EXPERIMENT NO:

30-08-24

ONE-PASS ASSEMBLER

Aim : Implement one-pass assembler for c programming.

Algorithm :

```
begin
   read first input line
   if OPCODE ='START' then
 begin
    save #[OPERAND] as starting address
    initialize LOCCTR as starting address
    read next input line
  end [if START]
 else
    initialize LOCCTR to 0
 while opcode ≠ 'END' do
 begin
    if there is a not comment line then
    begin
      SEARCH SYNTAB for label
      if found then
  begin
        if symbol values as null
        set symbol value as LOCCTR and search the linked list
        with the corresponding operand.
        PTR addresses and generate operand addresses as corresponding
        symbol values.
        set symbol value as LOCCTR in symbol table
        end.
        else
        insert (LABEL,LOCCTR) into SYMTAB
      end
        search OPTAB for opcode
         if found then
         begin
        search SYMTAB for operand address
      if found then
         if symbol value not equal to null then
         store symbol values as operand address.
```

```
else
    insert at the end of the linkedlist
     with a node with address as LOCCTR
else
     insert [symbol, name, null]
       add 3 to LOCCTR
 end
   else if OPCODE = 'WORD' then
     add 3 to LOCCTR & convert comment to object code.
   else if OPCODE = 'RESW' then
      add 3 # [OPERAND] to LOCCTR
       else if OPCODE = 'RESB' then
        add # [OPERAND] to LOCCTR
       else if OPCODE = 'BYTE' then
  begin
       find length of constant in bytes
       add length to LOCCTR
       convert constant to object code
     end
     if object code will not fit into current
     ~~text~~ record then
     begin
       write text record to object program
       Initialize new text record
     end
        add object code to Text record
     end.
```

Result: Program executed succefully and then the output verified.

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>

int checkop(char a[]){
	int code;

	char opcode[10];
	FILE* optab=fopen("optab.txt","r");
	do{
		fscanf(optab,"%s\t%d",opcode,&code);
		if (strcmp(a,opcode)==0){
			fclose(optab);
			return code;
			}
		}while(strcmp(opcode,"END")!=0);
	fclose(optab);
	return NULL;
	}

int checksy(char a[]){
	int code;

	char opcode[10];
	FILE* symtab=fopen("symtab.txt","r");
	do{
		fscanf(symtab,"%s\t%d",opcode,&code);
		if (strcmp(a,opcode)==0){
			fclose(symtab);
			return code;
			}
		}while(strcmp(opcode,"END")!=0);
	fclose(symtab);
	printf("%s ",a);
	return NULL;
	}
```

```
void main(){
        bool symfin=true;
        char label[10],opcode[10],oprand[10],objcode[10],ch;
        FILE* obj=fopen("objcode.txt","w");
        FILE* symtab=fopen("symtab.txt","w");
        FILE* temp=fopen("temp.txt","w");
        FILE *input = fopen("oinput.txt","r");
        int lencount[200],count=0,address=0,code,scode,i,j,start,addresses[10],recordcount=0,rlens[10],rlen=0;
        fscanf(input,"%s\t%s\t%s",label,opcode,oprand);
        if(strcmp(opcode,"START")==0){
                start=atoi(oprand);
                fprintf(obj,"H^%-6s^%06d^",label,start);
                fscanf(input,"%s\t%s\t%s",label,opcode,oprand);
                }
        else{
                start=0;
                fprintf(obj,"H^progra^000000");
                }
        addresses[0]=start;
        do{
                code=checkop(opcode);
                if (code!=NULL){
                        if (symfin){
                                fclose(symtab);
                                symfin=false;
                                }
                        scode=checksy(oprand);
                        rlen+=6;
                        lencount[count++]=6;
                        fprintf(temp,"%d%d\n",code,scode);
                        address+=3;
                        }
                else{
                        fprintf(symtab,"%s\t%d\n",label,(start+address));
                        if(strcmp(opcode,"BYTE")==0){
                                for(i=2;oprand[i]!=(char)39;i++){
                                        fprintf(temp,"%x",oprand[i]);
                                        }
                                address+=i-2;
                                fprintf(temp,"\n");
                        rlen+=(i-2)*2;
                        lencount[count++]=(i-2)*2;
                        }
                else if(strcmp(opcode,"WORD")==0){
                        address+=3;
                        rlen+=6;
                        lencount[count++]=6;
                        fprintf(temp,"%06d\n",atoi(oprand));
                        }
                else if (strcmp(opcode,"RESW")==0)
                        address+=atoi(oprand)*3;
                else if (strcmp(opcode,"RESB")==0)
                        address+=atoi(oprand);
                }
        if (rlen>50){
                rlens[recordcount++]=rlen-lencount[count-1];
                addresses[recordcount]=start+address;
                rlen=lencount[count-1];
                }
        fscanf(input,"%s\t%s\t%s",label,opcode,oprand);
        }while(strcmp(opcode,"END")!=0);
rlens[recordcount++]=rlen-lencount[count-1];
fclose(temp);
temp=fopen("temp.txt","r");
fprintf(obj,"%06d\n",address);
i=0;
recordcount=0;
while(i<count){
        fprintf(obj,"\nT^%06d^%d",addresses[recordcount],rlens[recordcount]);
        recordcount++;
        for(j=0;j<50&&i<count;j+=lencount[i++]){
                fscanf(temp,"%s",objcode);
                fprintf(obj,"^%s",objcode);
                }
        }
fclose(temp);
fprintf(obj,"\n\nE^%06d\n",start);
fclose(obj);
obj=fopen("objcode.txt","r");
symtab=fopen("symtab.txt","r");
```

```
        FILE* optab=fopen("optab.txt","r");
        printf("OPTAB\n\n");
        ch=fgetc(optab);
        while (ch!=EOF){
                printf("%c",ch);
                ch=fgetc(optab);
                }
        printf("\n\nSYMTAB\n\n");
        ch=fgetc(symtab);
        while (ch!=EOF){
                printf("%c",ch);
                ch=fgetc(symtab);
                }
        printf("\n\nOBJECT FILE\n\n");
        ch=fgetc(obj);
        while (ch!=EOF){
                printf("%c",ch);
                ch=fgetc(obj);
                }
        fclose(symtab);
        fclose(obj);
        fclose(optab);
        }
```

OUTPUT

INPUT FILE

```
TEST        START       2000
ALPHA       RESW        2
FIVE        WORD        5
CHARZ       BYTE        C'Z'
C1          RESB        1
**          LDA         FIVE
**          STA         ALPHA
**          LDCH        CHARZ
**          STCH        C1
**          END         **
```

OPTAB

```
LDA   33
STA   44
LDCH 53
STCH  57
END   **
```

SYMTAB

```
ALPHA        2000
FIVE         2006
CHARZ        2009
C1           2010
```

OBJECT FILE

```
H^TEST  ^002000^000023

T^002000^26^000005^5a^332006^442000^532009^572010

E^002000
```