**EXPERIMENT NO. 3**

**AIM:** Design and implementation of macroprocessor

**PROGRAM:**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

FILE \*ptr;

FILE \*preptr;

FILE \*fp;

int mdt\_count = 0;

typedef struct{

int mdtc;

char macro[50];

}mdt;

typedef struct{

char macro\_name[10];

int mdt\_line;

}mnt;

typedef struct{

char index[3];

char dummy\_arg[3];

}ala\_1;

typedef struct{

char index[3];

char arg[3];

}ala\_2;

mdt def\_table[10];

mnt name\_table;

ala\_1 arglist\_1[2];

ala\_2 arglist\_2[2];

void printMDT(){

int j;

printf("\nMDT:");

for(j=0;j<mdt\_count;j++){

printf("\n%d %s",def\_table[j].mdtc,def\_table[j].macro);

}

}

void printMNT(){

printf("\n\nMNT:");

printf("\n%s %d",name\_table.macro\_name,name\_table.mdt\_line);

}

void printALA(){

int j;

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

for(j=0;j<2;j++){

printf("\n%s %s",arglist\_1[j].index,arglist\_1[j].dummy\_arg);

}

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

for(j=0;j<2;j++){

printf("\n%s %s",arglist\_2[j].index,arglist\_2[j].arg);

}

printf("\n");

}

void prepareALA\_2(){

int j;

char buf[10];

for(j=0;j<2;j++){

sprintf(arglist\_2[j].index,"#%d",j);

fscanf(fp,"%s",buf);

strcpy(arglist\_2[j].arg,buf);

// printf("\n%s %s\n",arglist\_2[j].index,arglist\_2[j].arg);

}

}

char \*getArg(char \*str){

int j;

for(j=0;j<2;j++){

if(strcmp(arglist\_2[j].index,str) == 0){

return arglist\_2[j].arg;

}

}

}

void getMDT(){

int j,k;

printf("\n\n");

for(j=1;j<mdt\_count-1;j++){

char buf[3];

for(k=0;k<strlen(def\_table[j].macro)-2;k++){

printf("%c",def\_table[j].macro[k]);

}

sprintf(buf,"%c%c",def\_table[j].macro[k],def\_table[j].macro[k+1]);

strcpy(buf,getArg(buf));

printf("%s\n",buf);

}

}

char \*getIndex(char \*str){

int j;

for(j=0;j<2;j++){

if(strcmp(arglist\_1[j].dummy\_arg,str) == 0){

return arglist\_1[j].index;

}

}

}

void main(){

size\_t len = 0;

ptr = fopen("macro.txt","r");

preptr = fopen("macro.txt","r");

fp = fopen("macro.txt","r");

char buffer[10];

char ch;

while(fscanf(ptr,"%s",buffer) == 1){

// printf("%s\n",buffer);

if(strcmp(buffer,"MACRO")==0){

// printf("%s\n",buffer);

fscanf(preptr,"%s",buffer);

fscanf(fp,"%s",buffer);

// printf("%s\n",buffer);

fscanf(ptr,"%s",buffer);

strcpy(name\_table.macro\_name,buffer);

name\_table.mdt\_line = mdt\_count + 1;

fscanf(ptr,"%s",buffer);

strcpy(arglist\_1[0].index,"#0");

strcpy(arglist\_1[0].dummy\_arg,buffer);

fscanf(ptr,"%s",buffer);

strcpy(arglist\_1[1].index,"#1");

strcpy(arglist\_1[1].dummy\_arg,buffer);

ch = fgetc(preptr);

// printf("\*\*\*\*\*%d\*\*\*\*\*\*\n",(int)ch);

ch = fgetc(preptr);

while(1){

ch = fgetc(preptr);

if(ch == ' '){

fscanf(fp,"%s",buffer);

// printf("%s\n",buffer);

def\_table[mdt\_count].mdtc = mdt\_count+1;

if(buffer[0] == '&'){

char index[3];

strcpy(index,getIndex(buffer));

strcat(def\_table[mdt\_count].macro,index);

}

else{

strcat(def\_table[mdt\_count].macro,buffer);

}

strcat(def\_table[mdt\_count].macro," ");

}

if(ch == '\n'){

fscanf(fp,"%s",buffer);

// printf("%s\n",buffer);

def\_table[mdt\_count].mdtc = mdt\_count+1;

if(buffer[0] == '&'){

char index[3];

strcpy(index,getIndex(buffer));

strcat(def\_table[mdt\_count].macro,index);

}

else{

strcat(def\_table[mdt\_count].macro,buffer);

}

mdt\_count++;

}

if(strcmp(buffer,"MEND") == 0){

break;

}

}

ptr = preptr;

fp = preptr;

}

else{

printf("%s ",buffer);

// fscanf(fp,"%s",buffer);

// ch = fgetc(ptr);

while(1){

ch = fgetc(ptr);

if(ch == EOF){

break;

}

if(ch == ' ' || ch == '\n'){

fscanf(fp,"%s",buffer);

if(strcmp(buffer,name\_table.macro\_name) == 0){

prepareALA\_2();

getMDT();

}

else{

printf("%c",ch);

printf("%s",buffer);

}

}

}

printf("\n");

}

}

printMDT();

printMNT();

printALA();

fclose(ptr);

// fclose(preptr);

// fclose(fp);

// free(buf);

}

**OUTPUT:**

PROG START

USING \* 15

ST 1 D1

L 2 D2

D1 DC F 2

D2 DC F 3

END

MDT:

1 XYZ #0 #1

2 ST 1 #0

3 L 2 #1

4 MEND

MNT:

XYZ 1

ALA 1:

#0 &A

#1 &B

ALA 2:

#0 D1

#1 D2