Yasmin Ansari
TE – CMPN (A)
Roll No: 71

Experiment No.

Aim: Implementation of Macro processor

Pass 1

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
void main() {
FILE *f1,*f2,*f3;
char mne[20],opnd[20],la[20],ch;
clrscr();
f1=fopen("macro_in.txt","r");
f2=fopen("mnt.txt","w+");
f3=fopen("mdt.txt","w+");
fscanf(f1,"%s %s %s",la,mne,opnd);
while(strcmp(mne,"MEND")!=0)  {
if(strcmp(mne,"MACRO")==0) {
fprintf(f2,"%s\n",la);
fprintf(f3,"%s\t%s\n",la,opnd);  }
else
fprintf(f3,"%s\t%s\n",mne,opnd);
fscanf(f1,"%s %s %s",la,mne,opnd); }
fprintf(f3,"%s",mne);
fclose(f1);
fclose(f2);
fclose(f3);
printf("Input File\n");
f1=fopen("macro_in.txt","r");
while((ch=fgetc(f1))!=EOF)
printf("%c",ch);
fclose(f1);
printf("\nMacro Name Table\n");
f2=fopen("mnt.txt","r");
while((ch=fgetc(f2))!=EOF)
```

```
printf("%c",ch);
fclose(f2);
printf("\nMacro Definition Table\n");
f3=fopen("mdt.txt","r");
while((ch=fgetc(f3))!=EOF)
printf("%c",ch);
fclose(f3);
getch();
}
```

Output

Input File

```
DISP MACRO      &A,&B
-       L       &A
-       ST      &B
-       MEND    -
SAMPLE  START   1000
-       DISP N1,N2
N1      DC      F'1'
N2      DC      F'1'
-       END     -
```

Macro Name Table

```
DISP
```

Macro Definition Table

```
DISP    &A,&B
L       &A
ST      &B
MEND
```

Pass 2

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
void main() {
FILE *f1,*f2,*f3,*f4,*f5;
int i,len;
```

```
char mne[20],opnd[20],la[20],name[20],mne1[20],opnd1[20],arg[20],ch;
clrscr();
f1=fopen("macro_in.txt","r");
f2=fopen("mnt.txt","r");
f3=fopen("mdt.txt","r");
f4=fopen("ala.txt","w+");
f5=fopen("op.txt","w");
fscanf(f1,"%s %s %s",la,mne,opnd);
while(strcmp(mne,"END")!=0) {
if(strcmp(mne,"MACRO")==0) {
fscanf(f1,"%s %s %s",la,mne,opnd);
while(strcmp(mne,"MEND")!=0)
fscanf(f1,"%s %s %s",la,mne,opnd); }
else {
fscanf(f2,"%s",name);
if(strcmp(mne,name)==0) {
len=strlen(opnd);
for(i=0;i<len;i++) {
if(opnd[i]!=',')
fprintf(f4,"%c",opnd[i]);
else
fprintf(f4,"\n"); }
fseek(f2,SEEK_SET,0);
fseek(f4,SEEK_SET,0);
fscanf(f3,"%s%s",mne1,opnd1);
fprintf(f5,".\t%s\t%s\n",mne1,opnd);
fscanf(f3,"%s%s",mne1,opnd1);
while(strcmp(mne1,"MEND")!=0) {
if((opnd1[0]=='&')) {
fscanf(f4,"%s",arg);
fprintf(f5,"-\t%s\t%s\n",mne1,arg); }
else
fprintf(f5,"-\t%s\t%s\n",mne1,opnd1);
fscanf(f3,"%s%s",mne1,opnd1); } }
else
```

```
fprintf(f5,"%s\t%s\t%s\n",la,mne,opnd); }
fscanf(f1,"%s %s %s",la,mne,opnd); }
fprintf(f5,"%s\t%s\t%s\n",la,mne,opnd);
fclose(f1);
fclose(f2);
fclose(f3);
fclose(f4);
fclose(f5);
f4=fopen("ala.txt","r");
printf("\nArgument List Array\n");
while((ch=fgetc(f4))!=EOF)
printf("%c",ch);
fclose(f4);
printf("\nExpanded Code\n");
f5=fopen("op.txt","r");
while((ch=fgetc(f5))!=EOF)
printf("%c",ch);
fclose(f5);
getch(); }
```

Output

```
Argument List Array
N1
N2
Expanded Code
SAMPLE  START   1000
.       DISP    N1,N2
-       L       N1
-       ST      N2
N1      DC      F'1'
N2      DC      F'1'
-       END     -
```