IMPLEMENT A TWO PASS MACRO PROCESSOR

AIM:

To implement two pass macro processor using in C language.

ALGORITHM:

1. Start the program execution.

2. Macro instructions are included in a separate file.

3. The instructions with ‘macro’,’mend’,’call’ on them should not be printed in the output.

4. Print all other instructions such as start,load,store,add,sub Etc with their values.

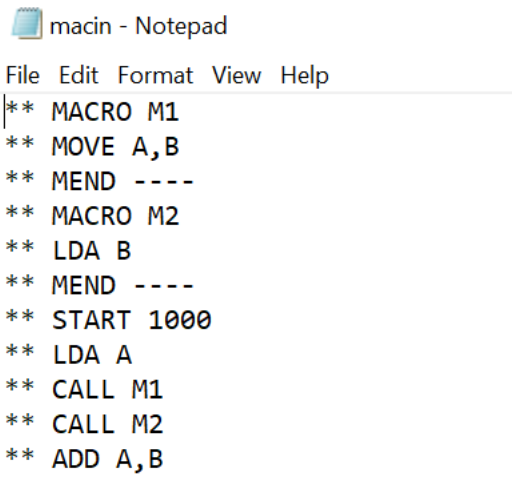
5. Stop the program execution.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
 char n1,n,c1,i;
 char fn[10][10],ilab[20],iopd[20],m[20][3],oper[20],opd[20];
 FILE *fp1,*fp2,*p[5];
 clrscr();
 n=0;
 fp1=fopen("macin.txt","r");
 while(!feof(fp1))
 {
  fscanf(fp1,"%s%s%s",ilab,iopd,oper);
  if(strcmp(iopd,"MACRO")==0)
  n++;
 }
 printf("No.of macros=%d\n",n);
 n1=n;
 printf("Enter the text filename \n");
 for(i=0;i<n;i++)
 {
  scanf("%s",fn[i]);
  p[i]=fopen(fn[i],"w");
 }
 n=0;
 rewind(fp1);
 while(!feof(fp1))
 {
  fscanf(fp1,"%s%s%s",ilab,iopd,oper);
  if(strcmp(iopd,"MACRO")==0)
  {
   strcpy(m[n],oper);
   fscanf(fp1,"%s%s%s",ilab,iopd,oper);
   while(strcmp(iopd,"MEND")!=0)
   {
    fprintf(p[n],"%s %s %s\n",ilab,iopd,oper);
    fscanf(fp1,"%s%s%s",ilab,iopd,oper);
   }
   fclose(p[n]);
   n++;
  }
 }
 for(i=0;i<n1;i++)
 p[i]=fopen(fn[i],"r");
 fp2=fopen("outm.txt","w");
 rewind(fp1);
 fscanf(fp1,"%s%s%s",ilab,iopd,oper);
 while(!feof(fp1))
 {
  if(strcmp(iopd,"CALL")==0)
  {
   for(i=0;i<n1;i++)
   {
    if(strcmp(m[i],oper)==0)
    {
     rewind(p[i]);
     fscanf(p[i],"%s%s%s",ilab,iopd,oper);
     while(!feof(p[i]))
     {
      fprintf(fp2,"%s %s %s",ilab,iopd,oper);
      c1=1;
      fscanf(p[i],"%s%s%s",ilab,iopd,oper);
     }
     break;
    }
   }
  }
  if(c1!=1)
  fprintf(fp2,"%s %s %s\n",ilab,iopd,oper);
  c1=0;
  fscanf(fp1,"%s%s%s",ilab,iopd,oper);
 }
 fprintf(fp2,"%s %s %s\n",ilab,iopd,oper);
}
```

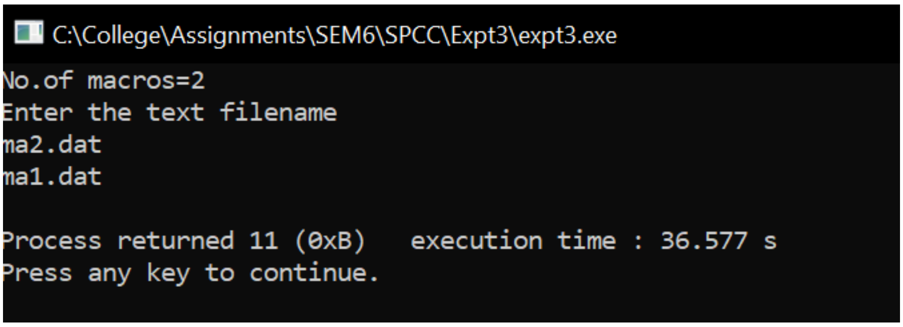
Input:

macin.txt

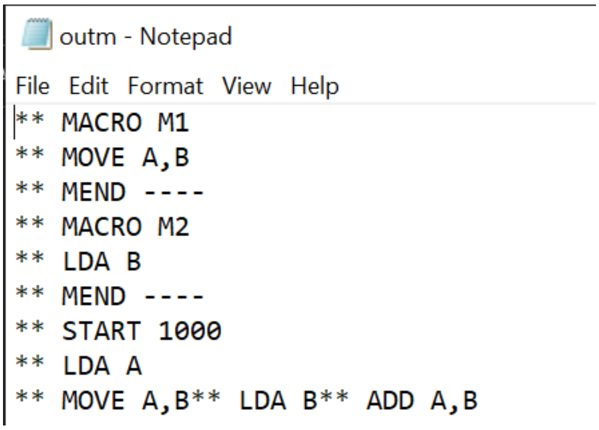


```
** MACRO M1
** MOVE A,B
** MEND ----
** MACRO M2
** LDA B
** MEND ----
** START 1000
** LDA A
** CALL M1
** CALL M2
** ADD A,B
```

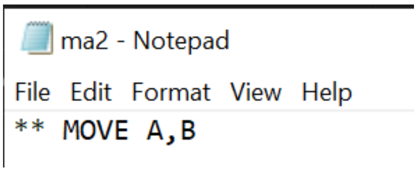
OUTPUT:



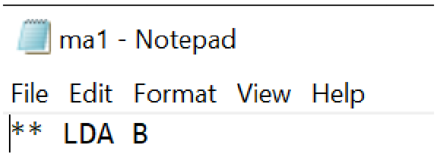
```
No.of macros=2
Enter the text filename
ma2.dat
ma1.dat

Process returned 11 (0xB)   execution time : 36.577 s
Press any key to continue.
```



```
** MACRO M1
** MOVE A,B
** MEND ----
** MACRO M2
** LDA B
** MEND ----
** START 1000
** LDA A
** MOVE A,B** LDA B** ADD A,B
```



```
** MOVE A,B
```



```
** LDA B
```

RESULT:

Thus, a two pass macro processor is implemented successfully using in C language.

Ex.No:6 IMPLEMENT A SINGLE PASS MACRO PROCESSOR

AIM:

To implement a single pass macro processor using in C language.

ALGORITHM:

STEP 1: GET THE STATEMENT FROM THE INPUT FILE

STEP 2: IF THE STATEMENT HAS THE DIRECTIVE "MACRO", THEN THE NUMBER OF MACRO "N" WILL BE INCREMENTED BY 1

STEP 3: REPEAT THE STEPS 1 AND 2 UNTIL AN END OF FILE IS ENCOUNTERED

STEP 4: OPEN "N" NUMBER OF MACRO FILES IN WRITE MODE AND REWIND THE INPUT FILE POINTER

STEP 5: IF THE DIRECTIVE IS "MACRO" THEN, DO THE FOLLOWING

STEP 5.1: ENTER THE MACRO NAME PRESENT IN THE OPERAND FIELD

STEP 5.2: WRITE THE LINE TO THE EXPANDED OUTPUT FILE

STEP 5.3: ENTER THE LINES IN THE BODY OF EACH MACRO IN TO THE CORRESPONDING FILES ALREADY OPENED IN STEP 4.

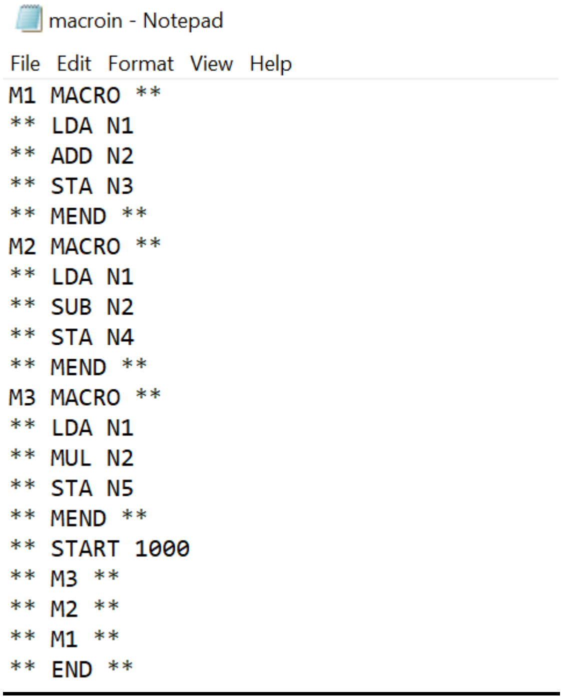
STEP 5.4: WRITE THE BODY OF EACH MACRO TO THE EXPANDED OUTPUT FILE UNTIL A "MEND" IS REACHED

STEP 6: WRITE THE REMAINING LINES DIRECTLY TO THE EXPANDED FILE.

PROGRAM:

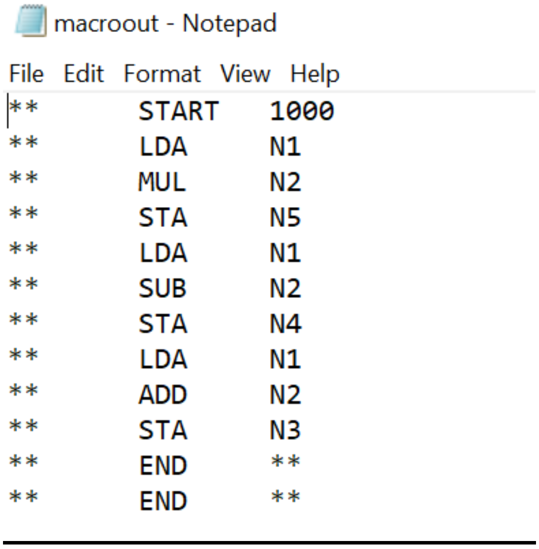
```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
 int n,flag,i;
 char ilab[20],iopd[20],oper[20],NAMTAB[20][20];
 FILE *fp1,*fp2,*DEFTAB;
 clrscr();
 fp1=fopen("macroin.dat","r");
 fp2=fopen("macroout.dat","w");
 n=0;
 rewind(fp1);
 fscanf(fp1,"%s%s%s",ilab,iopd,oper);
 while(!feof(fp1))
 {
  if(strcmp(iopd,"MACRO")==0)
  {
   strcpy(NAMTAB[n],ilab);
   DEFTAB=fopen(NAMTAB[n],"w");
   fscanf(fp1,"%s%s%s",ilab,iopd,oper);
   while(strcmp(iopd,"MEND")!=0)
   {
    fprintf(DEFTAB,"%s\t%s\t%s\n",ilab,iopd,oper);
    fscanf(fp1,"%s%s%s",ilab,iopd,oper);
   }
   fclose(DEFTAB);
   n++;
  }
  else
  {
   flag=0;
   for(i=0;i<n;i++)
   {
    if(strcmp(iopd,NAMTAB[i])==0)
    {
     flag=1;
     DEFTAB=fopen(NAMTAB[i],"r");
     fscanf(DEFTAB,"%s%s%s\n",ilab,iopd,oper);
     while(!feof(DEFTAB))
     {
      fprintf(fp2,"%s\t%s\t%s\n",ilab,iopd,oper);
      fscanf(DEFTAB,"%s%s%s",ilab,iopd,oper);
     }
     break;
    }
   }
   if(flag==0)
   fprintf(fp2,"%s\t%s\t%s\n",ilab,iopd,oper);
  }
  fscanf(fp1,"%s%s%s",ilab,iopd,oper);
 }
 fprintf(fp2,"%s\t%s\t%s\n",ilab,iopd,oper);
 getch();
}
```

INPUT:



```
M1 MACRO **
** LDA N1
** ADD N2
** STA N3
** MEND **
M2 MACRO **
** LDA N1
** SUB N2
** STA N4
** MEND **
M3 MACRO **
** LDA N1
** MUL N2
** STA N5
** MEND **
** START 1000
** M3 **
** M2 **
** M1 **
** END **
```

OUTPUT:



```
**      START   1000
**      LDA     N1
**      MUL     N2
**      STA     N5
**      LDA     N1
**      SUB     N2
**      STA     N4
**      LDA     N1
**      ADD     N2
**      STA     N3
**      END     **
**      END     **
```

RESULT:

Thus a single pass macro processor is implemented successfully in C language.