1.number to binary representation.

C program:

#include<stdio.h>

void printBinary(int number){

    int binary[100];

    int index;

    if(number==0){

        printf("0");

        return;

    }

    while(number>0){

        binary[index]=number%2;

        number=number/2;

        index++;

    }

    for(int i=index-1;i>=0;i--){

        printf("%d",binary[i]);

    }

}

int main(){

    int input;

    printf("enter the number:");

    scanf("%d",&input);

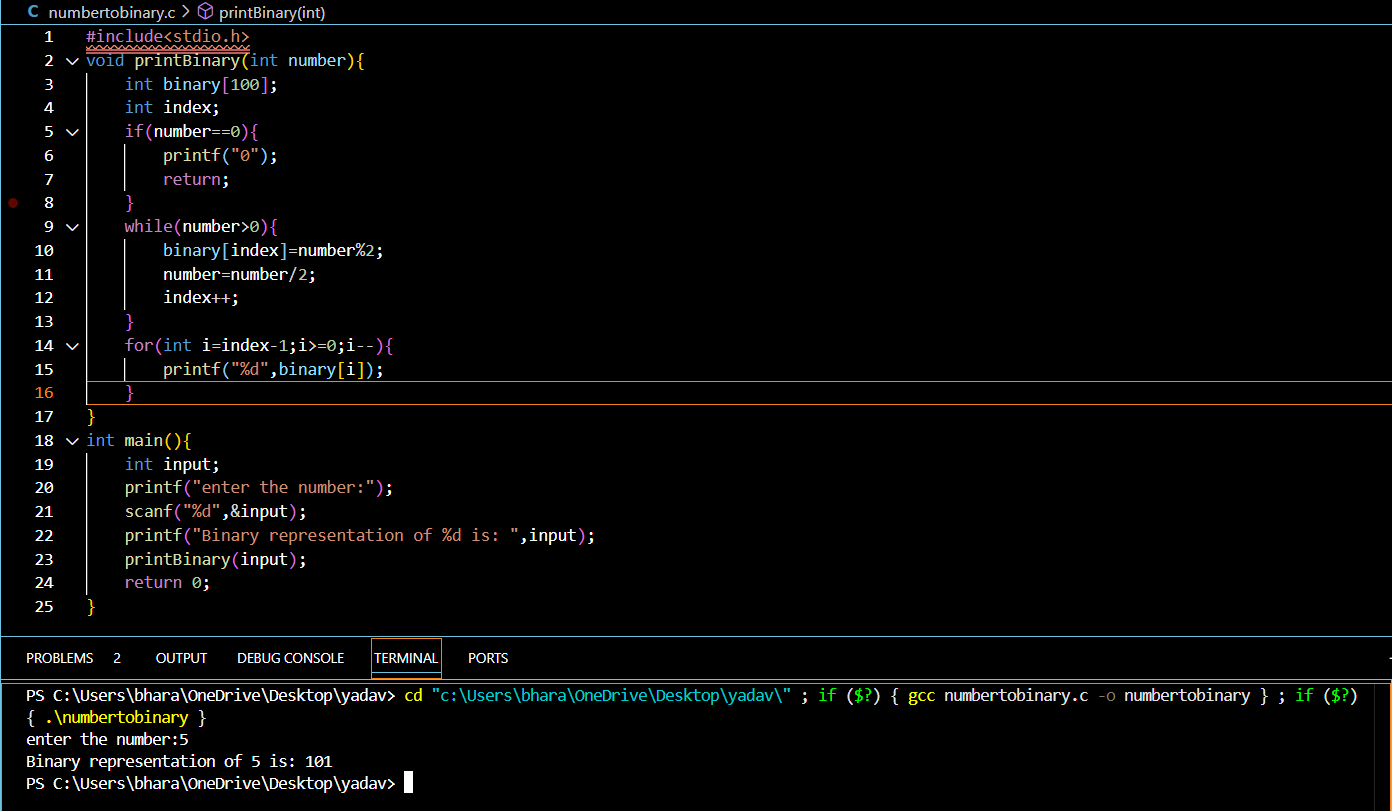
    printf("Binary representation of %d is: ",input);

    printBinary(input);

    return 0;

}

Output:



2.the count of number of vowels and consonents in a string.

C program:

#include<stdio.h>

#include<ctype.h>

void countVowelsAndConsonants(char \*str,int \*vowelCount,int \*consonantCount){

    \*vowelCount=0;

    \*consonantCount=0;

    while(\*str!='\0'){

        char ch=tolower(\*str);

        if(isalpha(ch)){

            if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'){

                (\*vowelCount)++;

            }else{

                (\*consonantCount)++;

            }

        }

        str++;

    }

}

int main(){

    char str[100];

    int vowels=0,consonants=0;

    printf("Enter the string:");

    fgets(str,sizeof(str),stdin);

    countVowelsAndConsonants(str,&vowels,&consonants);

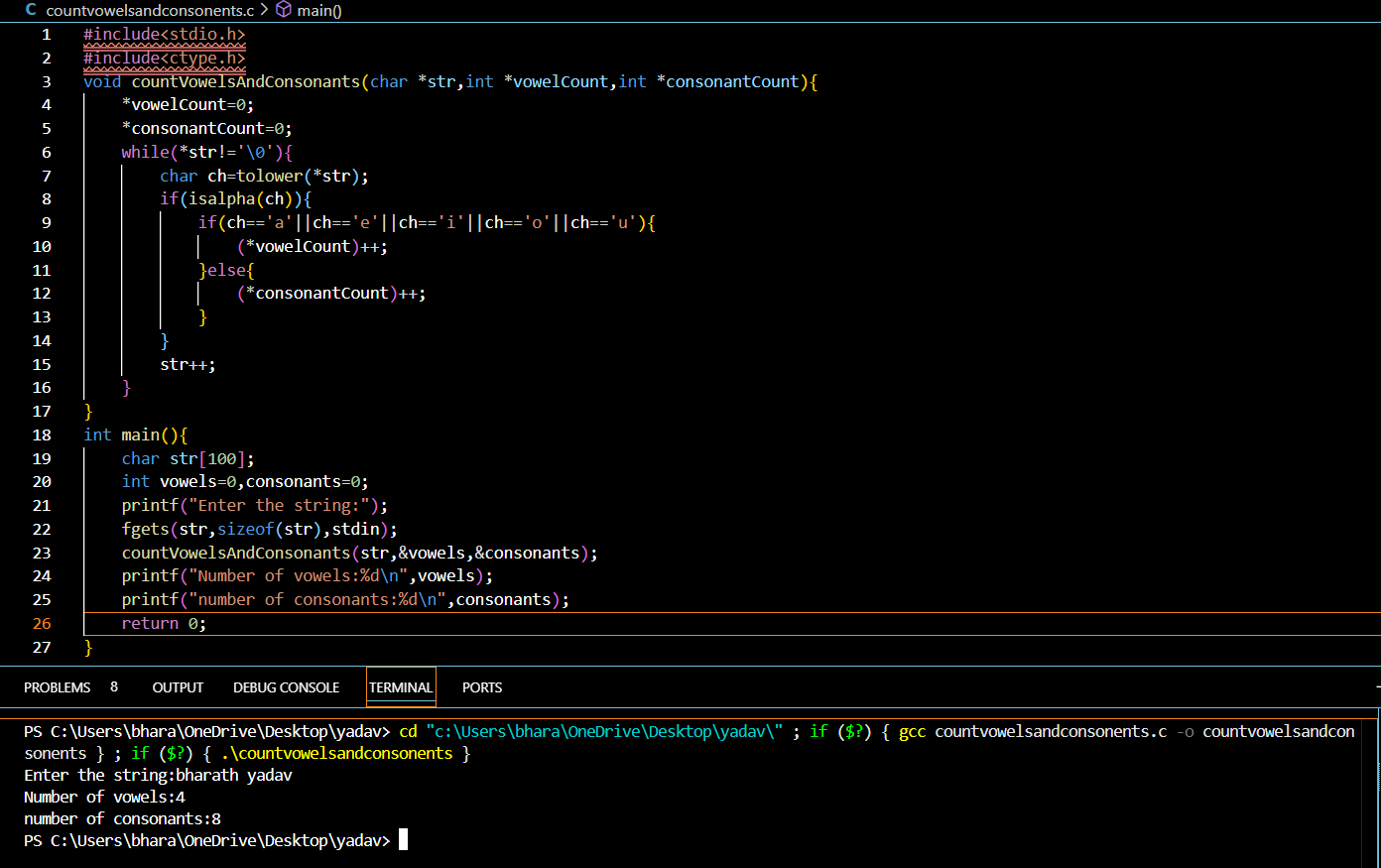
    printf("Number of vowels:%d\n",vowels);

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

    return 0;

}

Output:



3.caluclator.

#include<stdio.h>

void addition();

void subtraction();

void multiplication();

void divison();

int main(){

    int choice;

    do{

    printf("\nthe menu:\n");

    printf("1.addition\n");

    printf("2.subtraction\n");

    printf("3.mutiplication\n");

    printf("4.divison\n");

    printf("5.exit\n");

    printf("select the choice from the menu:");

    scanf("%d",&choice);

    switch(choice){

        case 1:

        addition();

        break;

        case 2:

        subtraction();

        break;

        case 3:

        multiplication();

        break;

        case 4:

        divison();

        break;

        case 5:

        printf("do not exist");

        break;

        default:

        printf("the invalid choice");

    }

    }while(choice !=5);

    return 0;

}

void addition(){

    int num1,num2;

    printf("enter the two numbers:");

    scanf("%d %d",&num1,&num2);

    int sum=num1+num2;

    printf("result:%d\n",sum);

}

void subtraction(){

    int num1,num2;

    printf("enter the two numbers:");

    scanf("%d %d",&num1,&num2);

    int subtraction=num1-num2;

    printf("result:%d\n",subtraction);

}

void multiplication(){

    int num1,num2;

    printf("enter the two numbers:");

    scanf("%d %d",&num1,&num2);

    int multiplication=num1\*num2;

    printf("result:%d\n",multiplication);

}

void divison(){

    int num1,num2;

    printf("enter the two numbers:");

    scanf("%d %d",&num1,&num2);

    if(num2==0){

        printf("divison cnnot possible");

    }else{

        int divison=num1/num2;

        printf("result:%d\n",divison);

    }

}

A screenshot of a computer program

Description automatically generated

4.diamond pattern.

C program:

#include<stdio.h>

int main(){

    int n,rows,cols,s;

    printf("enter the value of n:");

    scanf("%d",&n);

    for(rows=1;rows<=n;rows++){

        for(s=1;s<=n-rows;s++){

            printf(" ");

        }

        for(cols=1;cols<=(2\*rows-1);cols++){

            printf("\*");

        }

        printf("\n");

    }

    for(rows=n-1;rows>=1;rows--){

        for(s=1;s<=n-rows;s++){

            printf(" ");

        }

        for(cols=1;cols<=(2\*rows-1);cols++){

            printf("\*");

        }

        printf("\n");

    }

    return 0;

}



5.string length using pointers.

C program:

#include <stdio.h>

int main(){

    char str[100];

    char \*ptr;

    int length=0;

    printf("Enter a string: ");

    fgets(str,sizeof(str),stdin);

    ptr=str;

    while(\*ptr!='\0') {

        length++;

        ptr++;

    }

    printf("The length of the string %s is: %d\n",str,length);

    return 0;

}

A screenshot of a computer

Description automatically generated

6.GCD of two numbers using recursion.

C program:

#include <stdio.h>

int gcd(int a, int b) {

if (b==0){

        return a;

}else{

    return gcd(b,a%b);

}

}

int main() {

    int a,b;

    printf("Enter two numbers: ");

    scanf("%d %d",&a,&b);

    printf("GCD of %d and %d is: %d\n",a,b,gcd(a,b));

    return 0;

}

Output:

A screenshot of a computer

Description automatically generated