

Source File

hashcover <mathematical.header>

hashcover <exchangeInfo.header >

number main\_function LP RP

LB

!! Here identifier declaration

number num1 cm num2 sm

string str1 cm str2 sm

%&

mathematical

calculation is

given below

&%

IC LP str1 GT str2 RP

LB

num1 <- num1 ++ num2 sm

RB

OW

LB

num2 <- num2 \*\* num1 sm

RB

give 0 sm

RB

Flex(.I) File

%{

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

char headerStorage[10][50];

int headerCounter=0,a=0;

int headerOperation(char \*point)

{

for(a=0; a<headerCounter ;a++)

{

if(strcmp(point,headerStorage[a])==0)

```

        return 0;
    }
    strcpy(headerStorage[a],point);
    headerCounter++;
}

char keyWordStorage[50][50];
int keyWordCounter=0;
int keyWordOperation(char *point)
{
    for(a=0; a< keyWordCounter ;a++)
    {
        if(strcmp(point,keyWordStorage[a])==0)
            return 0;
    }
    strcpy(keyWordStorage[a],point);
    keyWordCounter++;
}

char operatorStorage[50][50];
int operatorCounter=0;
int operatorOperation(char *point)
{
    for(a=0; a< operatorCounter ;a++)
    {
        if(strcmp(point,operatorStorage[a])==0)
            return 0;
    }
    strcpy(operatorStorage[a],point);
    operatorCounter++;
}

char identifierStorage[50][50];
int identifierCounter=0;
int identifierOperation(char *point)
{
    for(a=0; a< identifierCounter ;a++)
    {
        if(strcmp(point,identifierStorage[a])==0)
            return 0;
    }
}

```

```

        }

        strcpy(identifierStorage[a],point);

        identifierCounter++;

    }

char punctuationSymbolStorage[50][50];

int punctuationSymbolCounter=0;

int punctuationSymbolOperation(char *point)
{
    for(a=0; a< punctuationSymbolCounter ;a++)
    {
        if(strcmp(point,punctuationSymbolStorage[a])==0)
            return 0;

    }

    strcpy(punctuationSymbolStorage[a],point);

    punctuationSymbolCounter++;

}

char singleLineCommentStorage[50][50];

int singleLineCommentCounter=0;

int singleLineCommentOperation(char *point)
{
    for(a=0; a< singleLineCommentCounter ;a++)
    {
        if(strcmp(point,singleLineCommentStorage[a])==0)
            return 0;

    }

    strcpy(singleLineCommentStorage[a],point);

    singleLineCommentCounter++;

}

char multipleLineCommentStorage[50][50];

int multipleLineCommentCounter=0;

int multipleLineCommentOperation(char *point)
{
    for(a=0; a< multipleLineCommentCounter ;a++)
    {
        if(strcmp(point,multipleLineCommentStorage[a])==0)
            return 0;

    }

```

```

        strcpy(multipleLineCommentStorage[a],point);
        multipleLineCommentCounter++;
    }
int outputFunction()
{
    for(a=0;a<keyWordCounter;a++)
        printf("%s\n",headerStorage[a]);
    printf("Total header = %d\n",headerCounter);
    for(a=0;a<keyWordCounter;a++)
        printf("%s\n",keyWordStorage[a]);
    printf("Total keyWord = %d\n\n",keyWordCounter);

    for(a=0;a<operatorCounter;a++)
        printf("%s\n",operatorStorage[a]);
    printf("Total operator = %d\n\n",operatorCounter);

    for(a=0;a<identifierCounter;a++)
        printf("%s\n",identifierStorage[a]);
    printf("Total identifier = %d\n\n",identifierCounter);

    for(a=0;a<punctuationSymbolCounter;a++)
        printf("%s\n",punctuationSymbolStorage[a]);
    printf("Total punctuationSymbol = %d\n\n",punctuationSymbolCounter);

    for(a=0;a<singleLineCommentCounter;a++)
        printf("%s\n",singleLineCommentStorage[a]);
    printf("Total singleLineComment = %d\n\n",singleLineCommentCounter);

    for(a=0;a<multipleLineCommentCounter;a++)
        printf("%s\n",multipleLineCommentStorage[a]);
    printf("Total multipleLineComment = %d\n\n",multipleLineCommentCounter);
    return 0;
}

%}

header                [^\n]+[.header][ ]*>

keyWord                number|string|IC|OW|give

```

```

oprator                "GT"|"++"|"--"|"<-"|"**"|"md"|"//"
identifier              [a-zA-Z_]+[0-9]*
punctuationSymbol      LB|RB|LP|RP|cm|sm
singleLineComment      [ ]*![!][a-zA-Z0-9!@#$$%^&*(){}_+.,:\|?><]*
multipleLineComment    [ ]*%[%][&][a-zA-Z0-9!@#$$*(){}_+.,:\|?><\n\t ]*[%][&]
%%

{header}                {headerOperation(yytext);}
{singleLineComment}     {singleLineCommentOperation(yytext);}
{multipleLineComment}   {multipleLineCommentOperation(yytext);}
{operator}              {operatorOperation(yytext);}
{punctuationSymbol}     {punctuationSymbolOperation(yytext);}
{keyWord}                {keyWordOperation(yytext);}
{identifier}            {identifierOperation(yytext);}

%%

int yywrap(){
    return 1;
}

int main(){
    freopen("input2.txt","r",stdin);
    yylex();
    freopen("out2.txt","w",stdout);
    outputFunction();
    return 0;
}

```

## Output File

hashcover <mathematical.header>

hashcover <exchangeInfo.header>

Total header = 2

number

string

IC

OW

give

Total keyWord = 5

GT

<-

++

\*\*

Total operator = 4

num1

num2

str1

str2

Total identifier = 5

LP

RP

LB

cm

sm

RB

Total punctuationSymbol = 6

!! Here identifier declaration

Total singleLineComment = 1

%&

mathematical

calculation is

given below

&%

Total multipleLineComment = 1