EX1:LEXICAL ANALYSIS

CODE:

#include<stdio.h>

#include<string.h>

#include<stdlib.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;

}

int main()

{

char ch, buffer[15],b[30], logical\_op[] = "><",math\_op[]="+-\*/=",numer[]=".0123456789",other[]=",;\(){}[]'':";

char kv[32][10],cv[32][10],iv[32][10];

int othc=0,aaa=0,j=0,kvc=0;

char ov[10],mv[10],lv[10];

int ovc=0,mvc=0,ivc=0,lvc=0,cvc=0;

FILE \*fp;

int mark[1000]={0};

fp=fopen("hello","r");

while((ch=fgetc(fp))!=EOF)

{

for(int i=0;i<12;i++)

{

if(ch==other[i])

{

int aa=ch;

if(mark[aa]!=1)

{

ov[ovc++]=ch;

mark[aa]=1;

}

}

}

for(int i=0;i<5;i++)

{

if(ch==math\_op[i])

{

int aa=ch;

if(mark[aa]!=1)

{

mv[mvc++]=ch;

mark[aa]=1;

}

}

}

for(int i=0;i<2;i++)

{

if(ch==logical\_op[i])

{

int aa=ch;

if(mark[aa]!=1)

{

lv[lvc++]=ch;

mark[aa]=1;

}

}

}

if(ch=='0' || ch=='1' || ch=='2' || ch=='3' || ch=='4' || ch=='5' || ch=='6' || ch=='7' || ch=='8' || ch=='9' || ch=='.' ||ch == ' ' || ch == '\n' || ch == ';'){

if(ch=='0' || ch=='1' || ch=='2' || ch=='3' || ch=='4' || ch=='5' || ch=='6' || ch=='7' || ch=='8' || ch=='9' || ch=='.')b[aaa++]=ch;

if((ch == ' ' || ch == '\n' || ch == ';') && (aaa != 0)){

b[aaa] = '\0';

aaa = 0;

char arr[30];

strcpy(arr,b);

strcpy(cv[cvc++],arr);

}

}

if(isalpha(ch))

{

buffer[j++]=ch;

}

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

{

buffer[j]='\0';

j=0;

if(isKeyword(buffer)==1)

{

strcpy(kv[kvc++],buffer);

}

else

{

if(mark[buffer[0]-'a']!=1 &&(strcmp(buffer,"printf")!=0 && strcmp(buffer,"includestdioh")!=0)){

if(strcmp(buffer,"main")!=0)

{

strcpy(iv[ivc++],buffer);

mark[buffer[0]-'a']=1;

}

}

}

}

}

printf("No of keywords are %d\n",kvc);

printf("keywords :");

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

{

printf("%s ",kv[i]);

}

printf("\n");

printf("No of identifiers are %d\n",ivc);

printf("Identifers: ");

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

{

printf("%s ",iv[i]);

}

printf("\n");

printf("No of Arthematic operators are %d\n",mvc);

printf("Arthematic Operators : ");

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

{

printf("%c ",mv[i]);

}

printf("\n");

printf("No of Relational operators are %d\n",lvc);

printf("Relational operator : ");

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

{

printf("%c ",lv[i]);

}

printf("\n");

printf("No of constants are %d\n",cvc-1);

for(int i=1;i<cvc;i++)

{

printf("%s ",cv[i]);

}

printf("\n");

printf("No of other symbols is %d\n",ovc);

printf("Other variables : ");

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

{

printf("%c ",ov[i]);

}

}

INPUT C CODE:

#include<stdio.h>

void main()

{

int a ,b , c , d;

a = 10;

b = 22;

c = 19;

d = c;

if (a > b )

{

b = b + 2.22;

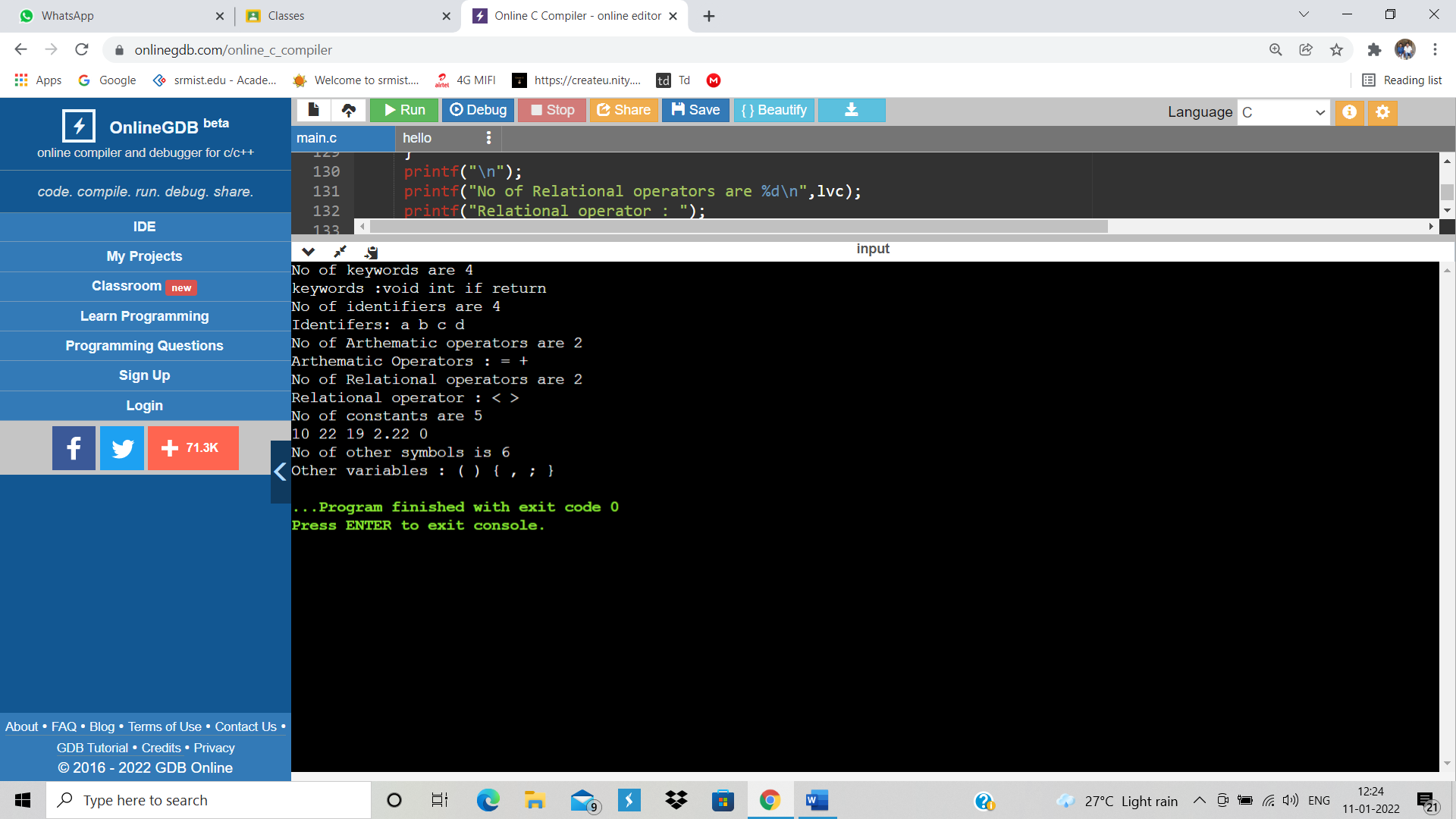
}

printf ("%d ",b);

return 0;

}

OUTPUT:



M.NISHANTH(RA1911003010772)