**Program 1: Write a C Program to check if a Given String is Palindrome or Not String: ABCDCBA.**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

  char s[200];

  int l,i,c=0;

  clrscr();

  printf("Enter your String:");

  scanf("%s",&s);

  l=strlen(s);

  for(i=0;i<l/2;i++)

  {

    if(s[i]==s[l-i-1])

    {

      c++;

    }

  }

    if(c==i)

    {

      printf("\n %s String is palindrome",s);

    }

    else

    {

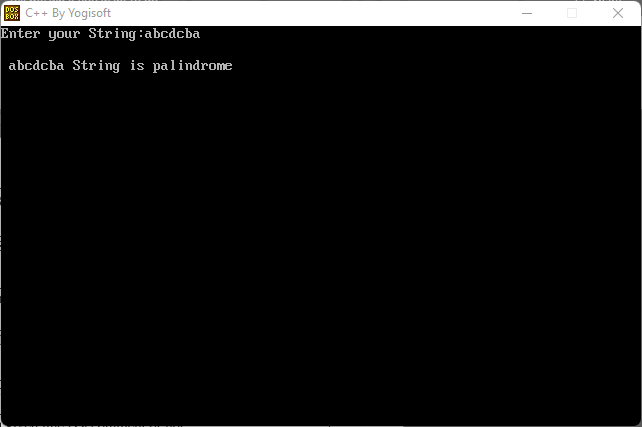
      printf("\n %s String is not palindrome",s);

    }

  getch();

}

**Output:**



**Program 2: C program to print Fibonacci series using recursion.**

#include<stdio.h>

#include<conio.h>

int Fibonacci(int);

void main(){

int n,i=0,c;

clrscr();

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

scanf("%d",&n);

printf("Fibonacci series :\n");

for(c=1;c<=n;c++){

printf("%d\n",Fibonacci(i));

i++;

}

getch();

}

int Fibonacci(int n){

if(n==0)

return 0;

else if(n==1)

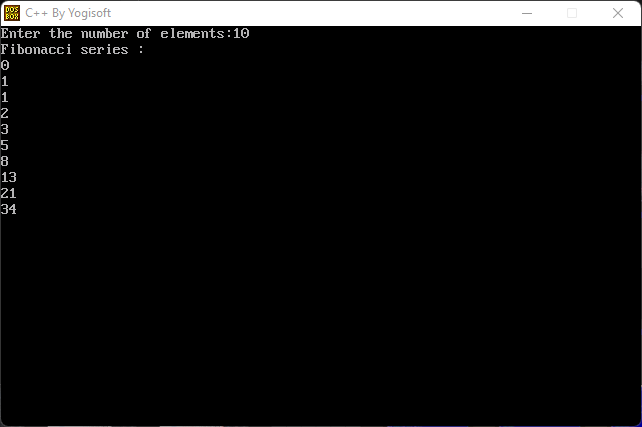
return 1;

else

return (Fibonacci(n-1)+Fibonacci(n-2));

}

**Output:**



**Program 3: Write a** [**Program to convert string from lower case to upper case**](https://beginnersbook.com/2015/02/c-program-to-convert-lowercase-string-to-uppercase-string/)**.**

#include<stdio.h>

#include<conio.h>

void main(){

 char str[50];

 int i;

 clrscr();

 printf("Enter your String:");

 gets(str);

 for(i=0;str[i]!='\0';i++){

  if(str[i]>='a' && str[i]<='z'){

   str[i]=str[i] - 32;

  }

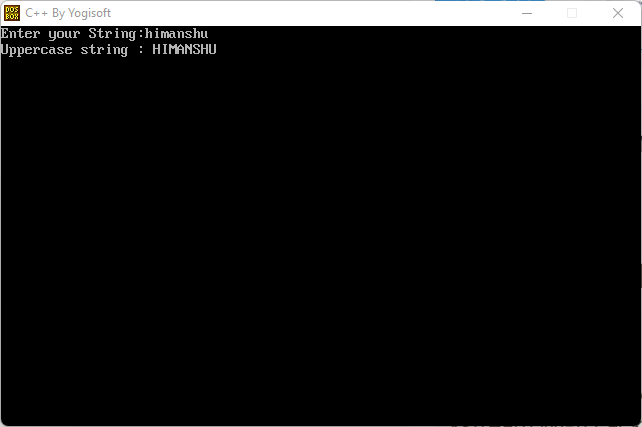
 }

 printf("Uppercase string : %s",str);

 getch();

}

**Output:**



**Program 4: Write a C Program to Scan and Count the number of characters, words, and lines in a file.**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

void main(){

 FILE \* file;

 char path[200];

 char ch;

 int characters,words,lines;

 clrscr();

 printf("Enter source file path:");

 scanf("%s",path);

 file=fopen(path,"r");

 if(file==NULL){

  printf("\n Unable to open the file. \n");

  printf("Please check if file exists and you have read privilege.\n");

  exit(EXIT\_FAILURE);

 }

 characters=words=lines=0;

 while((ch=fgetc(file))!=EOF){

  characters++;

  if(ch=='\n' || ch=='\0')

   lines++;

  if(ch==' ' || ch=='\t' || ch=='\n' || ch=='\0')

   words++;

 }

 if(characters > 0){

  words++;

  lines++;

 }

 printf("\n");

 printf("Total Characters = %d\n",characters);

 printf("Total Words=%d\n",words);

 printf("Total lines=%d\n",lines);

 fclose(file);

 getch();

}

**Output:**



**Program 5: Write a C program to identify whether a given line is a comment or not.**

#include<stdio.h>

#include<conio.h>

void main(){

 char com[100];

 int i=2,a=0;

 clrscr();

 printf("\n Enter Comment:");

 gets(com);

 if(com[0]=='/'){

  if(com[1]=='/'){

   printf("\n The give string is a comment");

  }else if(com[1]=='\*'){

   for(i=2;i<=100;i++){

    if(com[i]=='\*' && com[i+1]=='/'){

     printf("\n The give string is a Comment");

     a=1;

     break;

    }else{

     continue;

    }

    if(a==0)

     printf("\n it is not a comment");

   }

  }else{

   printf("\n It is not a comment");

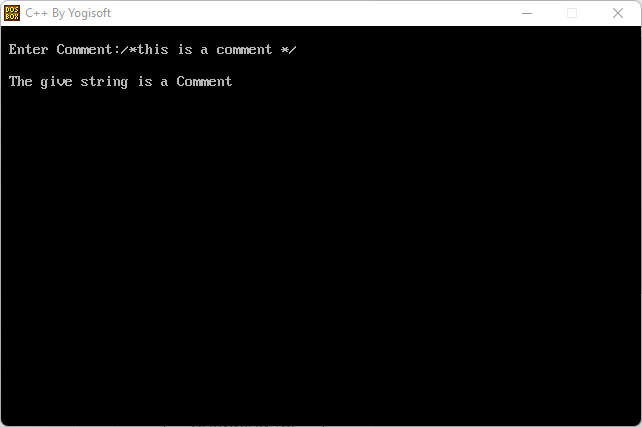
  }

 }

 getch();

}

**Output:**



**Program 6: Write a C program to detect tokens.**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<ctype.h>

int isKeyword(char buffer[]){

 char keywords[32][10] = {"auto","break","case","char","const","continue","default",

 "do","double","else","enum","extern","float","for","goto",

 "if","int","long","register","return","short","signed",

 "sizeof","static","struct","switch","typedef","union",

 "unsigned","void","volatile","while"};

 int i, flag = 0;

 for(i = 0; i < 32; ++i){

 if(strcmp(keywords[i], buffer) == 0){

 flag = 1;

 break;

 }

 }

 return flag;

}

void main(){

 char ch, buffer[15], operators[] = "+-\*/%=";

 FILE \*fp;

 int i,j=0;

 clrscr();

 fp = fopen("test.txt","r");

 if(fp == NULL){

 printf("error while opening the file\n");

 exit(0);

 }

 while((ch = fgetc(fp)) != EOF){

   for(i = 0; i < 6; ++i){

   if(ch == operators[i])

   printf("%c is operator\n", ch);

   }

   if(isalnum(ch)){

   buffer[j++] = ch;

   }

   else if((ch == ' ' || ch == '\n') && (j != 0)){

   buffer[j] = '\0';

   j = 0;

   if(isKeyword(buffer) == 1)

   printf("%s is keyword\n", buffer);

   else

   printf("%s is indentifier\n", buffer);

   }

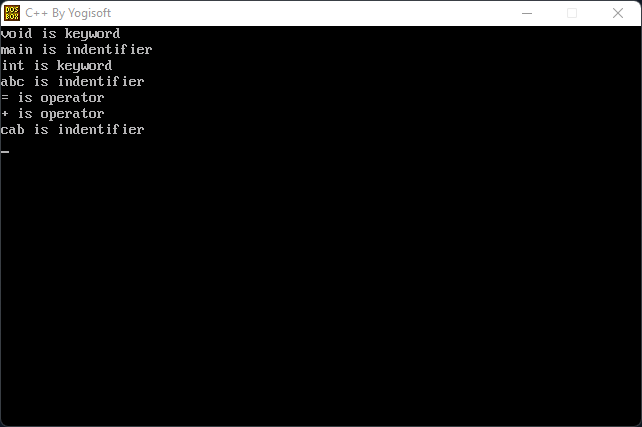
 }

 fclose(fp);

 getch();

}

**Output:**



**Program 7: Write a C program to test whether a given identifier is valid or not.**

#include<stdio.h>

#include<conio.h>

#include<ctype.h>

void main(){

 char a[50];

 int flag,i=1;

 clrscr();

 printf("\n Enter an identifier:");

 gets(a);

 if(isalpha(a[0]))

 flag=1;

 else

 printf("\n Not a vaild identifier");

 while(a[i]!='\0'){

  if(!isdigit(a[1]) && !isalpha(a[i])){

   flag=0;

   break;

  }

  i++;

 }

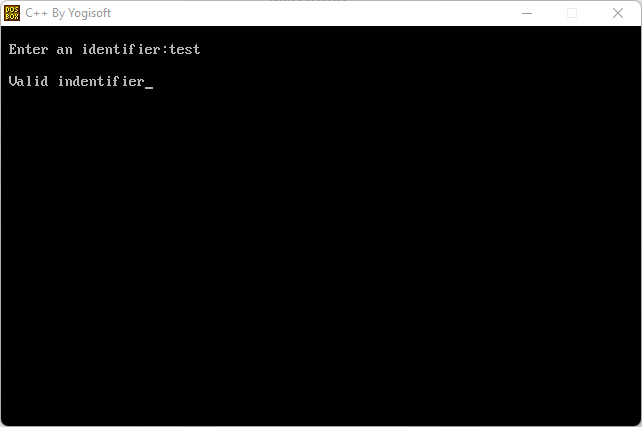
 if(flag==1)

 printf("\n Valid indentifier");

 getch();

}

**Output:**



**Program 8: Write a lex program to find out total number of vowels, and consonants from the given input string.**

%{

int ac=0;

int bc=0;

%}

%%

[aeiouAEIOU] {ac++;}

[a-zA-z] {bc++;}

%%

int yywrap() {}

int main()

{

printf("Enter the string \n");

yylex();

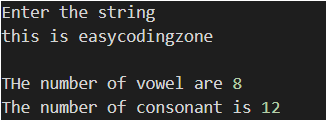
printf("THe number of vowel are %d\n",ac);

printf("The number of consonant is %d",bc);

return 0;

}

**Output:**



**Program 9: Write a C program to recognize strings under 'a\*', 'a\*b+', 'abb'.**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

void main()

{

 char s[20],c;

 int state=0,i=0;

 clrscr();

 printf("\n Enter a string:");

 gets(s);

 while(s[i]!='\0'){

  switch(state)

  {

   case 0: c=s[i++];

    if(c=='a')

   state=1;

   else if(c=='b')

  state=2;

   else

  state=6;

   break;

  case 1: c=s[i++];

   if(c=='a')

  state=3;

   else if(c=='b')

  state=4;

   else

  state=6;

   break;

  case 2: c=s[i++];

   if(c=='a')

  state=6;

   else if(c=='b')

  state=2;

   else

  state=6;

   break;

  case 3: c=s[i++];

   if(c=='a')

  state=3;

   else if(c=='b')

  state=2;

   else

  state=6;

   break;

  case 4: c=s[i++];

   if(c=='a')

  state=6;

   else if(c=='b')

  state=5;

   else

  state=6;

   break;

  case 5: c=s[i++];

   if(c=='a')

  state=6;

   else if(c=='b')

  state=2;

   else

  state=6;

  break;

  case 6: printf("\n %s is not recognised.",s);

  exit(0);

  }

 }

 if(state==1)

  printf("\n %s is accepted under rule 'a'",s);

 else if((state==2)||(state==4))

  printf("\n %s is accepted under rule 'a\*b+'",s);

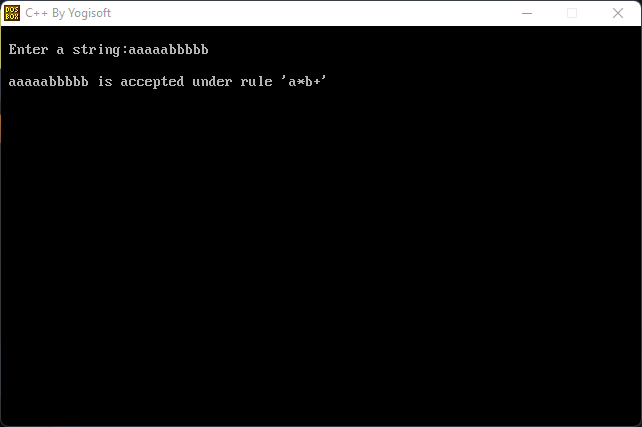
 else if(state==5)

  printf("\n %s is accepted under rule 'abb'",s);

 getch();

}

**Output:**



**Program 10: Write a C program to simulate lexical analyzer for validating operators.**

#include<stdio.h>

#include<conio.h>

void main()

{

char s[5];

clrscr();

printf("\n Enter any operator:");

gets(s);

switch(s[0])

{

case'>': if(s[1]=='=')

printf("\n Greater than or equal");

else

printf("\n Greater than");

break;

case'<': if(s[1]=='=')

printf("\n Less than or equal");

else

printf("\nLess than");

break;

case'=': if(s[1]=='=')

printf("\nEqual to");

else

printf("\nAssignment");

break;

case'!': if(s[1]=='=')

printf("\nNot Equal");

else

printf("\n Bit Not");

break;

case'&': if(s[1]=='&')

printf("\nLogical AND");

else

printf("\n Bitwise AND");

break;

case'|': if(s[1]=='|')

printf("\nLogical OR");

else

printf("\nBitwise OR");

break;

case'+': printf("\n Addition");

break;

case'-': printf("\nSubstraction");

break;

case'\*': printf("\nMultiplication");

break;

case'/': printf("\nDivision");

break;

case'%': printf("Modulus");

break;

default: printf("\n Not a operator");

}

getch();

}

**Output:**

