Question:
Suppose, we have a C source program scanned and filtered as it was done in Session 1. We now takethat modified file as input, and separate the lexemes first. We further recognize and mark thelexemes as different types of tokens like keywords, identifiers, operators, separators, parenthesis,numbers, etc.

Code:

```
#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;
}

int main(){
        char ch, buffer[15], operators[] = "+-*/%=";
        char separators[] = ",";";
   int isOperator= 0,isSeperator=0,isParanthesis=0;
        FILE *fp , *fp2;
        int i,j=0;
   int seperate = 0;
        fp = fopen("input.c","r");
        fp2 = fopen("output.txt", "w");
```

```
if(fp == NULL){
        printf("error while opening the file\n");
        exit(0);
}

printf("\nStep 1:\n");

while((ch = fgetc(fp)) != EOF){
        for(i = 0; i < 6; ++i){
                if(ch == operators[i]){
   isOperator = 1;

   seperate = 1;
                }

        }

        for(i = 0; i < 4; ++i){
                if(ch == separators[i]){
   isSeperator = 1;

   seperate = 1;
                }

        }

        if(ch == '(' || ch == ')'){
isParanthesis = 1;

seperate = 1;
        }



        if(isalnum(ch)){
                buffer[j++] = ch;
        }
        else if((ch == ' ' || ch == '\n' || seperate == 1) && (j != 0)){
                        buffer[j] = '\0';
                        j = 0;
                        seperate = 0;
```

```
                              if(isKeyword(buffer) == 1){
         fputs("[kw ", fp2);
         fputs(buffer, fp2);
         fputs("]", fp2);
         printf("%s ", buffer);
                              }
                              else{
         fputs("[id ", fp2);
        fputs(buffer, fp2);
        fputs("]", fp2);
        printf("%s ", buffer);
                              }

            }

if(isSeperator == 1 ){
   fputs("[sep ", fp2);
   fputc(ch, fp2);
   fputs("]", fp2);
   printf("%c ", ch);
   isSeperator = 0;
}
else if(isOperator == 1){
   fputs("[op ", fp2);
   fputc(ch, fp2);
   fputs("]", fp2);
   printf("%c ", ch);
   isOperator = 0;
}
else if(isParanthesis == 1){
   fputs("[par ", fp2);
   fputc(ch, fp2);
   fputs("]", fp2);
   printf("%c ", ch);
   isParanthesis = 0;
}



   }

   fclose(fp);
```

```
fp = fopen("input.c","r");

if(fp == NULL){
        printf("error while opening the file\n");
        exit(0);
}

printf("\nStep 2:\n");

while((ch = fgetc(fp)) != EOF){
        for(i = 0; i < 6; ++i){
                if(ch == operators[i]){
   isOperator = 1;

   seperate = 1;
                }

        }

        for(i = 0; i < 4; ++i){
                if(ch == separators[i]){
   isSeperator = 1;

   seperate = 1;
                }

        }

        if(ch == '(' || ch == ')'){
isParanthesis = 1;

seperate = 1;
        }




        if(isalnum(ch)){
                buffer[j++] = ch;
        }
        else if((ch == ' ' || seperate == 1) && (j != 0)){
                        buffer[j] = '\0';
                        j = 0;
```

```
                                                seperate = 0;


                                                if(isKeyword(buffer) == 1){

                    printf("[kw %s] ", buffer);
                                                }
                                                else{

                   printf("[id %s] ", buffer);
                                                }

                        }

        if(isSeperator == 1 ){

            printf("[sp %c] ", ch);
            isSeperator = 0;
        }
        else if(isOperator == 1){

            printf("[op %c] ", ch);
            isOperator = 0;
        }
        else if(isParanthesis == 1){

            printf("[pr %c] ", ch);
            isParanthesis = 0;
        }



            }



            fclose(fp);

            return 0;
}
```

Input:

```
char c; int x1, x_2; float y1, y2; x1=5; x_2=10; y1=2.5+x1*45; y2=100.o5-x_2/3; if(y1<=y2)c='y';
else c='n';
```

Output:

```
[kw char][id c][sep ;][kw int][id x1][sep ,][id x2][sep ;][kw float][id y1][sep ,][id y2][sep ;][id x1][op
=][id 5][sep ;][id x2][op =][id 10][sep ;][id y1][op =][id 25][op +][id x1][op *][id 45][sep ;][id y2][op
=][id 100o5][op -][id x2][op /][id 3][sep ;][kw if][par (][id y1][op =][id y2][par )][id c][op =][sep '][id
y][sep '][sep ;][kw else][id c][op =][sep '][id n][sep '][sep ;]
```