

PASS ONE OF A TWO PASS ASSEMBLER

AIM

Implement pass one of a two pass assembler.

SOURCE CODE

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

void passOne(char label[10], char opcode[10], char operand[10], char code[10],
char mnemonic[3]);

void display();

int main()
{
    char label[10], opcode[10], operand[10];
    char code[10], mnemonic[3];
    passOne(label, opcode, operand, code, mnemonic);
    return 0;
}

void passOne(char label[10], char opcode[10], char operand[10], char code[10],
char mnemonic[3])
{
    int locctr, start, length;

    FILE *fp1, *fp2, *fp3, *fp4, *fp5;

    fp1 = fopen("input.txt", "r");
    fp2 = fopen("optab.txt", "r");
    fp3 = fopen("symtab.txt", "w");
    fp4 = fopen("intermediate.txt", "w");
    fp5 = fopen("length.txt", "w");

    fscanf(fp1, "%s\t%s\t%s", label, opcode, operand);

    if (strcmp(opcode, "START") == 0) {
        start = atoi(operand);
        locctr = start;
        fprintf(fp4, "\t%s\t%s\t%s\n", label, opcode, operand);
```

```

    fscanf(fp1, "%s\t%s\t%s", label, opcode, operand);
}
else {
    locctr = 0;
}

while (strcmp(opcode, "END") != 0) {
    fprintf(fp4, "%d\t%s\t%s\t%s\n", locctr, label, opcode, operand);
    if (strcmp(label, "**") != 0) {
        fprintf(fp3, "%s\t%d\n", label, locctr);
    }
    fscanf(fp2, "%s\t%s", code, mnemonic);
    while (strcmp(code, "END") != 0) {
        if (strcmp(opcode, code) == 0) {
            locctr += 3;
            break;
        }
        fscanf(fp2, "%s\t%s", code, mnemonic);
    }
    if (strcmp(opcode, "WORD") == 0) {
        locctr += 3;
    }
    // RESW -> add 3*operand to locctr
    else if (strcmp(opcode, "RESW") == 0) {
        locctr += (3 * (atoi(operand)));
    }
    // BYTE -> add 1 to locctr
    else if (strcmp(opcode, "BYTE") == 0) {
        ++locctr;
    }
    // RESB -> add operand to locctr
    else if (strcmp(opcode, "RESB") == 0) {
        locctr += atoi(operand);
    }
    fscanf(fp1, "%s\t%s\t%s", label, opcode, operand);
}

fprintf(fp4, "%d\t%s\t%s\t%s\n", locctr, label, opcode, operand);
fclose(fp4);
fclose(fp3);
fclose(fp2);
fclose(fp1);

display();

length = locctr - start;
fprintf(fp5, "%d", length);
fclose(fp5);
printf("\nThe length of the code : %d\n", length);
}

```

```

void display() {
    char str;
    FILE *fp1, *fp2, *fp3;

    printf("\nThe contents of Input Table :\n\n");
    fp1 = fopen("input.txt", "r");
    str = fgetc(fp1);
    while (str != EOF) {
        printf("%c", str);
        str = fgetc(fp1);
    }

    fclose(fp1);

    printf("\n\nThe contents of Output Table :\n\n");
    fp2 = fopen("intermediate.txt", "r");
    str = fgetc(fp2);
    while (str != EOF) {
        printf("%c", str);
        str = fgetc(fp2);
    }

    fclose(fp2);

    printf("\n\nThe contents of Symbol Table :\n\n");
    fp3 = fopen("symtab.txt", "r");
    str = fgetc(fp3);
    while (str != EOF) {
        printf("%c", str);
        str = fgetc(fp3);
    }

    fclose(fp3);
}

```

OUTPUT

```
anjana-anjali@anjana-anjali:~/Documents/program/ss_lab/pgm$ ./a.out

The contents of Input Table :

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

The contents of Output Table :

          **      START      2000
2000      **      LDA        FIVE
2003      **      STA        ALPHA
2006      **      LDCH       CHARZ
2009      **      STCH       C1
2012      ALPHA   RESW       2
2018      FIVE    WORD       5
2021      CHARZ   BYTE      C'Z'
2022      C1      RESB       1
2023      **      END        **

The contents of Symbol Table :

ALPHA    2012
FIVE     2018
CHARZ    2021
C1       2022

The length of the code : 23
anjana-anjali@anjana-anjali:~/Documents/program/ss_lab/pgm$
```

RESULT

The program executed successfully and desired results obtained.

Submitted by,

ANJANA DILEEPKUMAR

ROLL NO :13

S5 CSE