf("First occurrence of '%s' found at index %d\n", word, result);

    else

        printf("Word not found in the string.\n");

    return 0;

}

Output:

79. Write C program to find last occurrence of a word in a given string

#include <stdio.h>

#include <string.h>

int find(const char \*str, const char \*word)

{

    int strLen = strlen(str);

    int wordLen = strlen(word);

    for (int i = strLen - wordLen; i >= 0; i--)

    {

        int j;

        for (j = 0; j < wordLen; j++)

        {

            if (str[i + j] != word[j])

                break;

        }

        if (j == wordLen)

            return i;

    }

    return -1;

}

int main()

{

    char str[100];

    char word[10];

    printf("Enter a string: ");

    gets(str);

    printf("Enter the word to find: ");

    scanf("%s", word);

    int result = find(str, word);

    if (result != -1)

        printf("Last occurrence of '%s' found at index %d\n", word, result);

    else

        printf("Word not found in the string.\n");

}

Output:

80. Write C program to search all occurrences of a word in given string.

#include <stdio.h>

#include <string.h>

void find(const char \*str, const char \*word)

{

    int strLen = strlen(str);

    int wordLen = strlen(word);

    int found = 0;

    for (int i = 0; i <= strLen - wordLen; i++)

    {

        int j;

        for (j = 0; j < wordLen; j++)

        {

            if (str[i + j] != word[j])

                break;

        }

        if (j == wordLen)

        {

            printf("Word found at index %d\n", i);

            found = 1;

        }

    }

    if (!found)

        printf("Word not found in the string.\n");

}

int main() {

    char str[50];

    char word[10];

    printf("Enter a string: ");

    gets(str);

    printf("Enter the word to find: ");

    scanf("%s", word);

    printf("Occurrences of '%s':\n", word);

    find(str, word);

}

Output:

81. Write C program to count occurrences of a word in a given string

#include <stdio.h>

#include <string.h>

int occur(const char \*str, const char \*word)

{

    int strLen = strlen(str);

    int wordLen = strlen(word);

    int count = 0;

    for (int i = 0; i <= strLen - wordLen; i++)

    {

        int j;

        for (j = 0; j < wordLen; j++)

        {

            if (str[i + j] != word[j])

                break;

        }

        if (j == wordLen)

            count++;

    }

    return count;

}

int main()

{

    char str[1000];

    char word[100];

    printf("Enter a string: ");

    gets(str);

    printf("Enter the word to count: ");

    scanf("%s", word);

    int result = occur(str, word);

    printf("Occurrences of '%s' in the string: %d\n", word, result);

}

Output:

82. Write C program to remove first occurrence of a word from string.

#include <stdio.h>

#include <string.h>

void rem(char \*str, const char \*word)

{

    int strLen = strlen(str);

    int wordLen = strlen(word);

    for (int i = 0; i <= strLen - wordLen; i++)

    {

        int j;

        for (j = 0; j < wordLen; j++)

        {

            if (str[i + j] != word[j])

                break;

        }

        if (j == wordLen)

        {

            for (int k = i; k < strLen - wordLen; k++)

                str[k] = str[k + wordLen];

            str[strLen - wordLen] = '\0';

            return;

        }

    }

}

int main()

{

    char str[50];

    char word[10];

    printf("Enter a string: ");

    gets(str);

    printf("Enter the word to remove: ");

    scanf("%s", word);

    rem(str, word);

    printf("%s\n", str);

}

Output:

83. Write C program to remove last occurrence of a word in given string.

#include <stdio.h>

#include <string.h>

void rem(char \*str, const char \*word)

{

    int strLen = strlen(str);

    int wordLen = strlen(word);

    for (int i = strLen - wordLen; i >= 0; i--)

    {

        int j;

        for (j = 0; j < wordLen; j++)

        {

            if (str[i + j] != word[j])

                break;

        }

        if (j == wordLen)

        {

            for (int k = i; k < strLen - wordLen; k++)

                str[k] = str[k + wordLen];

            str[strLen - wordLen] = '\0';

            return;

        }

    }

}

int main()

{

    char str[50];

    char word[10];

    printf("Enter a string: ");

    gets(str);

    printf("Enter the word to remove: ");

    scanf("%s", word);

    rem(str, word);

    printf("%s\n", str);

}

Output:

84. Write C program to remove all occurrence of a word in given string.

#include <stdio.h>

#include <string.h>

void rem(char str[], char word[])

{

    char \*ptr = strstr(str, word);

    while (ptr != NULL)

    {

        memmove(ptr, ptr + strlen(word), strlen(ptr + strlen(word)) + 1);

        ptr = strstr(ptr, word);

    }

}

int main()

{

    char input[50], word[10];

    printf("Enter a string: ");

    gets(input);

    printf("Enter the word to remove: ");

    gets(word);

    rem(input, word);

    printf("%s\n", input);

}

Output:

85. Write C program to trim leading white space characters from given string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void trim(char str[])

{

    int i = 0;

    while (isspace(str[i]))

    {

        i++;

    }

    memmove(str, str + i, strlen(str + i) + 1);

}

int main()

{

    char input[100];

    printf("Enter a string: ");

    gets(input);

    trim(input);

    printf("%s\n", input);

}

Output:

86. Write C program to trim trailing white space characters from given string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void trim(char str[])

{

    int i = strlen(str) - 1;

    while (i >= 0 && isspace(str[i]))

    {

        i--;

    }

    str[i + 1] = '\0';

}

int main()

{

    char input[50];

    printf("Enter a string: ");

    gets(input);

    trim(input);

    printf("%s\n", input);

}

Output:

87. Write C program to trim both leading and trailing white space characters from given string.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void trim(char str[])

{

    int i = 0, j = strlen(str) - 1;

    while (isspace(str[i]))

    {

        i++;

    }

    while (j >= 0 && isspace(str[j]))

    {

        j--;

    }

    memmove(str, str + i, j - i + 2);

}

int main()

{

    char input[50];

    printf("Enter a string: ");

    gets(input);

    trim(input);

    printf("%s\n", input);

}

Output:

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

#include <stdio.h>

#include <string.h>

void remspace(char str[])

{

    int i, j;

    for (i = 0, j = 0; str[i] != '\0'; i++)

    {

        if (str[i] != ' ' || (i > 0 && str[i - 1] != ' '))

            str[j++] = str[i];

    }

    str[j] = '\0';

}

int main()

{

    char input[50];

    printf("Enter a string: ");

    gets(input);

    remspace(input);

    printf("%s\n", input);

}

Output:

89. Write a C program to calculate electricity bill for the current month using function.

#include <stdio.h>

float bill(float units)

{

    float total;

    if (units <= 50)

        total = units \* 0.50;

    else if (units <= 150)

        total = 50 \* 0.50 + (units - 50) \* 0.75;

    else if (units <= 250)

        total = 50 \* 0.50 + 100 \* 0.75 + (units - 150) \* 1.20;

    else

        total = 50 \* 0.50 + 100 \* 0.75 + 100 \* 1.20 + (units - 250) \* 1.50;

    return total;

}

int main()

{

    float units;

    printf("Enter the number of units consumed: ");

    scanf("%f", &units);

    printf("Electricity Bill: %.2f\n", bill(units));

}

Output:

90. Write a program using function to calculate Net salary of employee. Basic salary entered by user. HRA is 40% basic, Da is 90% of basic, TA is 20 % of basic, PF is 25 % of basic.

#include <stdio.h>

float salary(float ctc)

{

    float hra, da, ta, pf, ihs;

    hra = 0.4 \* ctc;

    da = 0.9 \* ctc;

    ta = 0.2 \* ctc;

    pf = 0.25 \* ctc;

    ihs = ctc + hra + da + ta - pf;

    return ihs;

}

int main()

{

    float ctc;

    printf("Enter the basic salary: $");

    scanf("%f", &ctc);

    printf("Net Salary: %.2f\n", salary(ctc));

}

Output:

91. Write a program to find Fibonacci series using function.

#include <stdio.h>

void fibb(int n)

{

    int a = 0, b = 1, nxt;

    printf("Fibonacci Series: ");

    for (int i = 1; i <= n; i++)

    {

        printf("%d, ", a);

        nxt = a + b;

        a = b;

        b = nxt;

    }

    printf("\n");

}

int main()

{

    int n;

    printf("Enter the number of terms: ");

    scanf("%d", &n);

    fibb(n);

}

Output:

92. Write a program to find factorial of entered number using function

#include <stdio.h>

int fact(int n)

{

    if (n == 0 || n == 1)

        return 1;

    else

        return n \* fact(n - 1);

}

int main()

{

    int num;

    printf("Enter a number: ");

    scanf("%d", &num);

    printf("Factorial: %d\n", fact(num));

}

Output:

93. Write a program to swap the value of two variables using call by values and call by reference

#include <stdio.h>

void ref(int \*a, int \*b)

{

    int temp = \*a;

    \*a = \*b;

    \*b = temp;

}

int main()

{

    int x, y;

    printf("Enter two values: ");

    scanf("%d %d", &x, &y);

    int tmp = x;

    x = y;

    y = tmp;

    printf("Values after swapping (Value): x = %d, y = %d\n", x, y);

    ref(&y, &x);

    printf("Values after swapping (Reference): x = %d, y = %d\n", y, x);

}

Output:

94. C program to create, declare and initialize structure.

#include <stdio.h>

typedef struct{

    char name[10];

    int roll;

    float marks;

}student;

int main()

{

    student student1 = {"Nehesh", 45, 95};

    printf("Student Details:\n");

    printf("Name: %s\n", student1.name);

    printf("Roll Number: %d\n", student1.roll);

    printf("Marks: %.2f\n", student1.marks);

}

Output:

95. C program to read and print an employee's detail using structure

#include <stdio.h>

typedef struct {

    char name[10];

    int id;

    float salary;

}employee;

int main()

{

    employee employee1;

    printf("Enter employee details:\n");

    printf("Name: ");

    scanf("%s", employee1.name);

    printf("Employee ID: ");

    scanf("%d", &employee1.id);

    printf("Salary: ");

    scanf("%f", &employee1.salary);

    printf("\nEmployee Details:\n");

    printf("Name: %s\n", employee1.name);

    printf("Employee ID: %d\n", employee1.id);

    printf("Salary: %.2f\n", employee1.salary);

}

Output:

96. C program to demonstrate example of nested structure.

#include <stdio.h>

typedef struct{

    char city[50];

    char state[50];

}address;

typedef struct{

    char name[50];

    int id;

    float salary;

    address add;

}employee;

int main()

{

    employee employee1;

    printf("Enter employee details:\n");

    printf("Name: ");

    scanf("%s", employee1.name);

    printf("Employee ID: ");

    scanf("%d", &employee1.id);

    printf("Salary: ");

    scanf("%f", &employee1.salary);

    printf("City: ");

    scanf("%s", employee1.add.city);

    printf("State: ");

    scanf("%s", employee1.add.state);

    printf("\nEmployee Details:\n");

    printf("Name: %s\n", employee1.name);

    printf("Employee ID: %d\n", employee1.id);

    printf("Salary: %.2f\n", employee1.salary);

    printf("Address: %s, %s\n", employee1.add.city, employee1.add.state);

}

Output:

97. C program to demonstrate example structure pointer (structure with pointer)

#include <stdio.h>

typedef struct{

    float length;

    float width;

}rec;

int main()

{

    rec rectangle;

    rec \*ptrRectangle = &rectangle;

    printf("Enter the length and width of the rectangle: ");

    scanf("%f %f", &ptrRectangle->length, &ptrRectangle->width);

    printf("\nRectangle Details:\n");

    printf("Length: %.2f\n", ptrRectangle->length);

    printf("Width: %.2f\n", ptrRectangle->width);

}

Output:

98. C program to demonstrate example structure pointer (structure with pointer) using user define function.

#include <stdio.h>

typedef struct {

    float radius;

}circle;

void input(circle \*circ)

{

    printf("Enter the radius of the circle: ");

    scanf("%f", &circ->radius);

}

void display(circle \*circ)

{

    printf("Circle Details:\n");

    printf("Radius: %.2f\n", circ->radius);

}

int main()

{

    circle circ;

    circle \*ptrCircle = &circ;

    input(ptrCircle);

    display(ptrCircle);

}

Output:

99. C program to declare, initialize an union, example of union

#include <stdio.h>

typedef union{

    char model[20];

    int year;

    float price;

}car;

int main()

{

    car car1;

    printf("Enter car details:\n");

    printf("Model: ");

    scanf("%s", car1.model);

    printf("Year: ");

    scanf("%d", &car1.year);

    printf("Price: ");

    scanf("%f", &car1.price);

    printf("\nCar Details:\n");

    printf("Model: %s\n", car1.model);

    printf("Year: %d\n", car1.year);

    printf("Price: %.2f\n", car1.price);

}

Output:

100. C program to demonstrate example of structure of array.

#include <stdio.h>

typedef struct{

    char name[50];

    int scores[5];

}student;

int main()

{

    student students[3];

    for (int i = 0; i < 3; i++)

    {

        printf("Enter name of student %d: ", i + 1);

        scanf("%s", students[i].name);

        printf("Enter scores for 5 subjects for %s:\n", students[i].name);

        for (int j = 0; j < 5; j++)

        {

            printf("Subject %d: ", j + 1);

            scanf("%d", &students[i].scores[j]);

        }

    }

    printf("\nStudent Information:\n");

    for (int i = 0; i < 3; i++)

    {

        printf("\nStudent %d\n", i + 1);

        printf("Name: %s\n", students[i].name);

        printf("Scores: ");

        for (int j = 0; j < 5; j++)

            printf("%d ", students[i].scores[j]);

        printf("\n");

    }

}

Output:

101. C program to add two distances in feet and inches using structure

#include <stdio.h>

typedef struct{

    int feet;

    int inches;

}dist;

dist add(dist d1, dist d2)

{

    dist result;

    result.feet = d1.feet + d2.feet;

    result.inches = d1.inches + d2.inches;

    if (result.inches >= 12) {

        result.feet += result.inches / 12;

        result.inches %= 12;

    }

    return result;

}

int main()

{

    dist d1, d2, sum;

    printf("Enter distance 1\n");

    printf("Feet: ");

    scanf("%d", &d1.feet);

    printf("Inches: ");

    scanf("%d", &d1.inches);

    printf("Enter distance 2\n");

    printf("Feet: ");

    scanf("%d", &d2.feet);

    printf("Inches: ");

    scanf("%d", &d2.inches);

    sum = add(d1, d2);

    printf("\nSum of distances: %d feet %d inches\n", sum.feet, sum.inches);

}

Output:

102. C program to extract individual bytes from an unsigned int using union.

#include <stdio.h>

typedef union{

    unsigned int num;

    unsigned char bytes[4];

}extract;

int main()

{

    extract extractor;

    printf("Enter an unsigned integer: ");

    scanf("%u", &extractor.num);

    printf("\nIndividual bytes of the unsigned integer %u are:\n", extractor.num);

    for (int i = 0; i < 4; i++) {

        printf("Byte %d: %u\n", i + 1, extractor.bytes[i]);

    }

}

Output:

103. C program for passing structures as function arguments and returning a structure from a function.

#include <stdio.h>

typedef struct{

    int feet;

    int inches;

}dist;

dist add(dist d1, dist d2)

{

    dist result;

    result.feet = d1.feet + d2.feet;

    result.inches = d1.inches + d2.inches;

    if (result.inches >= 12) {

        result.feet += result.inches / 12;

        result.inches %= 12;

    }

    return result;

}

int main()

{

    dist d1, d2, sum;

    printf("Enter distance 1\n");

    printf("Feet: ");

    scanf("%d", &d1.feet);

    printf("Inches: ");

    scanf("%d", &d1.inches);

    printf("Enter distance 2\n");

    printf("Feet: ");

    scanf("%d", &d2.feet);

    printf("Inches: ");

    scanf("%d", &d2.inches);

    sum = add(d1, d2);

    printf("\nSum of distances: %d feet %d inches\n", sum.feet, sum.inches);

}

Output:

104. Calculate party expenses using C program

#include <stdio.h>

typedef struct{

    char name[50];

    float price;

}item;

int main()

{

    int numFriends;

    printf("Enter the number of friends: ");

    scanf("%d", &numFriends);

    if (numFriends <= 0)

    {

        printf("Invalid number of friends. Please enter a positive value.\n");

        return -1;

    }

    int numItems;

    printf("Enter the number of items used in the party: ");

    scanf("%d", &numItems);

    float totalExpenses = 0.0;

    item items[numItems];

    printf("Enter details for each item:\n");

    for (int i = 0; i < numItems; i++)

    {

        printf("Item %d:\n", i + 1);

        printf("Name: ");

        scanf("%s", items[i].name);

        printf("Price: $");

        scanf("%f", &items[i].price);

        totalExpenses += items[i].price;

    }

    float expensesPerFriend = totalExpenses / numFriends;

    printf("\nTotal expenses: $%.2f\n", totalExpenses);

    printf("Expenses per friend: $%.2f\n", expensesPerFriend);

}

Output:

105. C program to find the size of the union

#include <stdio.h>

typedef union {

    int integer;

    float floatingPoint;

    char character;

}sample;

int main()

{

    sample myUnion;

    printf("Size of the union: %lu bytes\n", sizeof(myUnion));

}

Output:

106. C program to find number of lines in a file.

Output:

107. C program to create, open and close a file.

Output:

108. C program to write text (characters) into file and print

Output:

109. C program to print given number of lines of a file (like head command in Linux).

Output:

110. C program to print contents in reverse order of a file (just like TAC command in Linux)

Output:

111. C program to compare contents of two files.

Output:

112. C program to copy number of bytes of from a specific offset to another file.

Output:

113. C program to read Content of a File using getc() using C Program.

Output:

114. C program to convert All Characters in Upper Case of a File using C Program.

Output:

115. C program to delete a specified file using remove() function

Output: