LAB FILE

Introduction to C Programming



Batch 2023-26

Section-D BCA Hon.(AI&DS)

Submitted by: Submitted to:

Nishit Agarwal Mr Rishi Kumar

2316012120 Ass. Prof (CSIT)

GEU

INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | PROJECT NAME | PAGE NO. | FACULTY SIGN | REMARKS |
| 1 | **WAP for hello world or this is my first C Program.** |  |  |  |
| 2 | **WAP to add two numbers.** |  |  |  |
| 3 | **WAP to find area of circle.** |  |  |  |
| 4 | **WAP to divide two numbers** |  |  |  |
| 5 | **WAP to print ASCII value.** |  |  |  |
| 6 | **WAP to multiply floating point numbers** |  |  |  |
| 7 | **WAP to SWAP two vairables number by using third variable.** |  |  |  |
| 8 | **WAP to SWAP two vairables number without using third variable** |  |  |  |
| 9 | **WAP to SWAP three vairable numbers without using third variables.** |  |  |  |
| 10 | **Wap to find the area of rectangle** |  |  |  |
| 11 | **WAP to find area of square** |  |  |  |
| 12 | **wap to find area of right angle triangle, isosceles triangle, any triangle with 3 sides** |  |  |  |
| 13 | **wap to find Area and Volume of Cube.** |  |  |  |
| 14 | **wap to find area and volume of cuboid.** |  |  |  |
| 15 | **WAP to find the largest number using the Logical AND operator** |  |  |  |
| 16 | **WAP to validate the username and password entered by the user is correct or not using the predefined username and password** |  |  |  |
| 17 | **WAP to input the positive number from the user to perform the Left shift operator** |  |  |  |
| 18 | **WAP to input the positive number from the user to perform the Right shift operator** |  |  |  |
| 19 | **WAP to perform the pre increment and pre decrement operator on two integers and print both original value and updated value** |  |  |  |
| 20 | **WAP to perform the post increment and post decrement operator on two integers and print both original value and updated value.** |  |  |  |
| 21 | **WAP for an integer number and to check whether it is divisible by 9 or 7 using OR logical operator** |  |  |  |
| 22 | **WAP to identify gender in single character and print full gender (Ex: if input is 'M' or 'm' – it should print "Male").** |  |  |  |
| 23 | **Write a C program to print all natural numbers in reverse (from n to 1).** |  |  |  |
| 24 | **Write a C program to print all alphabets from a to z.** |  |  |  |
| 25 | **Write a C program to print all natural numbers from 1 to n** |  |  |  |
| 26 | **Write a C program to print all natural numbers from 1 to n** |  |  |  |
| 27 | **Write a C program to print all odd number between 1 to 100** |  |  |  |
| 28 | **Write a C program to find sum of all natural numbers between 1 to n.** |  |  |  |
| 29 | Write a C program to find sum of all even numbers between 1 to n. |  |  |  |
| 30 | Write a C program to find sum of all odd numbers between 1 to n. |  |  |  |
| 31 | Write a C program to print multiplication table of any number |  |  |  |
| 32 | Write a C program to count number of digits in a number. |  |  |  |
| 33 | Write a C program to find first and last digit of a number |  |  |  |
| 34 | Write a C program to find sum of first and last digit of a number |  |  |  |
| 35 | Write a C program to swap first and last digits of a number |  |  |  |
| 36 |  |  |  |  |
| 37 |  |  |  |  |
| 38 |  |  |  |  |
| 39 |  |  |  |  |
| 40 |  |  |  |  |
| 41 |  |  |  |  |
| 42 |  |  |  |  |
| 43 |  |  |  |  |
| 44 |  |  |  |  |
| 45 |  |  |  |  |
| 46 |  |  |  |  |
| 47 |  |  |  |  |
| 48 |  |  |  |  |
| 49 |  |  |  |  |
| 50 |  |  |  |  |
| 51 |  |  |  |  |
| 52 |  |  |  |  |
| 53 |  |  |  |  |
| 54 |  |  |  |  |
| 55 |  |  |  |  |
| 56 |  |  |  |  |
| 57 |  |  |  |  |
| 58 |  |  |  |  |
| 59 |  |  |  |  |
| 60 |  |  |  |  |
| 61 |  |  |  |  |
| 62 |  |  |  |  |
| 63 |  |  |  |  |
| 64 |  |  |  |  |
| 65 |  |  |  |  |
| 66 |  |  |  |  |
| 67 |  |  |  |  |
| 68 |  |  |  |  |
| 69 |  |  |  |  |
| 70 |  |  |  |  |
| 71 |  |  |  |  |
| 72 |  |  |  |  |
| Pattern exercise-1 |  |  |  |  |
| Pattern exercise-2 |  |  |  |  |
| IF ELSE EXERCISES:-1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 or 14 |  |  |  |  |
| 13 or 15 |  |  |  |  |
| 16 |  |  |  |  |
| 17 |  |  |  |  |
| 18 |  |  |  |  |
| 19 |  |  |  |  |
| 20 |  |  |  |  |
| 21 |  |  |  |  |
| 22 |  |  |  |  |
| 23 |  |  |  |  |
| 24 |  |  |  |  |

**Program-1**

**WAP to print HELLO WORLD**

**Input**

#include<stdio.h>

int main()

{

    printf("NISHIT AGARWAL");

    printf("Hello World");

    return 0;

}

Output

C:\Users\abc\Pictures\Screenshot (3).png

**Program-2**

**Wap to add two numbers**

**Input**

#include <stdio.h>

int main()

{

    double num1, num2, sum;

    printf("nishit agarwal");

    printf("Enter the first number: ");

    scanf("%lf", &num1);

    printf("Enter the second number: ");

    scanf("%lf", &num2);

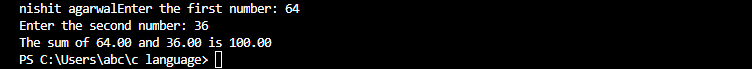
    sum = num1 + num2;

    printf("The sum of %.2lf and %.2lf is %.2lf\n", num1, num2, sum);

    return 0;

}

Output



**Program-3**

**Wap to find area of circle**

**Input**

#include<stdio.h>

int main()

{

printf("nishit agarwal");

float radius,areaofcircle;

printf("enter the radius of circle");

scanf("%f" ,&radius);

areaofcircle =3.14\*radius\*radius;

printf("area of circle with radius %0.2f is %0.2f\n",radius ,areaofcircle);

return 0;

}

Output

C:\Users\abc\Pictures\Screenshot (5).png

**Program-4**

**Wap to divide two numbers**

**Input**

**#include<stdio.h>;**

int main()

{

    float num1,num2,result;

    printf("nishit agarwal");

    printf("enter the first number");

    scanf("%f", &num1);

    printf("enter the second number");

    scanf("%f", &num2);

    if (num2!=0)

   result=num1/num2;

    else

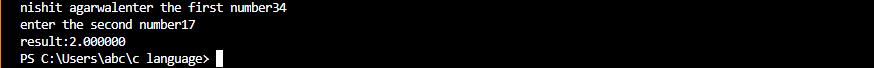
    printf("divide by zero is not allowed.\n");

    printf("result:%f\n", result);

    return 0;

}

Output



**Program-5**

**Wap to print ascii value**

**Input**

#include<stdio.h>;

int main(){

char ch;

printf("nishit agarwal");

printf("enter a character");

scanf("%c", &ch);

printf("asc2 value of %C is %d\n",ch,ch);

return 0;

}

Output

C:\Users\abc\Pictures\Screenshot (7).png

**Program-6**

**Wap to multiply floating point numbers**

**Inpu**t

#include<stdio.h>

int main()

{

float num1,num2,product;

printf("nishit agarwal");

printf("enter the first floating number");

scanf("%f", &num1);

printf("enter the second floating number");

scanf("%f", &num2);

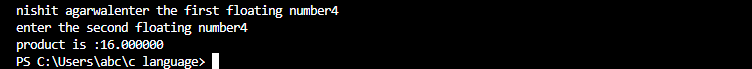
product =num1\*num2;

printf("product is :%f\n",product);

return 0;

}

Output



**Program-7**

**Wap to swap two variables number without using third variable**

**Input**

#include<stdio.h>

int main()

{

    int a,b;

    printf("nishit agarwal");

    printf("enter two numbers a and b \n");

    scanf("%d %d",&a,&b);

    printf("before swapping: a=%d,b=%d\n",a,b);

    a=a+b;

    b=a-b;

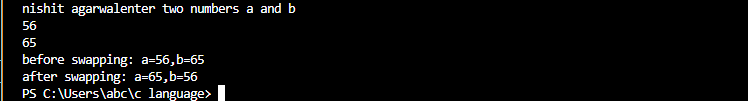
    a=a-b;

    printf("after swapping: a=%d,b=%d\n",a,b);

    return 0;

}

Output



**Program-8**

**Wap to swap two variables numbers by using third variable**

**Input**

#include<stdio.h>;

int main()

{

    float num1,num2,temp;

    printf("nishit agarwal");

    printf("enter the first number");

    scanf("%f", &num1);

    printf("enter the second number");

    scanf("%f", &num2);

    temp=num1;

    num1=num2;

    num2=temp;

    printf("after swapping:\n");

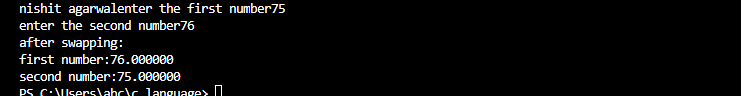
    printf("first number:%f\n",num1);

    printf("second number:%f\n",num2);

    return 0;

}

Output



**Program-9**

**WAP to find the area of rectangle**

#include<stdio.h>

int main()

{

    float length,width,area;

    printf("nishit agarwal");

    printf("enter the length of the rectangle");

    scanf("%f", &length);

    printf("enter the width of the rectangle");

    scanf("%f", &width);

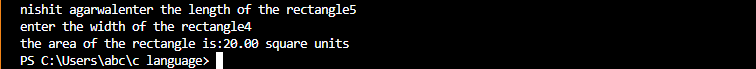
    area = length \* width;

    printf("the area of the rectangle is:%0.2f square units\n", area);

    return 0;

}

OUTPUT



**Program-10**

**Wap to find area of square**

**INPUT**

#include<stdio.h>

int main()

{

    float sidelength,area;

    printf("nishit agarwal");

    printf("enter the sidelength of the square");

    scanf("%f", &sidelength);

    area = sidelength \* sidelength;

    printf("the area of the square is:%0.2f square units\n", area);

    return 0;

}

OUTPUT

C:\Users\abc\Pictures\Screenshot (12).png

**Program-11**

**WAP to find area and volume of cube**

**INPUT**

#include<stdio.h>

int main()

{

    float sidelength;

    float area,volume;

    printf("nishit agarwal");

    printf("enter the side length");

    scanf("%f",&sidelength);

    area=6\*(sidelength\*sidelength);

    volume=sidelength\*sidelength\*sidelength;

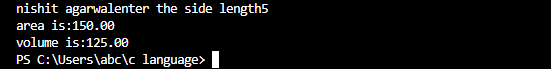
    printf("area is:%0.2f\n",area);

    printf("volume is:%0.2f\n",volume);

    return 0;

}

OUTPUT



**Program-12**

**WAP to find area and volume of cuboid**

**INPUT**

#include<stdio.h>

int main()

{

    float length,width,height;

    float area,volume;

    printf("nishit agarwal");

    printf("enter the length of the cuboid");

    scanf("%f",&length);

    printf("enter the width of the cuboid");

    scanf("%f",&width);

    printf("enter the height of the cuboid");

    scanf("%f",&height);

    area=2\*(length\*width+length\*height+height\*width);

    volume=length\*width\*height;

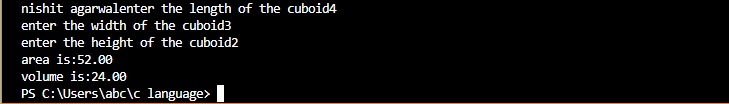
    printf("area is:%0.2f\n",area);

    printf("volume is:%0.2f\n",volume);

    return 0;

}

OUTPUT



**Program-13**

**Wap to swap three variable numbers without using third variables**

**INPUT**

**#include<stdio.h>**

int main()

{

    int a,b,c;

    printf("nishit agarwal");

    printf("enter three numbers a,b,c\n");

    scanf("%d %d %d", &a,&b,&c);

    printf("before swapping: a=%d,b=%d,c=%d\n",a,b,c);

    a=a+b;

    b=a-b;

    a=a-b;

    printf("after swapping: a=%d,b=%d,c=%d\n",a,b);

    return 0;

}

OUTPUT



**Program-14**

**Wap to find area of right angle triangle, isosceles triangle, any triangle with three sides**

**INPUT**

#include <stdio.h>

#include <math.h>

int main()

{

    int choice;

    printf("nishit agarwal");

    printf("Select the type of triangle:\n");

    printf("1. Right Angle Triangle\n");

    printf("2. Isosceles Triangle\n");

    printf("3. Any Triangle\n");

    scanf("%d", &choice);

    switch (choice) {

        case 1: {

            float base, height;

            printf("Enter the base length: ");

            scanf("%f", &base);

            printf("Enter the height length: ");

            scanf("%f", &height);

            float area = 0.5 \* base \* height;

            printf("Area of the right-angle triangle is: %.2f\n", area);

            break;

        }

        case 2: {

            float base, side;

            printf("Enter the base length: ");

            scanf("%f", &base);

            printf("Enter the side length: ");

            scanf("%f", &side);

            float area = 0.5 \* base \* sqrt(side \* side - (base \* base / 4));

            printf("Area of the isosceles triangle is: %.2f\n", area);

            break;

        }

        case 3: {

            float a, b, c;

            printf("Enter the lengths of the three sides of the triangle:\n");

            printf("Side 1: ");

            scanf("%f", &a);

            printf("Side 2: ");

            scanf("%f", &b);

            printf("Side 3: ");

            scanf("%f", &c);

            // Check if the sides form a valid triangle

            if (a + b > c && a + c > b && b + c > a) {

                float s = (a + b + c) / 2;

                float area = sqrt(s \* (s - a) \* (s - b) \* (s - c));

                printf("Area of the triangle is: %.2f\n", area);

            } else {

                printf("These sides cannot form a triangle.\n");

            }

            break;

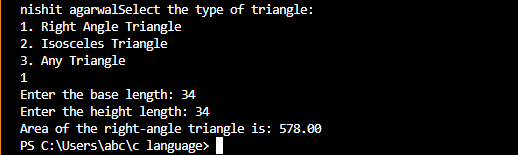
        }

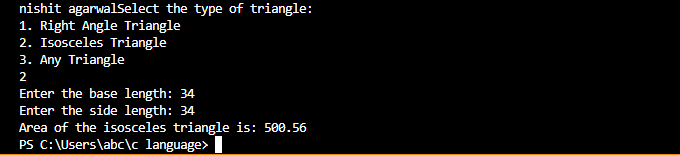
        default:

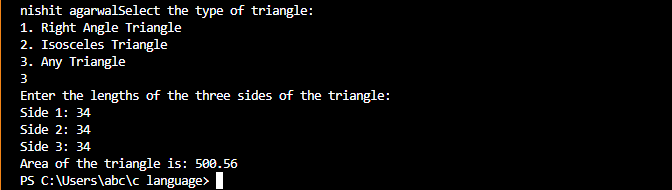
            printf("Invalid choice.\n");

                }    return 0;}

OUTPUT







**Program-15**

**Wap to find the largest number using the logical AND operator**

**INPUT**

#include<stdio.h>

int main()

{

    int a,b,c;

    printf("nishit agarwal");

    printf("enter three numbers:\n");

    scanf("%d", &a);

    printf("b:");

    scanf("%d", &b);

    printf("c:");

    scanf("%d", &c);

    if(a>b && a>c);

    printf("biggest number is %d",a);

     if(b>a && b>c);

    printf("biggest number is %d",a);

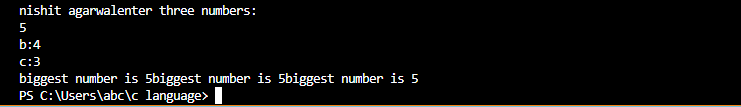
     if(c>b && c>a);

    printf("biggest number is %d",a);

    return 0;

}

OUTPUT



**Program-16**

**Wap to validate the username and password entered by the user is correct or not using the predefined username and password**

**INPUT**

#include<stdio.h>

int main()

{

    int pass, x=10;

    while(x!=0)

    {

        printf("nishit agarwal");

        printf("\ninput the password: ");

        scanf("%d", &pass);

        if(pass==1234)

        {

            printf("correct password");

            x = 0;

        }

        else

        {

            printf("wrong password, try another");

        }

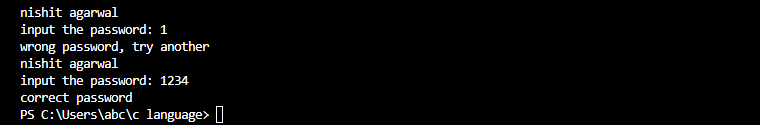
        printf("\n");

    }

    return 0;

}

OUTPUT



**Program-17**

**Wap to input the positive number from the user to perform the left shift operator**

**INPUT**

#include<stdio.h>

int main ()

{

    int num;

    printf("nishit agarwal");

    printf("enter a positive number: ");

    scanf("%d", &num);

    num=(num<<2);

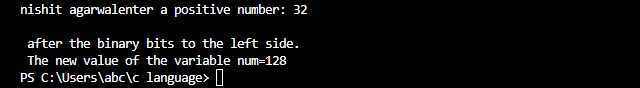
    printf("\n after the binary bits to the left side.");

    printf("\n The new value of the variable num=%d",num);

    return 0;

}

OUTPUT



**Program-18**

**Wap to input the positive number from the user to perform the right shift operator**

**INPUT**

#include<stdio.h>

int main ()

{

    int num;

    printf("nishit agarwal");

    printf("enter a positive number: ");

    scanf("%d", &num);

    num=(num>>2);

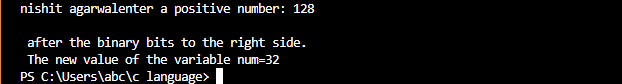
    printf("\n after the binary bits to the right side.");

    printf("\n The new value of the variable num=%d",num);

    return 0;

}

OUTPUT



**Program-19**

**Wap to perform the pre increment and pre decrement operator on two integers and print both original value and updated value**

**INPUT**

#include <stdio.h>

#include <conio.h>

int main ()

{

int a = 7;

int b = 2;

int x,y;

x = ++a;

y = ++b;

printf("nishit agarwal");

printf (" Use the pre-increment operator " );

printf (" The value of a is %d ", a);

printf (" The value of b is %d ", b);

printf (" After using the pre-increment operator ");

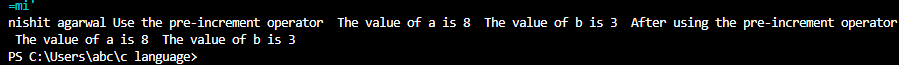
printf (" The value of a is %d ", x);

printf (" The value of b is %d ", y);

return 0;

}

OUTPUT



**Program-20**

**Wap to perform the post increment and post decrement operator on two integers and print both original value and updated value**

**INPUT**

#include <stdio.h>

#include <conio.h>

int main ()

{

int a = 7;

int b = 2;

int x,y;

x = a++;

y= b++;

printf("nishit agarwal");

printf (" Use the pre-increment operator " );

printf (" The value of a is %d ", a);

printf (" The value of b is %d ", b);

printf (" After using the pre-increment operator ");

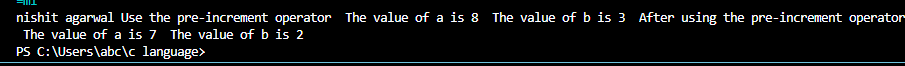
printf (" The value of a is %d ", x);

printf (" The value of b is %d ", y);

return 0;

}

OUTPUT



**Program-21**

**Wap for an integer number and to check whether it is divisible by 9 or 7 using OR logical operator**

**INPUT**

#include<stdio.h>

int main()

{

    int num;

    printf("nishit agarwal");

    printf("enter any number:");

    scanf("%d", &num);

    if((num%9==0)&&(num%7==0))

    {

        printf(" number is divisible by 9 and 7");

    }

    else

    {

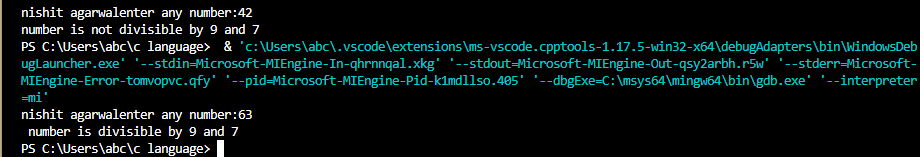
        printf("number is not divisible by 9 and 7");

    }

    return 0;

}

OUTPUT



**Program-22**

**Wap to identify gender in single character and print full gender (Ex: if input is ‘M’ or ‘m’- it should print “Male”)**

**INPUT**

#include<stdio.h>

int main()

{

    char gender;

    printf("nishit agarwal");

    printf("enter a single character");

    scanf("%c", &gender);

    switch (gender)

    {

        case 'M':

        case 'm':

        printf("male\n");

        break;

         case 'F':

        case 'f':

        printf("female\n");

        break;

         case 'O':

        case 'o':

        printf("other\n");

        break;

        default:

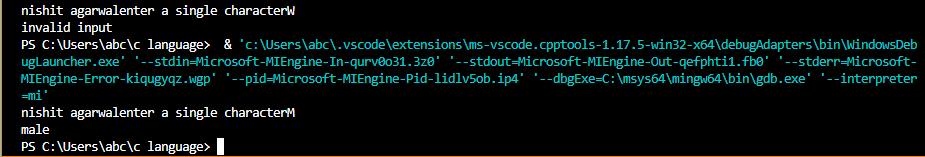
        printf("invalid input\n");

        return 0;

    }

}

OUTPUT



Programme - 23

Write a C program to print all natural numbers in reverse (from n to 1).

INPUT

#include<stdio.h>

int main()

{int n;

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer (n): ");

    scanf("%d", &n);

    if (n <= 0)

    {

        printf("Please enter a positive integer.\n");

    } else

    {

        printf("Natural numbers from %d to 1 in reverse order:\n", n);

        while (n >= 1)

        {

            printf("%d\n", n);

            n--;

        }

    }return 0;

}

OUTPUT

NISHIT AGARWALEnter a positive integer (n): 10

Natural numbers from 10 to 1 in reverse order:

10

9

8

7

6

5

4

3

2

1

PS C:\Users\abc\c language>

**Programme-24**

**Write a C program to print all alphabets from a to z**

**INPUT**

#include <stdio.h>

int main()

{

    char ch;

    printf("NISHIT AGARWAL");

    printf("Alphabets from a to z:\n");

    for (ch = 'a'; ch <= 'z'; ch++)

    {

        printf("%c ", ch);

    }

    printf("\n");

    return 0;

}

**OUTPUT**

**NISHIT AGARWALAlphabets from a to z:**

**a b c d e f g h i j k l m n o p q r s t u v w x y z**

**PS C:\Users\abc\c language>**

**Programme-25**

**Write a C Write a C program to print all natural numbers from 1 to n.**

**INPUT**

#include <stdio.h>

int main()

{

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer (n): ");

    scanf("%d", &n);

    if (n <= 0)

    {

        printf("Please enter a positive integer.\n");

    } else

    {

        printf("Natural numbers from 1 to %d:\n", n);

        for (int i = 1; i <= n; i++)

        {

            printf("%d\n", i);

        }

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a positive integer (n): 4**

**Natural numbers from 1 to 4:**

**1**

**2**

**3**

**4**

**PS C:\Users\abc\c language>**

**Programme -26**

program to print all even numbers between 1 to 100.

**INPUT**

#include <stdio.h>

int main()

{

    printf("NISHIT AGARWAL");

    printf("Even numbers between 1 and 100:\n");

    for (int i = 2; i <= 100; i += 2)

    {

        printf("%d\n", i);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEven numbers between 1 and 100:**

**2**

**4**

**6**

**8**

**10**

**12**

**14**

**16**

**18**

**20**

**22**

**24**

**26**

**28**

**30**

**32**

**34**

**36**

**38**

**40**

**42**

**44**

**46**

**48**

**50**

**52**

**54**

**56**

**58**

**60**

**62**

**64**

**66**

**68**

**70**

**72**

**74**

**76**

**78**

**80**

**82**

**84**

**86**

**88**

**90**

**92**

**94**

**96**

**98**

**100**

**PS C:\Users\abc\c language>**

**Programme -27**

**Write a C program to print all odd number between 1 to 100**

**INPUT**

#include<stdio.h>

int main()

{

    printf("NISHIT AGARWAL");

    printf("Odd numbers between 1 and 100:\n");

    for (int i = 1; i <= 100; i += 2)

    {

        printf("%d\n", i);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALOdd numbers between 1 and 100:**

**1**

**3**

**5**

**7**

**9**

**11**

**13**

**15**

**17**

**19**

**21**

**23**

**25**

**27**

**29**

**31**

**33**

**35**

**37**

**39**

**41**

**43**

**45**

**47**

**49**

**51**

**53**

**55**

**57**

**59**

**61**

**63**

**65**

**67**

**69**

**71**

**73**

**75**

**77**

**79**

**81**

**83**

**85**

**87**

**89**

**91**

**93**

**95**

**97**

**99**

**PS C:\Users\abc\c language>**

**Programme-28**

Write a C program to find sum of all natural numbers between 1 to n.

**Input**

#include <stdio.h>

int main()

{

    int n, sum = 0;

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer (n): ");

    scanf("%d", &n);

    if (n <= 0)

    {

        printf("Please enter a positive integer.\n");

    } else

    {

        for (int i = 1; i <= n; i++)

        {

            sum += i;

        }

        printf("The sum of natural numbers from 1 to %d is: %d\n", n, sum);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a positive integer (n): 567**

**The sum of natural numbers from 1 to 567 is: 161028**

**PS C:\Users\abc\c language>**

**Programme-29**

Write a C program to find sum of all even numbers between 1 to n.

Input

#include <stdio.h>

int main() {

    int n, sum = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer (n): ");

    scanf("%d", &n);

    // Check if the input is a positive integer

    if (n <= 0) {

        printf("Please enter a positive integer.\n");

    } else {

        // Calculate the sum of even numbers from 2 to n

        for (int i = 2; i <= n; i += 2) {

            sum += i;

        }

        printf("The sum of even numbers from 2 to %d is: %d\n", n, sum);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a positive integer (n): 5**

**The sum of even numbers from 2 to 5 is: 6**

**PS C:\Users\abc\c language>**

Programme-30

Write a C program to find sum of all odd numbers between 1 to n.

Input

#include <stdio.h>

int main() {

    int n, sum = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer (n): ");

    scanf("%d", &n);

    // Check if the input is a positive integer

    if (n <= 0) {

        printf("Please enter a positive integer.\n");

    } else {

        // Calculate the sum of odd numbers from 1 to n

        for (int i = 1; i <= n; i += 2) {

            sum += i;

        }

        printf("The sum of odd numbers from 1 to %d is: %d\n", n, sum);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a positive integer (n): 10**

**The sum of odd numbers from 1 to 10 is: 25**

**PS C:\Users\abc\c language>**

Programme-31

Write a C program to print multiplication table of any number

INPUT

#include <stdio.h>

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number to print its multiplication table: ");

    scanf("%d", &number);

    printf("Multiplication table for %d:\n", number);

    for (int i = 1; i <= 10; i++) {

        printf("%d x %d = %d\n", number, i, number \* i);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a number to print its multiplication table: 3**

**Multiplication table for 3:**

**3 x 1 = 3**

**3 x 2 = 6**

**3 x 3 = 9**

**3 x 4 = 12**

**3 x 5 = 15**

**3 x 6 = 18**

**3 x 7 = 21**

**3 x 8 = 24**

**3 x 9 = 27**

**3 x 10 = 30**

**PS C:\Users\abc\c language>**

Programme-32

Write a C program to count number of digits in a number

**INPUT**

#include <stdio.h>

int main() {

    long long number;

    int count = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%lld", &number);

    // Handle negative numbers

    if (number < 0) {

        number = -number;

    }

    // Count the number of digits

    if (number == 0) {

        count = 1; // If the number is 0, it has one digit

    } else {

        while (number > 0) {

            count++;

            number /= 10;

        }

    }

    printf("Number of digits: %d\n", count);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a number: 3**

**Number of digits: 1**

**PS C:\Users\abc\c language>**

Programme-33

Write a C program to find first and last digit of a number.

INPUT

#include <stdio.h>

int main() {

    int number, firstDigit, lastDigit;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    // Store the last digit

    lastDigit = number % 10;

    // Find the first digit

    firstDigit = number;

    while (firstDigit >= 10) {

        firstDigit /= 10;

    }

    printf("First digit: %d\n", firstDigit);

    printf("Last digit: %d\n", lastDigit);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number: 45

First digit: 4

Last digit: 5

PS C:\Users\abc\c language>

Programme-34

Write a C program to find sum of first and last digit of a number

INPUT

#include <stdio.h>

int main() {

    int number, firstDigit, lastDigit, sum;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    // Store the last digit

    lastDigit = number % 10;

    // Find the first digit

    firstDigit = number;

    while (firstDigit >= 10) {

        firstDigit /= 10;

    }

    // Calculate the sum

    sum = firstDigit + lastDigit;

    printf("First digit: %d\n", firstDigit);

    printf("Last digit: %d\n", lastDigit);

    printf("Sum of the first and last digits: %d\n", sum);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number: 45

First digit: 4

Last digit: 5

Sum of the first and last digits: 9

PS C:\Users\abc\c language>

Programme-35

Write a C program to swap first and last digits of a number.

INPUT

#include <stdio.h>

int main() {

    int number, originalNumber, firstDigit, lastDigit, swappedNumber = 0;

    int divisor = 1;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    originalNumber = number;

    // Find the last digit

    lastDigit = number % 10;

    // Find the number of digits

    while (number >= 10) {

        number /= 10;

        divisor \*= 10;

    }

    // Find the first digit

    firstDigit = originalNumber / divisor;

    // Swap the first and last digits

    swappedNumber = lastDigit \* divisor + originalNumber % divisor;

    swappedNumber = swappedNumber - lastDigit + firstDigit;

    printf("Original number: %d\n", originalNumber);

    printf("Number after swapping first and last digits: %d\n", swappedNumber);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number: 456

Original number: 456

Number after swapping first and last digits: 654

PS C:\Users\abc\c language>

Programme-36

Write a C program to calculate sum of digits of a number

INPUT

#include <stdio.h>

int main() {

    int number, sum = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    // Make sure the number is positive

    if (number < 0) {

        number = -number;

    }

    // Calculate the sum of digits

    while (number > 0) {

        sum += number % 10;

        number /= 10;

    }

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

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number: 567

Sum of digits: 18

PS C:\Users\abc\c language>

Programme-37

Write a C program to calculate product of digits of a number

INPUT

#include <stdio.h>

int main() {

    int number, product = 1;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    // Make sure the number is positive

    if (number < 0) {

        number = -number;

    }

    // Calculate the product of digits

    while (number > 0) {

        product \*= number % 10;

        number /= 10;

    }

    printf("Product of digits: %d\n", product);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number: 56

Product of digits: 30

PS C:\Users\abc\c language>

Programme-38

Write a C program to enter a number and print its reverse

INPUT

#include <stdio.h>

int main() {

    int number, reversedNumber = 0, remainder;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    // Reverse the number

    while (number != 0) {

        remainder = number % 10;

        reversedNumber = reversedNumber \* 10 + remainder;

        number /= 10;

    }

    printf("Reversed number: %d\n", reversedNumber);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number: 345

Reversed number: 543

PS C:\Users\abc\c language>

Programme-39

Write a C program to check whether a number is palindrome or not

**INPUT**

#include<stdio.h>

int main()

{

    int n, result=0, q, rem;

    printf("NISHIT AGARWAL");

    printf("please enter the number");

    scanf("%d", &n);

    q = n;

    while(q!=0)

    {

        rem = q%10;

        result = result\*10 + rem;

        q = q/10;

    }

    if(result == n)

    printf("its a palindrome");

    else

    printf("No! its not a palindromre");

    return 0;

}

**OUTPUT**

**NISHIT AGARWALplease enter the number12321**

**its a palindrome**

**PS C:\Users\abc\c language>**

Programme-40

Write a C program to find frequency of each digit in a given integer

INPUT

#include <stdio.h>

int main() {

    int number;

    int digit, count;

    int frequency[10] = {0};

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter an integer: ");

    scanf("%d", &number);

    // Make sure the number is positive

    if (number < 0) {

        number = -number;

    }

    // Calculate the frequency of each digit

    while (number > 0) {

        digit = number % 10;

        frequency[digit]++;

        number /= 10;

    }

    // Print the frequency of each digit

    printf("Frequency of each digit:\n");

    for (digit = 0; digit < 10; digit++) {

        if (frequency[digit] > 0) {

            printf("Digit %d occurs %d times\n", digit, frequency[digit]);

        }

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter an integer: 45

Frequency of each digit:

Digit 4 occurs 1 times

Digit 5 occurs 1 times

PS C:\Users\abc\c language>

PROGRAMME-41

Write a C program to enter a number and print it in words

INPUT

#include <stdio.h>

void printNumberInWords(int num) {

    // Arrays to store the words for numbers 1 to 19

    char \*units[] = {"", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};

    // Arrays to store the words for multiples of 10

    char \*tens[] = {"", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};

    if (num < 1 || num > 99) {

        printf("Number out of range (1-99).\n");

    } else if (num < 20) {

        printf("%s\n", units[num]);

    } else {

        printf("%s %s\n", tens[num / 10], units[num % 10]);

    }

}

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number (1-99): ");

    scanf("%d", &number);

    // Convert and print the number in words

    printNumberInWords(number);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number (1-99): 78

Seventy Eight

PS C:\Users\abc\c language>

Programme-42

Write a C program to print all ASCII character with their values

INPUT

#include <stdio.h>

int main()

{

    printf("NISHIT AGARWAL");

    printf("ASCII Characters and Their Values:\n");

    for (int i = 0; i <= 127; i++) {

        printf("Character: %c\tASCII Value: %d\n", i, i);

    }

    return 0;

}

OUTPUT

NISHIT AGARWALASCII Characters and Their Values:

Character: ASCII Value: 0

Character: ASCII Value: 1

Character: ASCII Value: 2

Character: ASCII Value: 3

Character: ASCII Value: 4

Character: ASCII Value: 5

Character: ASCII Value: 6

Character: ASCII Value: 7

Character: ASCII Value: 8

Character: ASCII Value: 9

Character:

ASCII Value: 10

Character:

ASCII Value: 11

Character:

ASCII Value: 12

CharacteASCII Value: 13

Character: ASCII Value: 14

Character: ASCII Value: 15

Character: ASCII Value: 16

Character: ASCII Value: 17

Character: ASCII Value: 18

Character: ASCII Value: 19

Character: ASCII Value: 20

Character: ASCII Value: 21

Character: ASCII Value: 22

Character: ASCII Value: 23

Character: ASCII Value: 24

Character: ASCII Value: 25

Character: ASCII Value: 26

Character: SCII Value: 27

Character: ASCII Value: 28

Character: ASCII Value: 29

Character: ASCII Value: 30

Character: ASCII Value: 31

Character: ASCII Value: 32

Character: ! ASCII Value: 33

Character: " ASCII Value: 34

Character: # ASCII Value: 35

Character: $ ASCII Value: 36

Character: % ASCII Value: 37

Character: & ASCII Value: 38

Character: ' ASCII Value: 39

Character: ( ASCII Value: 40

Character: ) ASCII Value: 41

Character: \* ASCII Value: 42

Character: + ASCII Value: 43

Character: , ASCII Value: 44

Character: - ASCII Value: 45

Character: . ASCII Value: 46

Character: / ASCII Value: 47

Character: 0 ASCII Value: 48

Character: 1 ASCII Value: 49

Character: 2 ASCII Value: 50

Character: 3 ASCII Value: 51

Character: 4 ASCII Value: 52

Character: 5 ASCII Value: 53

Character: 6 ASCII Value: 54

Character: 7 ASCII Value: 55

Character: 8 ASCII Value: 56

Character: 9 ASCII Value: 57

Character: : ASCII Value: 58

Character: ; ASCII Value: 59

Character: < ASCII Value: 60

Character: = ASCII Value: 61

Character: > ASCII Value: 62

Character: ? ASCII Value: 63

Character: @ ASCII Value: 64

Character: A ASCII Value: 65

Character: B ASCII Value: 66

Character: C ASCII Value: 67

Character: D ASCII Value: 68

Character: E ASCII Value: 69

Character: F ASCII Value: 70

Character: G ASCII Value: 71

Character: H ASCII Value: 72

Character: I ASCII Value: 73

Character: J ASCII Value: 74

Character: K ASCII Value: 75

Character: L ASCII Value: 76

Character: M ASCII Value: 77

Character: N ASCII Value: 78

Character: O ASCII Value: 79

Character: P ASCII Value: 80

Character: Q ASCII Value: 81

Character: R ASCII Value: 82

Character: S ASCII Value: 83

Character: T ASCII Value: 84

Character: U ASCII Value: 85

Character: V ASCII Value: 86

Character: W ASCII Value: 87

Character: X ASCII Value: 88

Character: Y ASCII Value: 89

Character: Z ASCII Value: 90

Character: [ ASCII Value: 91

Character: \ ASCII Value: 92

Character: ] ASCII Value: 93

Character: ^ ASCII Value: 94

Character: \_ ASCII Value: 95

Character: ` ASCII Value: 96

Character: a ASCII Value: 97

Character: b ASCII Value: 98

Character: c ASCII Value: 99

Character: d ASCII Value: 100

Character: e ASCII Value: 101

Character: f ASCII Value: 102

Character: g ASCII Value: 103

Character: h ASCII Value: 104

Character: i ASCII Value: 105

Character: j ASCII Value: 106

Character: k ASCII Value: 107

Character: l ASCII Value: 108

Character: m ASCII Value: 109

Character: n ASCII Value: 110

Character: o ASCII Value: 111

Character: p ASCII Value: 112

Character: q ASCII Value: 113

Character: r ASCII Value: 114

Character: s ASCII Value: 115

Character: t ASCII Value: 116

Character: u ASCII Value: 117

Character: v ASCII Value: 118

Character: w ASCII Value: 119

Character: x ASCII Value: 120

Character: y ASCII Value: 121

Character: z ASCII Value: 122

Character: { ASCII Value: 123

Character: | ASCII Value: 124

Character: } ASCII Value: 125

Character: ~ ASCII Value: 126

Character: ASCII Value: 127

PS C:\Users\abc\c language>

Programme-43

Write a C program to find power of a number using for loop

INPUT

#include <stdio.h>

int main() {

    double base, exponent, result = 1.0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the base: ");

    scanf("%lf", &base);

    printf("Enter the exponent: ");

    scanf("%lf", &exponent);

    // Calculate the power using a for loop

    for (int i = 1; i <= exponent; i++) {

        result \*= base;

    }

    printf("%.2lf^%.2lf = %.2lf\n", base, exponent, result);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter the base: 45

Enter the exponent: 3

45.00^3.00 = 91125.00

PS C:\Users\abc\c language>

Programme-44

Write a C program to find all factors of a number

INPUT

#include <stdio.h>

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

printf("Factors of %d are:\n", number);

for (int i = 1; i <= number; i++) {

        if (number % i == 0) {

            printf("%d\n", i);

        }

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number: 56

Factors of 56 are:

1

2

4

7

8

14

28

56

PS C:\Users\abc\c language>

Programme-45

Write a C program to calculate factorial of a number

INPUT

#include <stdio.h>

int main() {

    int number;

    long long factorial = 1;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a non-negative integer: ");

    scanf("%d", &number);

    if (number < 0) {

        printf("Factorial is not defined for negative numbers.\n");

    } else {

        for (int i = 1; i <= number; i++) {

            factorial \*= i;

        }

        printf("The factorial of %d is %lld\n", number, factorial);

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a non-negative integer: 56

The factorial of 56 is 6908521828386340864

PS C:\Users\abc\c language>

Programme-46

Write a C program to find HCF (GCD) of two numbers

INPUT

#include <stdio.h>

// Function to calculate the GCD using the Euclidean algorithm

int findGCD(int a, int b) {

    if (b == 0) {

        return a;

    } else {

        return findGCD(b, a % b);

    }

}

int main() {

    int num1, num2;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter two numbers: ");

    scanf("%d %d", &num1, &num2);

    if (num1 < 0 || num2 < 0) {

        printf("Please enter non-negative numbers.\n");

    } else {

        // Find and print the GCD

        int gcd = findGCD(num1, num2);

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

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter two numbers: 34

35

The GCD of 34 and 35 is 1

PS C:\Users\abc\c language>

Programme-47

Write a C program to find LCM of two numbers

INPUT

#include <stdio.h>

// Function to calculate the GCD (Greatest Common Divisor) using the Euclidean algorithm

int findGCD(int a, int b) {

    if (b == 0) {

        return a;

    } else {

        return findGCD(b, a % b);

    }

}

// Function to calculate the LCM (Least Common Multiple)

int findLCM(int a, int b) {

    int gcd = findGCD(a, b);

    int lcm = (a \* b) / gcd;

    return lcm;

}

int main() {

    int num1, num2;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter two numbers: ");

    scanf("%d %d", &num1, &num2);

    if (num1 < 0 || num2 < 0) {

        printf("Please enter non-negative numbers.\n");

    } else {

        // Find and print the LCM

        int lcm = findLCM(num1, num2);

        printf("The LCM of %d and %d is %d\n", num1, num2, lcm);

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter two numbers: 12

14

The LCM of 12 and 14 is 84

PS C:\Users\abc\c language>

Programme-48

Write a C program to check whether a number is Prime number or not

INPUT

#include<stdio.h>

int main()

{

    int n, i, c = 0;

    printf("NISHIT AGARWAL");

    printf("enter any number");

    scanf("%d", &n);

    for(i=1; i <= n; i++)

    {

        if (n%i == 0)

        {

            c++;

        }

    }

    if (c == 2)

    {

        printf("%d is a prime number",n);

    }

    else{

        printf("%d is not a prime number",n);

    }

    return 0;

}

OUTPUT

NISHIT AGARWALenter any number12

12 is not a prime number

PS C:\Users\abc\c language>

Programme-49

Write a C program to print all Prime numbers between 1 to n

INPUT

#include <stdio.h>

#include <stdbool.h>

bool isPrime(int num) {

    if (num <= 1) {

        return false;

    }

    for (int i = 2; i \* i <= num; ++i) {

        if (num % i == 0) {

            return false;

        }

    }

    return true;

}

int main() {

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &n);

    printf("Prime numbers between 1 and %d are: ", n);

    for (int i = 2; i <= n; ++i) {

        if (isPrime(i)) {

            printf("%d ", i);

        }

    }

    printf("\n");

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number: 12

Prime numbers between 1 and 12 are: 2 3 5 7 11

PS C:\Users\abc\c language>

Programme-50

Write a C program to find sum of all prime numbers between 1 to n

INPUT

#include <stdio.h>

#include <stdbool.h>

bool isPrime(int num) {

    if (num <= 1) {

        return false;

    }

    for (int i = 2; i \* i <= num; i++) {

        if (num % i == 0) {

            return false;

        }

    }

    return true;

}

int main() {

    int n, sum = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer (n): ");

    scanf("%d", &n);

    if (n <= 1) {

        printf("There are no prime numbers in the specified range.\n");

    } else {

        printf("Prime numbers between 1 and %d are:\n", n);

        for (int i = 2; i <= n; i++) {

            if (isPrime(i)) {

                printf("%d\n", i);

                sum += i;

            }

        }

        printf("Sum of prime numbers in the specified range: %d\n", sum);

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a positive integer (n): 12

Prime numbers between 1 and 12 are:

2

3

5

7

11

Sum of prime numbers in the specified range: 28

PS C:\Users\abc\c language>

Programme-51

Write a C program to find all prime factors of a number

INPUT

#include <stdio.h>

// Function to check if a number is prime

int isPrime(int num) {

    if (num <= 1) {

        return 0;  // Not prime

    }

    for (int i = 2; i \* i <= num; i++) {

        if (num % i == 0) {

            return 0;  // Not prime

        }

    }

    return 1;  // Prime

}

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    if (number <= 1) {

        printf("Prime factors are not defined for numbers less than 2.\n");

    } else {

        printf("Prime factors of %d are:\n", number);

        // Find and print the prime factors

        for (int i = 2; i <= number; i++) {

            if (number % i == 0 && isPrime(i)) {

                while (number % i == 0) {

                    printf("%d ", i);

                    number /= i;

                }

            }

        }

        printf("\n");

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a positive integer: 12

Prime factors of 12 are:

2 2 3

PS C:\Users\abc\c language>

Programme-52

Write a C program to check whether a number is Armstrong number or not

INPUT

#include <stdio.h>

#include <math.h>

int isArmstrong(int num) {

    int originalNum, remainder, n = 0, result = 0;

    originalNum = num;

    // Count the number of digits

    while (originalNum != 0) {

        originalNum /= 10;

        n++;

    }

    originalNum = num;

    // Calculate the sum of digits each raised to the power of n

    while (originalNum != 0) {

        remainder = originalNum % 10;

        result += pow(remainder, n);

        originalNum /= 10;

    }

    return (result == num);

}

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    if (isArmstrong(number)) {

        printf("%d is an Armstrong number.\n", number);

    } else {

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

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a number: 12

12 is not an Armstrong number.

PS C:\Users\abc\c language>

Programme-53

Write a C program to print all Armstrong numbers between 1 to n

INPUT

#include <stdio.h>

#include <math.h>

int isArmstrong(int num) {

    int originalNum, remainder, n = 0, result = 0;

    originalNum = num;

    // Count the number of digits

    while (originalNum != 0) {

        originalNum /= 10;

        n++;

    }

    originalNum = num;

    // Calculate the sum of digits each raised to the power of n

    while (originalNum != 0) {

        remainder = originalNum % 10;

        result += pow(remainder, n);

        originalNum /= 10;

    }

    return (result == num);

}

int main() {

    int n;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer (n): ");

    scanf("%d", &n);

    if (n < 1) {

        printf("Armstrong numbers are not defined for numbers less than 1.\n");

    } else {

        printf("Armstrong numbers between 1 and %d are:\n", n);

        for (int i = 1; i <= n; i++) {

            if (isArmstrong(i)) {

                printf("%d\n", i);

            }

        }

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a positive integer (n): 12

Armstrong numbers between 1 and 12 are:

1

2

3

4

5

6

7

8

9

PS C:\Users\abc\c language>

PROGRAMME-54

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

INPUT

#include <stdio.h>

int isPerfect(int num) {

    int sum = 0;

    // Find the divisors and calculate the sum of proper divisors

    for (int i = 1; i < num; i++) {

        if (num % i == 0) {

            sum += i;

        }

    }

    return (sum == num);

}

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    if (number <= 0) {

        printf("Perfect numbers are defined for positive integers only.\n");

    } else if (isPerfect(number)) {

        printf("%d is a perfect number.\n", number);

    } else {

        printf("%d is not a perfect number.\n", number);

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a positive integer: 12

12 is not a perfect number.

PS C:\Users\abc\c language>

Programme-55

Write a C program to print all Perfect numbers between 1 to n

INPUT

#include <stdio.h>

int isPerfect(int num) {

    int sum = 0;

    // Find the divisors and calculate the sum of proper divisors

    for (int i = 1; i < num; i++) {

        if (num % i == 0) {

            sum += i;

        }

    }

    return (sum == num);

}

int main() {

    int n;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer (n): ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Perfect numbers are defined for positive integers only.\n");

    } else {

        printf("Perfect numbers between 1 and %d are:\n", n);

        for (int i = 1; i <= n; i++) {

            if (isPerfect(i)) {

                printf("%d\n", i);

            }

        }

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a positive integer (n): 12

Perfect numbers between 1 and 12 are:

6

PS C:\Users\abc\c language>

Programme-56

Write a C program to check whether a number is Strong number or not

INPUT

#include <stdio.h>

int factorial(int num) {

    if (num == 0) {

        return 1;

    } else {

        int fact = 1;

        for (int i = 1; i <= num; i++) {

            fact \*= i;

        }

        return fact;

    }

}

int isStrong(int num) {

    int originalNum, remainder, sum = 0;

    originalNum = num;

    // Calculate the sum of the factorial of digits

    while (originalNum != 0) {

        remainder = originalNum % 10;

        sum += factorial(remainder);

        originalNum /= 10;

    }

    return (sum == num);

}

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer: ");

    scanf("%d", &number);

    if (number <= 0) {

        printf("Strong numbers are defined for positive integers only.\n");

    } else if (isStrong(number)) {

        printf("%d is a strong number.\n", number);

    } else {

        printf("%d is not a strong number.\n", number);

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a positive integer: 12

12 is not a strong number.

PS C:\Users\abc\c language>

Programme-57

Write a C program to print all Strong numbers between 1 to n

INPUT

#include <stdio.h>

int factorial(int num)

{

    if (num == 0) {

        return 1;

    } else {

        int fact = 1;

        for (int i = 1; i <= num; i++) {

            fact \*= i;

        }

        return fact;

    }

}

int isStrong(int num) {

    int originalNum, remainder, sum = 0;

    originalNum = num;

    // Calculate the sum of the factorial of digits

    while (originalNum != 0) {

        remainder = originalNum % 10;

        sum += factorial(remainder);

        originalNum /= 10;

    }

    return (sum == num);

}

int main() {

    int n;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a positive integer (n): ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Strong numbers are defined for positive integers only.\n");

    } else {

        printf("Strong numbers between 1 and %d are:\n", n);

        for (int i = 1; i <= n; i++) {

            if (isStrong(i)) {

                printf("%d\n", i);

            }

        }

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a positive integer (n): 12

Strong numbers between 1 and 12 are:

1

2

PS C:\Users\abc\c language>

Programme-58

Write a C program to print Fibonacci series up to n terms

INPUT

#include <stdio.h>

int main() {

    int n, first = 0, second = 1, next;

    // Ask the user for input

    printf("NISHIT AGARWAL");

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

    scanf("%d", &n);

    printf("Fibonacci Series up to %d terms:\n", n);

    for (int i = 0; i < n; i++) {

        if (i <= 1) {

            next = i;

        } else {

            next = first + second;

            first = second;

            second = next;

        }

        printf("%d ", next);

    }

    printf("\n");

    return 0;

}

OUTPUT

NISHIT AGARWALEnter the number of terms: 12

Fibonacci Series up to 12 terms:

0 1 1 2 3 5 8 13 21 34 55 89

PS C:\Users\abc\c language>

Programme-59

Write a C program to find one's complement of a binary number

INPUT

#include <stdio.h>

int main() {

    char binary[32];

    char onesComplement[32];

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a binary number: ");

    scanf("%s", binary);

    // Calculate the one's complement

    int i = 0;

    while (binary[i] != '\0') {

        if (binary[i] == '0') {

            onesComplement[i] = '1';

        } else if (binary[i] == '1') {

            onesComplement[i] = '0';

        } else {

            printf("Invalid binary number.\n");

            return 1;

        }

        i++;

    }

    onesComplement[i] = '\0';

    // Print the one's complement

    printf("One's complement: %s\n", onesComplement);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a binary number: 12

Invalid binary number.

PS C:\Users\abc\c language>

Programme-60

Write a C program to find two's complement of a binary number

INPUT

#include <stdio.h>

void reverse(char binary[]) {

    int start = 0;

    int end = strlen(binary) - 1;

    while (start < end) {

        char temp = binary[start];

        binary[start] = binary[end];

        binary[end] = temp;

        start++;

        end--;

    }

}

void onesComplement(char binary[]) {

    for (int i = 0; binary[i] != '\0'; i++) {

        if (binary[i] == '0') {

            binary[i] = '1';

        } else if (binary[i] == '1') {

            binary[i] = '0';

        }

    }

}

int main() {

    char binary[32];

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a binary number: ");

    scanf("%s", binary);

    // Reverse the binary number

    reverse(binary);

    // Calculate the one's complement

    onesComplement(binary);

    // Add 1 to the LSB to find the two's complement

    int carry = 1;

    for (int i = 0; binary[i] != '\0'; i++) {

        if (binary[i] == '0' && carry == 1) {

            binary[i] = '1';

            carry = 0;

        } else if (binary[i] == '1' && carry == 1) {

            binary[i] = '0';

        }

    }

    // Reverse the result to obtain two's complement

    reverse(binary);

    // Print the two's complement

    printf("Two's complement: %s\n", binary);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a binary number: 1100

Two's complement: 0100

PS C:\Users\abc\c language>

Programme-61

Write a C program to convert Binary to Octal number system

INPUT

#include <stdio.h>

#include <math.h>

// Function to convert a binary digit to an octal digit

int binaryToOctalDigit(char binaryDigit) {

    int decimalDigit = binaryDigit - '0'; // Convert ASCII '0' or '1' to integer 0 or 1

    return decimalDigit;

}

int main() {

    char binary[32];

    int octal[32];

    int length, octalDigit, i, j;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a binary number: ");

    scanf("%s", binary);

    // Calculate the length of the binary number

    length = strlen(binary);

    // Make sure the binary length is a multiple of 3 by adding leading zeros if needed

    int remainder = length % 3;

    if (remainder > 0) {

        for (i = 0; i < 3 - remainder; i++) {

            binary[length + i] = '0';

        }

        length += 3 - remainder;

    }

    // Initialize the octal array to zero

    for (i = 0; i < length / 3; i++) {

        octal[i] = 0;

    }

    // Convert binary to octal

    for (i = 0, j = length / 3 - 1; i < length; i += 3, j--) {

        octalDigit = binaryToOctalDigit(binary[i]) \* 4 +

                     binaryToOctalDigit(binary[i + 1]) \* 2 +

                     binaryToOctalDigit(binary[i + 2]);

        octal[j] = octalDigit;

    }

    // Print the octal number

    printf("Octal equivalent: ");

    for (i = 0; i < length / 3; i++) {

        printf("%d", octal[i]);

    }

    printf("\n");

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a binary number: 1100

Octal equivalent: 06

PS C:\Users\abc\c language>

Programme-62

Write a C program to convert Binary to Decimal number system

INPUT

#include <stdio.h>

#include <string.h>

int main() {

    char binary[32];

    int decimal = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a binary number: ");

    scanf("%s", binary);

    int length = strlen(binary);

    int base = 1; // 2^0

    // Convert binary to decimal

    for (int i = length - 1; i >= 0; i--) {

        if (binary[i] == '1') {

            decimal += base;

        }

        base \*= 2; // Multiply by 2 for the next position

    }

    // Print the decimal number

    printf("Decimal equivalent: %d\n", decimal);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a binary number: 1100

Decimal equivalent: 12

PS C:\Users\abc\c language>

Programme-63

Write a C program to convert Binary to Hexadecimal number system.

INPUT

#include <stdio.h>

#include <string.h>

// Function to convert a 4-bit binary number to a hexadecimal digit

char binaryToHexDigit(char binaryDigits[4]) {

    int decimal = 0;

    for (int i = 0; i < 4; i++) {

        decimal = (decimal << 1) | (binaryDigits[i] - '0');

    }

    if (decimal >= 0 && decimal <= 9) {

        return (char)(decimal + '0');

    } else {

        return (char)(decimal - 10 + 'A');

    }

}

int main() {

    char binary[32];

    char hex[32];

    int length, hexDigitCount = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a binary number: ");

    scanf("%s", binary);

    // Calculate the length of the binary number

    length = strlen(binary);

    // Make sure the binary length is a multiple of 4 by adding leading zeros if needed

    int remainder = length % 4;

    if (remainder > 0) {

        for (int i = 0; i < 4 - remainder; i++) {

            binary[length + i] = '0';

        }

        length += 4 - remainder;

    }

    // Convert binary to hexadecimal

    for (int i = 0; i < length; i += 4) {

        char binaryDigits[4];

        for (int j = 0; j < 4; j++) {

            binaryDigits[j] = binary[i + j];

        }

        hex[hexDigitCount] = binaryToHexDigit(binaryDigits);

        hexDigitCount++;

    }

    hex[hexDigitCount] = '\0';

    // Print the hexadecimal number

    printf("Hexadecimal equivalent: %s\n", hex);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a binary number: 1100

Hexadecimal equivalent: C

PS C:\Users\abc\c language>

Programme-64

Write a C program to convert Octal to Binary number system

INPUT

#include <stdio.h>

#include <string.h>

// Function to convert an octal digit to its binary representation

void octalToBinaryDigit(char octalDigit, char binaryDigits[4]) {

    int decimal = octalDigit - '0';

    int index = 0;

    // Convert octal to binary

    while (decimal > 0) {

        binaryDigits[index] = (decimal % 2) + '0';

        decimal /= 2;

        index++;

    }

    // Pad with leading zeros to make it 3 bits

    while (index < 3) {

        binaryDigits[index] = '0';

        index++;

    }

    // Null-terminate the string

    binaryDigits[3] = '\0';

    // Reverse the binaryDigits array to get the correct order

    char temp;

    for (int i = 0; i < index / 2; i++) {

        temp = binaryDigits[i];

        binaryDigits[i] = binaryDigits[index - i - 1];

        binaryDigits[index - i - 1] = temp;

    }

}

int main() {

    char octal[32];

    char binary[96];  // Maximum 32 octal digits can be converted to 96 binary digits

    int length, binaryDigitCount = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter an octal number: ");

    scanf("%s", octal);

    // Calculate the length of the octal number

    length = strlen(octal);

    // Convert octal to binary

    for (int i = 0; i < length; i++) {

        char binaryDigits[4];

        octalToBinaryDigit(octal[i], binaryDigits);

        strcat(binary, binaryDigits);

        binaryDigitCount += 3;

    }

    // Remove leading zeros if any

    int leadingZeroCount = 0;

    while (binary[leadingZeroCount] == '0') {

        leadingZeroCount++;

    }

    // Print the binary number

    if (binary[leadingZeroCount] == '\0') {

        printf("Binary equivalent: 0\n");

    } else {

        printf("Binary equivalent: %s\n", binary + leadingZeroCount);

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter an octal number: 200

Binary equivalent: 010000000

PS C:\Users\abc\c language>

Programme-65

Write a C program to convert Octal to Decimal number system

INPUT

#include <stdio.h>

#include <string.h>

int main() {

    char octal[32];

    int decimal = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter an octal number: ");

    scanf("%s", octal);

    int length = strlen(octal);

    int base = 1; // 8^0

    // Convert octal to decimal

    for (int i = length - 1; i >= 0; i--) {

        int octalDigit = octal[i] - '0';

        decimal += octalDigit \* base;

        base \*= 8; // Multiply by 8 for the next position

    }

    // Print the decimal number

    printf("Decimal equivalent: %d\n", decimal);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter an octal number: 200

Decimal equivalent: 128

PS C:\Users\abc\c language>

Programme-66

Write a C program to convert Octal to Hexadecimal number system

INPUT

#include <stdio.h>

#include <string.h>

// Function to convert an octal digit to its binary representation

void octalToBinaryDigit(char octalDigit, char binaryDigits[4]) {

    int decimal = octalDigit - '0';

    int index = 0;

    // Convert octal to binary

    while (decimal > 0) {

        binaryDigits[index] = (decimal % 2) + '0';

        decimal /= 2;

        index++;

    }

    // Pad with leading zeros to make it 3 bits

    while (index < 3) {

        binaryDigits[index] = '0';

        index++;

    }

    // Null-terminate the string

    binaryDigits[3] = '\0';

    // Reverse the binaryDigits array to get the correct order

    char temp;

    for (int i = 0; i < index / 2; i++) {

        temp = binaryDigits[i];

        binaryDigits[i] = binaryDigits[index - i - 1];

        binaryDigits[index - i - 1] = temp;

    }

}

// Function to convert a 4-bit binary number to a hexadecimal digit

char binaryToHexDigit(char binaryDigits[4]) {

    int decimal = 0;

    for (int i = 0; i < 4; i++) {

        decimal = (decimal << 1) | (binaryDigits[i] - '0');

    }

    if (decimal >= 0 && decimal <= 9) {

        return (char)(decimal + '0');

    } else {

        return (char)(decimal - 10 + 'A');

    }

}

int main() {

    char octal[32];

    char binary[96];  // Maximum 32 octal digits can be converted to 96 binary digits

    char hex[32];     // Maximum 32 octal digits can be converted to 8 hexadecimal digits

    int length, binaryDigitCount = 0, hexDigitCount = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter an octal number: ");

    scanf("%s", octal);

    // Calculate the length of the octal number

    length = strlen(octal);

    // Convert octal to binary

    for (int i = 0; i < length; i++) {

        char binaryDigits[4];

        octalToBinaryDigit(octal[i], binaryDigits);

        strcat(binary, binaryDigits);

        binaryDigitCount += 3;

    }

    // Make sure the binary length is a multiple of 4 by adding leading zeros if needed

    int remainder = binaryDigitCount % 4;

    if (remainder > 0) {

        for (int i = 0; i < 4 - remainder; i++) {

            binary[binaryDigitCount + i] = '0';

        }

        binaryDigitCount += 4 - remainder;

    }

    // Convert binary to hexadecimal

    for (int i = 0; i < binaryDigitCount; i += 4) {

        char binaryDigits[4];

        for (int j = 0; j < 4; j++) {

            binaryDigits[j] = binary[i + j];

        }

        hex[hexDigitCount] = binaryToHexDigit(binaryDigits);

        hexDigitCount++;

    }

    hex[hexDigitCount] = '\0';

    // Remove leading zeros if any

    int leadingZeroCount = 0;

    while (hex[leadingZeroCount] == '0') {

        leadingZeroCount++;

    }

    // Print the hexadecimal number

    if (hex[leadingZeroCount] == '\0') {

        printf("Hexadecimal equivalent: 0\n");

    } else {

        printf("Hexadecimal equivalent: %s\n", hex + leadingZeroCount);

    }

    return 0;

}

OUTPUT

NISHIT AGARWALEnter an octal number: 200

Hexadecimal equivalent: 400

PS C:\Users\abc\c language>

Programme-67

Write a C program to convert Decimal to Binary number system

INPUT

#include <stdio.h>

int main() {

    int decimal;

    int binary[32]; // To store the binary digits

    int i = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a decimal number: ");

    scanf("%d", &decimal);

    if (decimal < 0) {

        printf("Binary representation is not defined for negative numbers.\n");

        return 1;

    }

    // Convert decimal to binary

    if (decimal == 0) {

        binary[i] = 0;

        i++;

    } else {

        while (decimal > 0) {

            binary[i] = decimal % 2;

            decimal = decimal / 2;

            i++;

        }

    }

    // Print the binary representation

    printf("Binary equivalent: ");

    if (i == 0) {

        printf("0"); // Handle the case when the input is 0

    } else {

        for (int j = i - 1; j >= 0; j--) {

            printf("%d", binary[j]);

        }

    }

    printf("\n");

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a decimal number: 345

Binary equivalent: 101011001

PS C:\Users\abc\c language>

Programme-68

Write a C program to convert Decimal to Octal number system

INPUT

#include <stdio.h>

int main() {

    int decimal;

    int octal[32]; // To store the octal digits

    int i = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a decimal number: ");

    scanf("%d", &decimal);

    if (decimal < 0) {

        printf("Octal representation is not defined for negative numbers.\n");

        return 1;

    }

    // Convert decimal to octal

    if (decimal == 0) {

        octal[i] = 0;

        i++;

    } else {

        while (decimal > 0) {

            octal[i] = decimal % 8;

            decimal = decimal / 8;

            i++;

        }

    }

    // Print the octal representation

    printf("Octal equivalent: ");

    if (i == 0) {

        printf("0"); // Handle the case when the input is 0

    } else {

        for (int j = i - 1; j >= 0; j--) {

            printf("%d", octal[j]);

        }

    }

    printf("\n");

    return 0;

}

OUTPUT

NISHIT AGARWALEnter a decimal number: 345

Octal equivalent: 531

PS C:\Users\abc\c language>

**Programme-69**

**Write a C program to convert Decimal to Hexadecimal number system**

**INPUT**

#include <stdio.h>

int main() {

    int decimal;

    char hexadecimal[32]; // To store the hexadecimal digits

    int i = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a decimal number: ");

    scanf("%d", &decimal);

    if (decimal < 0) {

        printf("Hexadecimal representation is not defined for negative numbers.\n");

        return 1;

    }

    // Convert decimal to hexadecimal

    if (decimal == 0) {

        hexadecimal[i] = '0';

        i++;

    } else {

        while (decimal > 0) {

            int remainder = decimal % 16;

            if (remainder < 10) {

                hexadecimal[i] = remainder + '0';

            } else {

                hexadecimal[i] = remainder - 10 + 'A';

            }

            decimal = decimal / 16;

            i++;

        }

    }

    // Print the hexadecimal representation

    printf("Hexadecimal equivalent: ");

    if (i == 0) {

        printf("0"); // Handle the case when the input is 0

    } else {

        for (int j = i - 1; j >= 0; j--) {

            printf("%c", hexadecimal[j]);

        }

    }

    printf("\n");

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a decimal number: 345**

**Hexadecimal equivalent: 159**

**PS C:\Users\abc\c language>**

Programme-70

Write a C program to convert Hexadecimal to Binary number system

INPUT

#include <stdio.h>

#include <string.h>

// Function to convert a hexadecimal digit to its binary representation

void hexToBinaryDigit(char hexDigit, char binaryDigits[5]) {

    int decimal;

    if (hexDigit >= '0' && hexDigit <= '9') {

        decimal = hexDigit - '0';

    } else if (hexDigit >= 'A' && hexDigit <= 'F') {

        decimal = hexDigit - 'A' + 10;

    } else if (hexDigit >= 'a' && hexDigit <= 'f') {

        decimal = hexDigit - 'a' + 10;

    } else {

        printf("Invalid hexadecimal digit: %c\n", hexDigit);

        return;

    }

    // Convert decimal to binary

    int index = 0;

    while (decimal > 0) {

        binaryDigits[index] = (decimal % 2) + '0';

        decimal /= 2;

        index++;

    }

    // Pad with leading zeros to make it 4 bits

    while (index < 4) {

        binaryDigits[index] = '0';

        index++;

    }

    // Null-terminate the string

    binaryDigits[4] = '\0';

    // Reverse the binaryDigits array to get the correct order

    char temp;

    for (int i = 0; i < index / 2; i++) {

        temp = binaryDigits[i];

        binaryDigits[i] = binaryDigits[index - i - 1];

        binaryDigits[index - i - 1] = temp;

    }

}

int main() {

    char hexadecimal[32];

    char binary[128];  // Maximum 32 hexadecimal digits can be converted to 128 binary digits

    int length, binaryDigitCount = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a hexadecimal number: ");

    scanf("%s", hexadecimal);

    // Calculate the length of the hexadecimal number

    length = strlen(hexadecimal);

    // Convert hexadecimal to binary

    for (int i = 0; i < length; i++) {

        char binaryDigits[5];

        hexToBinaryDigit(hexadecimal[i], binaryDigits);

        strcat(binary, binaryDigits);

        binaryDigitCount += 4;

    }

    // Print the binary representation

    printf("Binary equivalent: %s\n", binary);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a hexadecimal number: ABC**

**Binary equivalent: 101010111100**

**PS C:\Users\abc\c language>**

Programme-71

Write a C program to convert Hexadecimal to Octal number system

INPUT

#include <stdio.h>

#include <string.h>

// Function to convert a hexadecimal digit to its 4-bit binary representation

void hexToBinaryDigit(char hexDigit, char binaryDigits[5]) {

    int decimal;

    if (hexDigit >= '0' && hexDigit <= '9') {

        decimal = hexDigit - '0';

    } else if (hexDigit >= 'A' && hexDigit <= 'F') {

        decimal = hexDigit - 'A' + 10;

    } else if (hexDigit >= 'a' && hexDigit <= 'f') {

        decimal = hexDigit - 'a' + 10;

    } else {

        printf("Invalid hexadecimal digit: %c\n", hexDigit);

        return;

    }

    // Convert decimal to binary

    int index = 0;

    while (decimal > 0) {

        binaryDigits[index] = (decimal % 2) + '0';

        decimal /= 2;

        index++;

    }

    // Pad with leading zeros to make it 4 bits

    while (index < 4) {

        binaryDigits[index] = '0';

        index++;

    }

    // Null-terminate the string

    binaryDigits[4] = '\0';

    // Reverse the binaryDigits array to get the correct order

    char temp;

    for (int i = 0; i < index / 2; i++) {

        temp = binaryDigits[i];

        binaryDigits[i] = binaryDigits[index - i - 1];

        binaryDigits[index - i - 1] = temp;

    }

}

int main() {

    char hexadecimal[32];

    char binary[128];  // Maximum 32 hexadecimal digits can be converted to 128 binary digits

    char octal[128];   // Maximum 32 hexadecimal digits can be converted to 42 octal digits

    int length, binaryDigitCount = 0, octalDigitCount = 0;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a hexadecimal number: ");

    scanf("%s", hexadecimal);

    // Calculate the length of the hexadecimal number

    length = strlen(hexadecimal);

    // Convert hexadecimal to binary

    for (int i = 0; i < length; i++) {

        char binaryDigits[5];

        hexToBinaryDigit(hexadecimal[i], binaryDigits);

        strcat(binary, binaryDigits);

        binaryDigitCount += 4;

    }

    // Make sure the binary length is a multiple of 3 by adding leading zeros if needed

    int remainder = binaryDigitCount % 3;

    if (remainder > 0) {

        for (int i = 0; i < 3 - remainder; i++) {

            binary[binaryDigitCount + i] = '0';

        }

        binaryDigitCount += 3 - remainder;

    }

    // Convert binary to octal

    for (int i = 0; i < binaryDigitCount; i += 3) {

        char binaryDigits[4];

        for (int j = 0; j < 3; j++) {

            binaryDigits[j] = binary[i + j];

        }

        binaryDigits[3] = '\0';

        int decimal = 0;

        for (int j = 0; j < 3; j++) {

            decimal = (decimal << 1) | (binaryDigits[j] - '0');

        }

        octal[octalDigitCount] = decimal + '0';

        octalDigitCount++;

    }

    octal[octalDigitCount] = '\0';

    // Remove leading zeros if any

    int leadingZeroCount = 0;

    while (octal[leadingZeroCount] == '0') {

        leadingZeroCount++;

    }

    // Print the octal representation

    if (octal[leadingZeroCount] == '\0') {

        printf("Octal equivalent: 0\n");

    } else {

        printf("Octal equivalent: %s\n", octal + leadingZeroCount);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a hexadecimal number: ABC**

**Octal equivalent: 5274**

**PS C:\Users\abc\c language>**

Programme-72

Write a C program to convert Hexadecimal to Decimal number system.

INPUT

#include <stdio.h>

#include <string.h>

// Function to convert a hexadecimal digit to its decimal value

int hexToDecimalDigit(char hexDigit) {

    if (hexDigit >= '0' && hexDigit <= '9') {

        return hexDigit - '0';

    } else if (hexDigit >= 'A' && hexDigit <= 'F') {

        return hexDigit - 'A' + 10;

    } else if (hexDigit >= 'a' && hexDigit <= 'f') {

        return hexDigit - 'a' + 10;

    } else {

        printf("Invalid hexadecimal digit: %c\n", hexDigit);

        return -1; // Return an error value

    }

}

int main() {

    char hexadecimal[32];

    int decimal = 0;

    int length;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a hexadecimal number: ");

    scanf("%s", hexadecimal);

    // Calculate the length of the hexadecimal number

    length = strlen(hexadecimal);

    // Convert hexadecimal to decimal

    for (int i = length - 1, power = 0; i >= 0; i--, power++) {

        int digitValue = hexToDecimalDigit(hexadecimal[i]);

        if (digitValue == -1) {

            // Handle invalid digit

            return 1;

        }

        decimal += digitValue \* (1 << (4 \* power)); // Equivalent to 16^power

    }

    // Print the decimal number

    printf("Decimal equivalent: %d\n", decimal);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a hexadecimal number: ABC**

**Decimal equivalent: 2748**

**PS C:\Users\abc\c language**

**Pattern Exercises**

1. Star pattern programs - Write a C program to print the given star patterns.

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

Pyramid Star Pattern

\*

\* \*

\* \*

\* \*

\*\*\*\*\*\*\*\*\*

Hollow Pyramid Star Pattern

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

Inverted Pyramid Star Pattern

\*\*\*\*\*\*\*\*\*

\* \*

\* \*

\* \*

\*

Hollow Inverted Pyramid Star Pattern

\*

\*\*

\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

Half Diamond Star Pattern

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

Mirrored Half Diamond Star Pattern

**INPUT**

#include <stdio.h>

// Function to print a star pattern

void printStarPattern(int n)

{

    printf("NISHIT AGARWAL");

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= i; j++)

        {

            printf("\*");

        }

        printf("\n");

    }

}

// Function to print a hollow star pattern

void printHollowStarPattern(int n) {

    for (int i = 1; i <= n; i++) {

        if (i == 1 || i == n) {

            for (int j = 1; j <= n; j++) {

                printf("\*");

            }

        } else {

            printf("\*");

        }

        printf("\n");

    }

}

// Function to print an inverted star pattern

void printInvertedStarPattern(int n) {

    for (int i = n; i >= 1; i--) {

        for (int j = 1; j <= i; j++) {

            printf("\*");

        }

        printf("\n");

    }

}

// Function to print a hollow inverted star pattern

void printHollowInvertedStarPattern(int n) {

    for (int i = n; i >= 1; i--) {

        if (i == 1 || i == n) {

            for (int j = 1; j <= i; j++) {

                printf("\*");

            }

        } else {

            printf("\*");

        }

        printf("\n");

    }

}

// Function to print a half diamond star pattern

void printHalfDiamondStarPattern(int n) {

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= i; j++) {

            printf("\*");

        }

        printf("\n");

    }

    for (int i = n - 1; i >= 1; i--) {

        for (int j = 1; j <= i; j++) {

            printf("\*");

        }

        printf("\n");

    }

}

int main() {

    int n;

    // Ask the user for input

    printf("Enter the value of n: ");

    scanf("%d", &n);

    printf("Star Pattern:\n");

    printStarPattern(n);

    printf("Hollow Star Pattern:\n");

    printHollowStarPattern(n);

    printf("Inverted Star Pattern:\n");

    printInvertedStarPattern(n);

    printf("Hollow Inverted Star Pattern:\n");

    printHollowInvertedStarPattern(n);

    printf("Half Diamond Star Pattern:\n");

    printHalfDiamondStarPattern(n);

    return 0;

}

**OUTPUT**

**Enter the value of n: 5**

**Star Pattern:**

**NISHIT AGARWAL\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

**Hollow Star Pattern:**

**\*\*\*\*\***

**\***

**\***

**\***

**\*\*\*\*\***

**Inverted Star Pattern:**

**\*\*\*\*\***

**\*\*\*\***

**\*\*\***

**\*\***

**\***

**Hollow Inverted Star Pattern:**

**\*\*\*\*\***

**\***

**\***

**\***

**\***

**Half Diamond Star Pattern:**

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

**\*\*\*\***

**\*\*\***

**\*\***

**\***

**PS C:\Users\abc\c language>**

Number pattern programs - Write a C program to print the given number patterns

**Square number patterns**

11111

11111

11111

11111

11111

**Number pattern 1**

11111

00000

11111

00000

11111

**Number pattern 2**

01010

01010

01010

0101001010

**Number pattern 3**

11111

10001

10001

10001

11111

**Number pattern 4**

11111

11111

11011

11111

11111

**Number pattern 5**

10101

01010

10101

01010

10101

**INPUT**

#include <stdio.h>

// Function to print a square pattern of 1s

void printSquarePattern(int n)

{

    printf("NISHIT AGARWAL");

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= n; j++) {

            printf("1");

        }

        printf("\n");

    }

}

// Function to print a square pattern of alternating 1s and 0s

void printNumberPattern1(int n) {

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= n; j++) {

            if ((i + j) % 2 == 0) {

                printf("1");

            } else {

                printf("0");

            }

        }

        printf("\n");

    }

}

// Function to print a square pattern with a 1 in the center

void printNumberPattern2(int n) {

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= n; j++) {

            if (i == n / 2 + 1 && j == n / 2 + 1) {

                printf("1");

            } else {

                printf("0");

            }

        }

        printf("\n");

    }

}

// Function to print a square pattern with a 1 on the diagonals

void printNumberPattern3(int n) {

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= n; j++) {

            if (i == j || i + j == n + 1) {

                printf("1");

            } else {

                printf("0");

            }

        }

        printf("\n");

    }

}

// Function to print a square pattern with a 1 in a cross pattern

void printNumberPattern4(int n) {

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= n; j++) {

            if (i == n / 2 + 1 || j == n / 2 + 1) {

                printf("1");

            } else {

                printf("0");

            }

        }

        printf("\n");

    }

}

// Function to print a square pattern of alternating 1s and 0s

void printNumberPattern5(int n) {

    for (int i = 1; i <= n; i++) {

        for (int j = 1; j <= n; j++) {

            if ((i + j) % 2 == 0) {

                printf("1");

            } else {

                printf("0");

            }

        }

        printf("\n");

    }

}

int main() {

    int n;

    // Ask the user for input

    printf("Enter the value of n: ");

    scanf("%d", &n);

    printf("Square number pattern:\n");

    printSquarePattern(n);

    printf("Number pattern 1:\n");

    printNumberPattern1(n);

    printf("Number pattern 2:\n");

    printNumberPattern2(n);

    printf("Number pattern 3:\n");

    printNumberPattern3(n);

    printf("Number pattern 4:\n");

    printNumberPattern4(n);

    printf("Number pattern 5:\n");

    printNumberPattern5(n);

    return 0;

}

**OUTPUT**

**Enter the value of n: 5**

**Square number pattern:**

**NISHIT AGARWAL**

**11111**

**11111**

**11111**

**11111**

**11111**

**Number pattern 1:**

**10101**

**01010**

**10101**

**01010**

**10101**

**Number pattern 2:**

**00000**

**00000**

**00100**

**00000**

**00000**

**Number pattern 3:**

**10001**

**01010**

**00100**

**01010**

**10001**

**Number pattern 4:**

**00100**

**00100**

**11111**

**00100**

**00100**

**Number pattern 5:**

**10101**

**01010**

**10101**

**01010**

**10101**

**PS C:\Users\abc\c language>**

**If…Else Exercises**

**Programme-1**

**Write a C program to find maximum between two numbers**

**INPUT**

#include <stdio.h>

int main() {

    int num1, num2;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the first number: ");

    scanf("%d", &num1);

    printf("Enter the second number: ");

    scanf("%d", &num2);

    // Find the maximum number

    int max = (num1 > num2) ? num1 : num2;

    // Print the maximum number

    printf("The maximum number is: %d\n", max);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the first number: 34**

**Enter the second number: 23**

**The maximum number is: 34**

**PS C:\Users\abc\c language>**

**Programme-2**

**Write a C program to find maximum between three numbers**

**INPUT**

#include <stdio.h>

int main() {

    int num1, num2, num3;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the first number: ");

    scanf("%d", &num1);

    printf("Enter the second number: ");

    scanf("%d", &num2);

    printf("Enter the third number: ");

    scanf("%d", &num3);

    // Find the maximum number

    int max = num1;

    if (num2 > max) {

        max = num2;

    }

    if (num3 > max) {

        max = num3;

    }

    // Print the maximum number

    printf("The maximum number is: %d\n", max);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the first number: 34**

**Enter the second number: 35**

**Enter the third number: 36**

**The maximum number is: 36**

**PS C:\Users\abc\c language>**

**Programme-3**

**Write a C program to check whether a number is negative, positive or zero**

**INPUT**

#include <stdio.h>

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    // Check if the number is negative, positive, or zero

    if (number < 0) {

        printf("The number is negative.\n");

    } else if (number > 0) {

        printf("The number is positive.\n");

    } else {

        printf("The number is zero.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a number: 45**

**The number is positive.**

**PS C:\Users\abc\c language>**

**Programme-4**

**Write a C program to check whether a number is divisible by 5 and 11 or not**

**INPUT**

#include <stdio.h>

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    // Check if the number is divisible by both 5 and 11

    if (number % 5 == 0 && number % 11 == 0) {

        printf("The number is divisible by both 5 and 11.\n");

    } else {

        printf("The number is not divisible by both 5 and 11.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a number: 55**

**The number is divisible by both 5 and 11.**

**PS C:\Users\abc\c language>**

**Programme-5**

**Write a C program to check whether a number is even or odd**

**INPUT**

#include <stdio.h>

int main() {

    int number;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a number: ");

    scanf("%d", &number);

    // Check if the number is even or odd

    if (number % 2 == 0) {

        printf("The number is even.\n");

    } else {

        printf("The number is odd.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a number: 56**

**The number is even.**

**PS C:\Users\abc\c language>**

**Programme-6**

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

**INPUT**

#include <stdio.h>

int main() {

    int year;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a year: ");

    scanf("%d", &year);

    // Check if the year is a leap year

    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

        printf("%d is a leap year.\n", year);

    } else {

        printf("%d is not a leap year.\n", year);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a year: 2019**

**2019 is not a leap year.**

**PS C:\Users\abc\c language>**

**Programme-7**

**Write a C program to check whether a character is alphabet or not**

**INPUT**

#include <stdio.h>

int main() {

    char character;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a character: ");

    scanf(" %c", &character); // Note the space before %c to consume any leading whitespace

    // Check if the character is an alphabet

    if ((character >= 'A' && character <= 'Z') || (character >= 'a' && character <= 'z')) {

        printf("%c is an alphabet.\n", character);

    } else {

        printf("%c is not an alphabet.\n", character);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a character: A**

**A is an alphabet.**

**PS C:\Users\abc\c language>**

**Programme-8**

**Write a C program to input any alphabet and check whether it is vowel or consonant**

**INPUT**

#include <stdio.h>

int main() {

    char character;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a character: ");

    scanf(" %c", &character); // Note the space before %c to consume any leading whitespace

    // Check if the character is a vowel or consonant

    if ((character >= 'A' && character <= 'Z') || (character >= 'a' && character <= 'z')) {

        // Convert the character to uppercase for consistent vowel checking

        char uppercaseCharacter = (character >= 'a' && character <= 'z') ? character - 32 : character;

        if (uppercaseCharacter == 'A' || uppercaseCharacter == 'E' || uppercaseCharacter == 'I' || uppercaseCharacter == 'O' || uppercaseCharacter == 'U') {

            printf("%c is a vowel.\n", character);

        } else {

            printf("%c is a consonant.\n", character);

        }

    } else {

        printf("Invalid input. Please enter an alphabet.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a character: A**

**A is a vowel.**

**PS C:\Users\abc\c language>**

**Programme-9**

**Write a C program to input any character and check whether it is alphabet, digit or special**

**character.**

**INPUT**

#include <stdio.h>

int main() {

    char character;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a character: ");

    scanf(" %c", &character); // Note the space before %c to consume any leading whitespace

    // Check whether the character is an alphabet, digit, or special character

    if ((character >= 'A' && character <= 'Z') || (character >= 'a' && character <= 'z')) {

        printf("%c is an alphabet.\n", character);

    } else if (character >= '0' && character <= '9') {

        printf("%c is a digit.\n", character);

    } else {

        printf("%c is a special character.\n", character);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a character: A**

**A is an alphabet.**

**PS C:\Users\abc\c language>**

**Programme-10**

**Write a C program to check whether a character is uppercase or lowercase alphabet**

**INPUT**

#include <stdio.h>

int main() {

    char character;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a character: ");

    scanf(" %c", &character); // Note the space before %c to consume any leading whitespace

    // Check whether the character is an uppercase or lowercase alphabet

    if (character >= 'A' && character <= 'Z') {

        printf("%c is an uppercase alphabet.\n", character);

    } else if (character >= 'a' && character <= 'z') {

        printf("%c is a lowercase alphabet.\n", character);

    } else {

        printf("%c is not an alphabet.\n", character);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a character: N**

**N is an uppercase alphabet.**

**PS C:\Users\abc\c language>**

**Programme-11**

**Write a C program to input week number and print week day**

**INPUT**

#include <stdio.h>

int main() {

    int weekNumber;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a week number (1 to 7): ");

    scanf("%d", &weekNumber);

    // Check and print the corresponding weekday

    switch (weekNumber) {

        case 1:

            printf("Monday\n");

            break;

        case 2:

            printf("Tuesday\n");

            break;

        case 3:

            printf("Wednesday\n");

            break;

        case 4:

            printf("Thursday\n");

            break;

        case 5:

            printf("Friday\n");

            break;

        case 6:

            printf("Saturday\n");

            break;

        case 7:

            printf("Sunday\n");

            break;

        default:

            printf("Invalid week number. Please enter a number between 1 and 7.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a week number (1 to 7): 6**

**Saturday**

**PS C:\Users\abc\c language>**

**Programme-12 OR 14**

**Write a C program to input month number and print number of days in that month**

**INPUT**

#include <stdio.h>

int main() {

    int monthNumber;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter a month number (1 to 12): ");

    scanf("%d", &monthNumber);

    // Check and print the number of days in the given month

    if (monthNumber >= 1 && monthNumber <= 12) {

        int daysInMonth;

        switch (monthNumber) {

            case 4:

            case 6:

            case 9:

            case 11:

                daysInMonth = 30;

                break;

            case 2:

                // Check for leap year

                printf("Enter a year (e.g., 2023): ");

                int year;

                scanf("%d", &year);

                if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

                    daysInMonth = 29; // February in a leap year

                } else {

                    daysInMonth = 28; // February in a non-leap year

                }

                break;

            default:

                daysInMonth = 31;

                break;

        }

        printf("Number of days in month %d: %d\n", monthNumber, daysInMonth);

    } else {

        printf("Invalid month number. Please enter a number between 1 and 12.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter a month number (1 to 12): 11**

**Number of days in month 11: 30**

**PS C:\Users\abc\c language>**

**Programme-13 OR 15**

**Write a C program to count total number of notes in given amount**

**INPUT**

#include <stdio.h>

int main() {

    int amount, notesCount = 0;

    int denominations[] = {500, 100, 50, 20, 10, 5, 1};

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the amount: ");

    scanf("%d", &amount);

    if (amount <= 0) {

        printf("Invalid amount. Please enter a positive amount.\n");

    } else {

        printf("Count of each denomination:\n");

        for (int i = 0; i < 7; i++) {

            int count = amount / denominations[i];

            if (count > 0) {

                printf("%d x %d\n", denominations[i], count);

                notesCount += count;

                amount %= denominations[i];

            }

        }

        printf("Total number of notes: %d\n", notesCount);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the amount: 5555**

**Count of each denomination:**

**500 x 11**

**50 x 1**

**5 x 1**

**Total number of notes: 13**

**PS C:\Users\abc\c language>**

**Programme-16**

**Write a C program to input angles of a triangle and check whether triangle is valid or not**

**INPUT**

#include <stdio.h>

int main() {

    int angle1, angle2, angle3;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the first angle: ");

    scanf("%d", &angle1);

    printf("Enter the second angle: ");

    scanf("%d", &angle2);

    printf("Enter the third angle: ");

    scanf("%d", &angle3);

    // Check and print whether the triangle is valid

    if (angle1 + angle2 + angle3 == 180) {

        printf("The triangle is valid.\n");

    } else {

        printf("The triangle is not valid.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the first angle: 56**

**Enter the second angle: 56**

**Enter the third angle: 68**

**The triangle is valid.**

**PS C:\Users\abc\c language>**

**Programme-17**

**Write a C program to input all sides of a triangle and check whether triangle is valid or not**

**INPUT**

#include <stdio.h>

int main() {

    double side1, side2, side3;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the length of the first side: ");

    scanf("%lf", &side1);

    printf("Enter the length of the second side: ");

    scanf("%lf", &side2);

    printf("Enter the length of the third side: ");

    scanf("%lf", &side3);

    // Check and print whether the triangle is valid

    if (side1 + side2 > side3 && side1 + side3 > side2 && side2 + side3 > side1) {

        printf("The triangle is valid.\n");

    } else {

        printf("The triangle is not valid.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the length of the first side: 3**

**Enter the length of the second side: 4**

**Enter the length of the third side: 5**

**The triangle is valid.**

**PS C:\Users\abc\c language>**

**Programme-18**

**Write a C program to check whether the triangle is equilateral, isosceles or scalene triangle**

**INPUT**

#include <stdio.h>

int main() {

    double side1, side2, side3;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the length of the first side: ");

    scanf("%lf", &side1);

    printf("Enter the length of the second side: ");

    scanf("%lf", &side2);

    printf("Enter the length of the third side: ");

    scanf("%lf", &side3);

    // Check and print the type of triangle

    if (side1 == side2 && side2 == side3) {

        printf("It's an equilateral triangle.\n");

    } else if (side1 == side2 || side1 == side3 || side2 == side3) {

        printf("It's an isosceles triangle.\n");

    } else {

        printf("It's a scalene triangle.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the length of the first side: 4**

**Enter the length of the second side: 5**

**Enter the length of the third side: 3**

**It's a scalene triangle.**

**PS C:\Users\abc\c language>**

**Programme-19**

**Write a C program to find all roots of a quadratic equation**

**INPUT**

#include<stdio.h>

#include<math.h>

int main()

{

    float a, b, c, discriminant, root1, root2;

    printf("NISHIT AGARWAL");

    printf("input values of a, b, and c\n");

    scanf("%f %f %f", &a, &b, &c);

    discriminant = b\*b - 4\*a\*c;

    if(discriminant < 0)

    printf("\n\nroots are imaginary");

    else

    {

        root1 = (-b + sqrt(discriminant))/(2.0\*a);

        root2 = (-b - sqrt(discriminant))/(2.0\*a);

        printf("\n\nroot1 = %.2f\n\nroot2 = %.2f\n", root1,root2);

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALinput values of a, b, and c**

**1**

**2**

**0**

**root1 = 0.00**

**root2 = -2.00**

**PS C:\Users\abc\c language>**

**Programme-20**

**Write a C program to calculate profit or loss**

**INPUT**

#include <stdio.h>

int main() {

    float costPrice, sellingPrice;

    float profit, loss;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the cost price: ");

    scanf("%f", &costPrice);

    printf("Enter the selling price: ");

    scanf("%f", &sellingPrice);

    // Calculate profit or loss

    if (sellingPrice > costPrice) {

        profit = sellingPrice - costPrice;

        printf("You made a profit of %.2f\n", profit);

    } else if (costPrice > sellingPrice) {

        loss = costPrice - sellingPrice;

        printf("You incurred a loss of %.2f\n", loss);

    } else {

        printf("You neither made a profit nor incurred a loss.\n");

    }

    return 0;

}

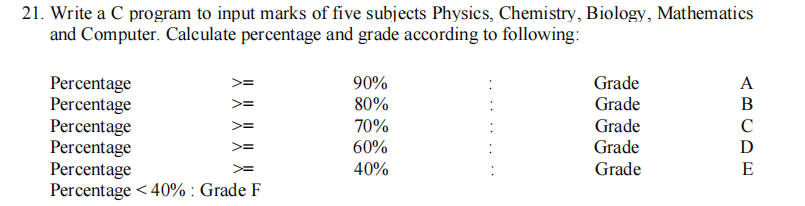
**OUTPUT**

**NISHIT AGARWALEnter the cost price: 4**

**Enter the selling price: 78**

**You made a profit of 74.00**

**PS C:\Users\abc\c language>**



**INPUT**

#include <stdio.h>

int main() {

    float physics, chemistry, biology, mathematics, computer;

    float totalMarks, percentage;

    char grade;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter marks in Physics: ");

    scanf("%f", &physics);

    printf("Enter marks in Chemistry: ");

    scanf("%f", &chemistry);

    printf("Enter marks in Biology: ");

    scanf("%f", &biology);

    printf("Enter marks in Mathematics: ");

    scanf("%f", &mathematics);

    printf("Enter marks in Computer: ");

    scanf("%f", &computer);

    // Calculate the total marks and percentage

    totalMarks = physics + chemistry + biology + mathematics + computer;

    percentage = (totalMarks / 500) \* 100;

    // Determine the grade based on the percentage

    if (percentage >= 90) {

        grade = 'A';

    } else if (percentage >= 80) {

        grade = 'B';

    } else if (percentage >= 70) {

        grade = 'C';

    } else if (percentage >= 60) {

        grade = 'D';

    } else if (percentage >= 40) {

        grade = 'E';

    } else {

        grade = 'F';

    }

    // Print the percentage and grade

    printf("Percentage: %.2f%%\n", percentage);

    printf("Grade: %c\n", grade);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter marks in Physics: 85**

**Enter marks in Chemistry: 86**

**Enter marks in Biology: 87**

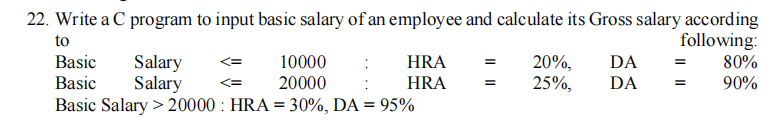
**Enter marks in Mathematics: 88**

**Enter marks in Computer: 89**

**Percentage: 87.00%**

**Grade: B**

**PS C:\Users\abc\c language>**



**INPUT**

#include <stdio.h>

int main() {

    float basicSalary, grossSalary;

    float hra, da;

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the basic salary: ");

    scanf("%f", &basicSalary);

    // Calculate HRA and DA based on the provided conditions

    if (basicSalary <= 10000) {

        hra = 0.2 \* basicSalary;

        da = 0.8 \* basicSalary;

    } else if (basicSalary <= 20000) {

        hra = 0.25 \* basicSalary;

        da = 0.9 \* basicSalary;

    } else {

        hra = 0.3 \* basicSalary;

        da = 0.95 \* basicSalary;

    }

    // Calculate the gross salary

    grossSalary = basicSalary + hra + da;

    // Print the gross salary

    printf("Gross Salary: %.2f\n", grossSalary);

    return 0;

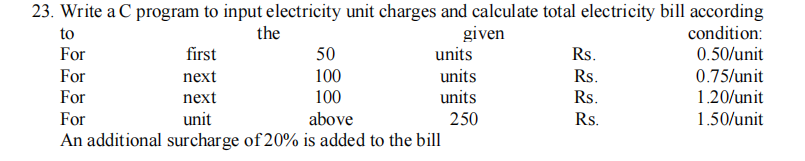
}

**OUTPUT**

**NISHIT AGARWALEnter the basic salary: 45000**

**Gross Salary: 101250.00**

**PS C:\Users\abc\c language>**



**INPUT**

#include <stdio.h>

int main() {

    float unitCharges, totalBill;

    float surcharge = 0.20;  // 20% surcharge

    // Ask the user for input

    printf("NISHIT AGARWAL");

    printf("Enter the electricity unit charges: ");

    scanf("%f", &unitCharges);

    // Calculate the total electricity bill based on the given conditions

    if (unitCharges <= 50) {

        totalBill = unitCharges \* 0.50;

    } else if (unitCharges <= 150) {

        totalBill = (50 \* 0.50) + ((unitCharges - 50) \* 0.75);

    } else if (unitCharges <= 250) {

        totalBill = (50 \* 0.50) + (100 \* 0.75) + ((unitCharges - 150) \* 1.20);

    } else {

        totalBill = (50 \* 0.50) + (100 \* 0.75) + (100 \* 1.20) + ((unitCharges - 250) \* 1.50);

    }

    // Add the surcharge to the total bill

    totalBill += surcharge \* totalBill;

    // Print the total electricity bill

    printf("Total Electricity Bill: Rs. %.2f\n", totalBill);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the electricity unit charges: 8**

**Total Electricity Bill: Rs. 4.80**

**PS C:\Users\abc\c language>**

Programme-24

**Write a C program to convert specified days into years, weeks and days.**

INPUT

#include <stdio.h>

int main() {

    int days, years, weeks, remainingDays;

    // Ask the user for input

    printf("NISHIT AGARWAL");

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

    scanf("%d", &days);

    // Calculate years, weeks, and remaining days

    years = days / 365;

    weeks = (days % 365) / 7;

    remainingDays = days - (years \* 365) - (weeks \* 7);

    // Print the result

    printf("%d days is equivalent to:\n", days);

    printf("Years: %d\n", years);

    printf("Weeks: %d\n", weeks);

    printf("Days: %d\n", remainingDays);

    return 0;

}

OUTPUT

NISHIT AGARWALEnter the number of days: 45

45 days is equivalent to:

Years: 0

Weeks: 6

Days: 3

PS C:\Users\abc\c language>

UNIT 3

ARRAY

PROGRAMME 1

Write a program in C to read n number of values in an array and display them in reverse order

INPUT

#include <stdio.h>

int main() {

int n;

  printf("NISHIT AGARWAL");

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

    scanf("%d", &n);

    if (n <= 0) {

        printf("Please enter a positive value for n.\n");

        return 1;

    }

    int arr[n];

    printf("Enter %d values:\n", n);

    // Read n values into the array

    for (int i = 0; i < n; i++) {

        scanf("%d", &arr[i]);

    }

    printf("Values in reverse order:\n");

    // Display the values in reverse order

    for (int i = n - 1; i >= 0; i--) {

        printf("%d ", arr[i]);

    }

    printf("\n");

    return 0;

}

OUTPUT

NISHIT AGARWALEnter the number of values: 5

Enter 5 values:

1

2

3

4

5

Values in reverse order:

5 4 3 2 1

PS C:\Users\abc\c language>

PROGRAMME 2

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

**INPUT**

#include <stdio.h>

int main() {

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter the number of elements in the array: ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Please enter a positive value for the number of elements.\n");

        return 1;

    }

    int arr[n];

    int sum = 0;

    printf("Enter %d elements:\n", n);

    // Read n elements into the array and calculate the sum

    for (int i = 0; i < n; i++) {

        scanf("%d", &arr[i]);

        sum += arr[i];

    }

    printf("Sum of all elements in the array: %d\n", sum);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the number of elements in the array: 5**

**Enter 5 elements:**

**1**

**2**

**3**

**4**

**5**

**Sum of all elements in the array: 15**

**PS C:\Users\abc\c language>**

**PROGRAMME 3**

Write a program in C to copy the elements of one array into another array

**INPUT**

#include <stdio.h>

int main() {

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter the number of elements in the source array: ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Please enter a positive value for the number of elements.\n");

        return 1;

    }

    int source[n];

    int destination[n];

    printf("Enter %d elements for the source array:\n", n);

    // Read n elements into the source array

    for (int i = 0; i < n; i++) {

        scanf("%d", &source[i]);

    }

    // Copy elements from source to destination

    for (int i = 0; i < n; i++) {

        destination[i] = source[i];}

    printf("Elements copied from source array to destination array:\n");

    // Print elements in the destination array

    for (int i = 0; i < n; i++) {

        printf("%d ", destination[i]);

    }

    printf("\n");

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the number of elements in the source array: 5**

**Enter 5 elements for the source array:**

**1**

**2**

**3**

**4**

**5**

**Elements copied from source array to destination array:**

**1 2 3 4 5**

**PS C:\Users\abc\c language>**

**PROGRAMME 4**

Write a program in C to count the total number of duplicate elements in an array

**INPUT**

#include <stdio.h>

int main() {

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter the number of elements in the array: ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Please enter a positive value for the number of elements.\n");

        return 1;

    }

    int arr[n];

    int count = 0;

    printf("Enter %d elements:\n", n);

    // Read n elements into the array

    for (int i = 0; i < n; i++) {

        scanf("%d", &arr[i]);

    }

    // Compare each element with all other elements

    for (int i = 0; i < n - 1; i++) {

        for (int j = i + 1; j < n; j++) {

            if (arr[i] == arr[j]) {

                count++;

                break; // Break to avoid double-counting duplicates

            }

        }

    }

    printf("Total number of duplicate elements in the array: %d\n", count);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the number of elements in the array: 5**

**Enter 5 elements:**

**1**

**2**

**3**

**4**

**5**

**Total number of duplicate elements in the array: 0**

**PS C:\Users\abc\c language>**

**PROGRAMME 5**

Write a program in C to find the maximum and minimum elements in an array.

**INPUT**

#include <stdio.h>

int main() {

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter the number of elements in the array: ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Please enter a positive value for the number of elements.\n");

        return 1;

    }

    int arr[n];

    printf("Enter %d elements:\n", n);

    // Read n elements into the array

    for (int i = 0; i < n; i++) {

        scanf("%d", &arr[i]);

    }

    int max = arr[0]; // Initialize max to the first element

    int min = arr[0]; // Initialize min to the first element

    // Find the maximum and minimum elements

    for (int i = 1; i < n; i++) {

        if (arr[i] > max) {

            max = arr[i];

        }

        if (arr[i] < min) {

            min = arr[i];

        }

    }

    printf("Maximum element in the array: %d\n", max);

    printf("Minimum element in the array: %d\n", min);

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the number of elements in the array: 5**

**Enter 5 elements:**

**1**

**2**

**3**

**4**

**5**

**Maximum element in the array: 5**

**Minimum element in the array: 1**

**PS C:\Users\abc\c language>**

**PROGRAMME 6**

Write a C program to sort the elements of an array in descending order.

**INPUT**

#include <stdio.h>

void selectionSort(int arr[], int n) {

    int i, j, maxIndex, temp;

    for (i = 0; i < n - 1; i++) {

        maxIndex = i;

        for (j = i + 1; j < n; j++) {

            if (arr[j] > arr[maxIndex]) {

                maxIndex = j;

            }

        }

        if (maxIndex != i) {

            // Swap arr[i] and arr[maxIndex]

            temp = arr[i];

            arr[i] = arr[maxIndex];

            arr[maxIndex] = temp;

        }

    }

}

int main() {

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter the number of elements in the array: ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Please enter a positive value for the number of elements.\n");

        return 1;

    }

    int arr[n];

    printf("Enter %d elements:\n", n);

    // Read n elements into the array

    for (int i = 0; i < n; i++) {

        scanf("%d", &arr[i]);

    }

    // Sort the array in descending order

    selectionSort(arr, n);

    printf("Array sorted in descending order:\n");

    // Print the sorted array

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

    printf("\n");

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the number of elements in the array: 5**

**Enter 5 elements:**

**1**

**2**

**3**

**4**

**5**

**Array sorted in descending order:**

**5 4 3 2 1**

**PS C:\Users\abc\c language>**

**PROGRAMME 7**

Write a program in C to separate odd and even integers into separate arrays.

**INPUT**

#include <stdio.h>

int main() {

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter the number of elements in the array: ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Please enter a positive value for the number of elements.\n");

        return 1;

    }

    int originalArray[n];

    int evenArray[n];

    int oddArray[n];

    int evenCount = 0;

    int oddCount = 0;

    printf("Enter %d elements:\n", n);

    // Read n elements into the original array

    for (int i = 0; i < n; i++) {

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

    }

    // Separate odd and even integers into separate arrays

    for (int i = 0; i < n; i++) {

        if (originalArray[i] % 2 == 0) {

            evenArray[evenCount] = originalArray[i];

            evenCount++;

        } else {

            oddArray[oddCount] = originalArray[i];

            oddCount++;

        }

    }

    printf("Even integers:\n");

    for (int i = 0; i < evenCount; i++) {

        printf("%d ", evenArray[i]);

    }

    printf("\n");

    printf("Odd integers:\n");

    for (int i = 0; i < oddCount; i++) {

        printf("%d ", oddArray[i]);

    }

    printf("\n");

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the number of elements in the array: 5**

**Enter 5 elements:**

**1**

**2**

**3**

**4**

**5**

**Even integers:**

**2 4**

**Odd integers:**

**1 3 5**

**PS C:\Users\abc\c language>**

**PROGRAMME 8**

Write a program in C to merge two arrays of the same size sorted in descending/ascending order.

**INPUT**

#include <stdio.h>

int main() {

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter the number of elements in each array: ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Please enter a positive value for the number of elements.\n");

        return 1;

    }

    int arr1[n];

    int arr2[n];

    int mergedArray[2 \* n];

    int i, j, k;

    printf("Enter %d elements for the first array (ascending order):\n", n);

    for (i = 0; i < n; i++) {

        scanf("%d", &arr1[i]);

    }

    printf("Enter %d elements for the second array (ascending order):\n", n);

    for (i = 0; i < n; i++) {

        scanf("%d", &arr2[i]);

    }

    // Merge the two arrays in ascending order

    i = 0;  // Index for arr1

    j = 0;  // Index for arr2

    k = 0;  // Index for mergedArray

    while (i < n && j < n) {

        if (arr1[i] < arr2[j]) {

            mergedArray[k] = arr1[i];

            i++;

        } else {

            mergedArray[k] = arr2[j];

            j++;

        }

        k++;

    }

    // Copy the remaining elements from arr1 (if any)

    while (i < n) {

        mergedArray[k] = arr1[i];

        i++;

        k++;

    }

    // Copy the remaining elements from arr2 (if any)

    while (j < n) {

        mergedArray[k] = arr2[j];

        j++;

        k++;

    }

    printf("Merged array in ascending order:\n");

    for (i = 0; i < 2 \* n; i++) {

        printf("%d ", mergedArray[i]);

    }

    printf("\n");

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the number of elements in each array: 5**

**Enter 5 elements for the first array (ascending order):**

**1**

**2**

**3**

**4**

**5**

**Enter 5 elements for the second array (ascending order):**

**6**

**7**

**8**

**9**

**9**

**Merged array in ascending order:**

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

**PS C:\Users\abc\c language>**

**PROGRAMME 9**

Write a program in C to merge two arrays of the same size sorted in descending order

**INPUT**

#include <stdio.h>

int main() {

    int n;

    printf("NISHIT AGARWAL");

    printf("Enter the number of elements in each array: ");

    scanf("%d", &n);

    if (n <= 0) {

        printf("Please enter a positive value for the number of elements.\n");

        return 1;

    }

    int arr1[n];

    int arr2[n];

    int mergedArray[2 \* n];

    int i, j, k;

    printf("Enter %d elements for the first array (descending order):\n", n);

    for (i = 0; i < n; i++) {

        scanf("%d", &arr1[i]);

    }

    printf("Enter %d elements for the second array (descending order):\n", n);

    for (i = 0; i < n; i++) {

        scanf("%d", &arr2[i]);

    }

    // Merge the two arrays in descending order

    i = 0;  // Index for arr1

    j = 0;  // Index for arr2

    k = 0;  // Index for mergedArray

    while (i < n && j < n) {

        if (arr1[i] > arr2[j]) {

            mergedArray[k] = arr1[i];

            i++;

        } else {

            mergedArray[k] = arr2[j];

            j++;

        }

        k++;

    }

    // Copy the remaining elements from arr1 (if any)

    while (i < n) {

        mergedArray[k] = arr1[i];

        i++;

        k++;

    }

    // Copy the remaining elements from arr2 (if any)

    while (j < n) {

        mergedArray[k] = arr2[j];

        j++;

        k++;

    }

    printf("Merged array in descending order:\n");

    for (i = 0; i < 2 \* n; i++) {

        printf("%d ", mergedArray[i]);

    }

    printf("\n");

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the number of elements in each array: 5**

**Enter 5 elements for the first array (descending order):**

**1**

**2**

**3**

**4**

**5**

**Enter 5 elements for the second array (descending order):**

**6**

**7**

**8**

**9**

**8**

**Merged array in descending order:**

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

**PS C:\Users\abc\c language>**

**PROGRAMME 10**

WAP using Switch case: Consider two matrices of the size m and n. Implement matrix operation

and display. Show these things in program 1) Read matrix elements and display 2) Matrix

Multiplication and display 3) addition of matrix and display 4)Subtraction of Matrix and display

5)Transpose of Matrix and display

**INPUT**

#include <stdio.h>

void readMatrix(int matrix[][10], int m, int n) {

    printf("Enter the elements of the matrix:\n");

    for (int i = 0; i < m; i++) {

        for (int j = 0; j < n; j++) {

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

        }

    }

}

void displayMatrix(int matrix[][10], int m, int n) {

    printf("Matrix:\n");

    for (int i = 0; i < m; i++) {

        for (int j = 0; j < n; j++) {

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

        }

        printf("\n");

    }

}

void matrixMultiplication(int A[][10], int B[][10], int C[][10], int m, int n, int p) {

    for (int i = 0; i < m; i++) {

        for (int j = 0; j < p; j++) {

            C[i][j] = 0;

            for (int k = 0; k < n; k++) {

                C[i][j] += A[i][k] \* B[k][j];

            }

        }

    }

}

void matrixAddition(int A[][10], int B[][10], int C[][10], int m, int n) {

    for (int i = 0; i < m; i++) {

        for (int j = 0; j < n; j++) {

            C[i][j] = A[i][j] + B[i][j];

        }

    }

}

void matrixSubtraction(int A[][10], int B[][10], int C[][10], int m, int n) {

    for (int i = 0; i < m; i++) {

        for (int j = 0; j < n; j++) {

            C[i][j] = A[i][j] - B[i][j];

        }

    }

}

void transposeMatrix(int A[][10], int B[][10], int m, int n) {

    for (int i = 0; i < m; i++) {

        for (int j = 0; j < n; j++) {

            B[j][i] = A[i][j];

        }

    }

}

int main() {

    int m, n, p;

    int A[10][10], B[10][10], C[10][10], transpose[10][10];

    printf("NISHIT AGARWAL");

    printf("Enter the number of rows and columns of matrix A: ");

    scanf("%d %d", &m, &n);

    readMatrix(A, m, n);

    printf("Enter the number of rows and columns of matrix B: ");

    scanf("%d %d", &n, &p);

    readMatrix(B, n, p);

    int choice;

    printf("Choose an operation:\n");

    printf("1. Display matrices\n");

    printf("2. Matrix Multiplication\n");

    printf("3. Matrix Addition\n");

    printf("4. Matrix Subtraction\n");

    printf("5. Transpose of Matrix A\n");

    scanf("%d", &choice);

    switch (choice) {

        case 1:

            displayMatrix(A, m, n);

            displayMatrix(B, n, p);

            break;

        case 2:

            if (n != n) {

                printf("Matrix multiplication is not possible.\n");

            } else {

                matrixMultiplication(A, B, C, m, n, p);

                displayMatrix(C, m, p);

            }

            break;

        case 3:

            if (m != n || m != p) {

                printf("Matrix addition is not possible. Matrices must have the same dimensions.\n");

            } else {

                matrixAddition(A, B, C, m, n);

                displayMatrix(C, m, n);

            }

            break;

        case 4:

            if (m != n || m != p) {

                printf("Matrix subtraction is not possible. Matrices must have the same dimensions.\n");

            } else {

                matrixSubtraction(A, B, C, m, n);

                displayMatrix(C, m, n);

            }

            break;

        case 5:

            transposeMatrix(A, transpose, m, n);

            displayMatrix(transpose, n, m);

            break;

        default:

            printf("Invalid choice.\n");

    }

    return 0;

}

**OUTPUT**

**NISHIT AGARWALEnter the number of rows and columns of matrix A: 2**

**2**

**Enter the elements of the matrix:**

**1**

**2**

**3**

**4**

**Enter the number of rows and columns of matrix B: 2**

**2**

**Enter the elements of the matrix:**

**5**

**6**

**7**

**8**

**Choose an operation:**

**1. Display matrices**

**2. Matrix Multiplication**

**3. Matrix Addition**

**4. Matrix Subtraction**

**5. Transpose of Matrix A**

**1**

**Matrix:**

**1 2**

**3 4**

**Matrix:**

**5 6**

**7 8**

**2**

**Matrix:**

**7 10**

**15 22**

**3**

**Matrix:**

**2 4**

**6 8**

**4**

**Matrix:**

**-4 -4**

**-4 -4**

**5**

**Matrix:**

**1 3**

**2 4**

**PS C:\Users\abc\c language>**