ARTI BHATIA

CS5A, 16

**PASS 2 ASSEMBLER**

Q- Implement a two pass assembler.

Program:

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <ctype.h>

main()

{

    FILE \*fint, \*ftab, \*flen, \*fsym, \*fout;

    int op1[10], txtlen, txtlen1, i, j = 0, len;

    char add[5], symadd[5], op[5], start[10], temp[30], line[20], label[20], mne[10], operand[10], symtab[10], opmne[10];

    clrscr();

    fint = fopen("input.txt", "r");

    flen = fopen("length.txt", "r");

    ftab = fopen("optab.txt", "r");

    fsym = fopen("symbol.txt", "r");

    fscanf(fint, "%s%s%s%s", add, label, mne, operand);

    if (strcmp(mne, "START") == 0)

    {

        strcpy(start, operand);

        fscanf(flen, "%d", &len);

    }

    printf("H^%s^%s^%d\nT^00%s^", label, start, len, start);

    fscanf(fint, "%s%s%s%s", add, label, mne, operand);

    while (strcmp(mne, "END") != 0)

    {

        fscanf(ftab, "%s%s", opmne, op);

        while (!feof(ftab))

        {

            if (strcmp(mne, opmne) == 0)

            {

                fclose(ftab);

                fscanf(fsym, "%s%s", symadd, symtab);

                while (!feof(fsym))

                {

                    if (strcmp(operand, symtab) == 0)

                    {

                        printf("%s%s^", op, symadd);

                        break;

                    }

                    else

                        fscanf(fsym, "%s%s", symadd, symtab);

                }

                break;

            }

            else

                fscanf(ftab, "%s%s", opmne, op);

        }

        if ((strcmp(mne, "BYTE") == 0) || (strcmp(mne, "WORD") == 0))

        {

            if (strcmp(mne, "WORD") == 0)

                printf("0000%s^", operand);

            else

            {

                len = strlen(operand);

                for (i = 2; i < len; i++)

                {

                    printf("%d", operand[i]);

                }

                printf("^");

            }

        }

        fscanf(fint, "%s%s%s%s", add, label, mne, operand);

        ftab = fopen("optab.txt", "r");

        fseek(ftab, SEEK\_SET, 0);

    }

    printf("\nE^00%s", start);

    fclose(fint);

    fclose(ftab);

    fclose(fsym);

    fclose(flen);

    fclose(fout);

    getch();

}

Algorithm:





