Exp. No: 11 Date : 30-10-2020

SINGLE PASS ASSEMBLER

AIM

Implement a single pass assembler.

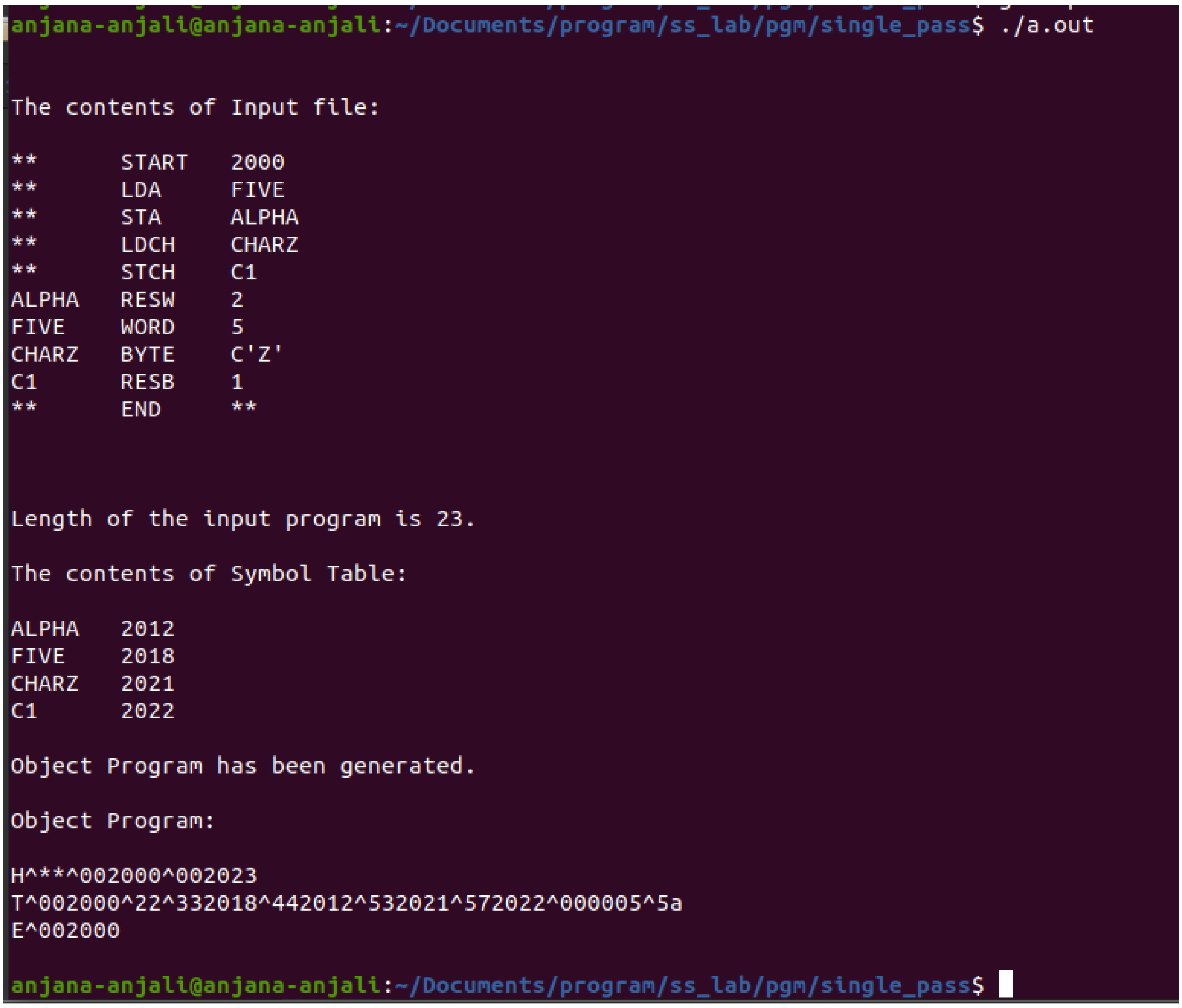
SOURCE CODE

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

int main()
{
    char opcode[10], operand[10], label[10], a[10], ad[10], symbol[10], ch;
    char code[10][10], code1[10][10] = {"33", "44", "53", "57"};
    char mnemonic[10][10] = {"START", "LDA", "STA", "LDCH", "STCH", "END"};
    char mnemonic1[10][10] = {"LDA", "STA", "LDCH", "STCH"};
    int locctr, start, length, i = 0, j = 0, k, l = 0;
    int st, diff, address, add, len, actual_len, find_addr, prev_addr;
    FILE *in, *sym, *int_file, *out, *sym_r, *int_file_r, *ob_c;

    in= fopen("input.txt", "r");
    sym = fopen("symtab.txt", "w");
    int_file = fopen("intermediate.txt", "w");
    fscanf(in, "%s%s%s", label, opcode, operand);

    if (strcmp(opcode, "START") == 0)
    {
        start = atoi(operand);
        locctr = start;
        fprintf(int_file, "%s\t%s\t%s\n", label, opcode, operand);
        fscanf(in, "%s%s%s", label, opcode, operand);
    }
    else
        locctr = 0;

    while (strcmp(opcode, "END") != 0)
    {
        fprintf(int_file, "%d", locctr);

        if (strcmp(label, "**") != 0)
            fprintf(sym, "%s\t%d\n", label, locctr);

        strcpy(code[i], mnemonic[j]);

        while (strcmp(mnemonic[j], "END") != 0)
        {
```

```
        if (strcmp(opcode, mnemonic[j]) == 0)
        {
            locctr += 3;
            break;
        }
        strcpy(code[i], mnemonic[j]);
        j++;
    }
    if (strcmp(opcode, "WORD") == 0)
        locctr += 3;
    else if (strcmp(opcode, "RESW") == 0)
        locctr += (3 * (atoi(operand)));
    else if (strcmp(opcode, "RESB") == 0)
        locctr += (atoi(operand));
    else if (strcmp(opcode, "BYTE") == 0)
        ++locctr;
    fprintf(int_file, "\t%s\t%s\t%s\n", label, opcode, operand);
    fscanf(in, "%s%s%s", label, opcode, operand);
}

fprintf(int_file, "%d\t%s\t%s\t%s\n", locctr, label, opcode, operand);
length = locctr - start;

fclose(int_file);
fclose(sym);
fclose(in);

printf("\n\nThe contents of Input file:\n\n");
in = fopen("input.txt", "r");
ch = fgetc(in);
while (ch != EOF)
{
    printf("%c", ch);
    ch = fgetc(in);
}
printf("\n\nLength of the input program is %d.", length);
printf("\n\nThe contents of Symbol Table:\n\n");

sym = fopen("symtab.txt", "r");
ch = fgetc(sym);
while (ch != EOF)
{
    printf("%c", ch);
    ch = fgetc(sym);
}

fclose(sym);
fclose(in);

out = fopen("output.txt", "w");
```

```
    sym_r = fopen("symtab.txt", "r");
    int_file_r = fopen("intermediate.txt", "r");
    ob_c = fopen("objcode.txt", "w");
    fscanf(int_file_r, "%s%s%s", label, opcode, operand);


    while (strcmp(opcode, "END") != 0)
    {
        prev_addr = address;
        fscanf(int_file_r, "%d%s%s%s", &address, label, opcode, operand);
    }
    find_addr = address;
    fclose(int_file_r);

    int_file_r = fopen("intermediate.txt", "r");
    fscanf(int_file_r, "%s%s%s", label, opcode, operand);
    if (strcmp(opcode, "START") == 0)
    {
        fprintf(out, "\t%s\t%s\t%s\n", label, opcode, operand);
        fprintf(ob_c, "H^%s^00%s^00%d\n", label, operand, find_addr);
        fscanf(int_file_r, "%d%s%s%s", &address, label, opcode, operand);
        st = address;
        diff = prev_addr - st;
        fprintf(ob_c, "T^00%d^%d", address, diff);
    }

    while (strcmp(opcode, "END") != 0)
    {

        if (strcmp(opcode, "BYTE") == 0)
        {
            fprintf(out, "%d\t%s\t%s\t%s\t", address, label, opcode, operand);
            len = strlen(operand);
            actual_len = len - 3;
            fprintf(ob_c, "^");
            for (k = 2; k < (actual_len + 2); k++)
            {
                sprintf( ad, "%x",operand[k]);
                fprintf(out, "%s", ad);
                fprintf(ob_c, "%s", ad);
            }
            fprintf(out, "\n");
        }
        else if (strcmp(opcode, "WORD") == 0)
        {
            len = strlen(operand);
            sprintf( a,"%d",atoi(operand));
            fprintf(out, "%d\t%s\t%s\t%s\t00000%s\n", address, label, opcode,
operand, a);
            fprintf(ob_c, "^00000%s", a);
        }
        else if ((strcmp(opcode, "RESB") == 0) || (strcmp(opcode, "RESW") == 0))
            fprintf(out, "%d\t%s\t%s\t%s\n", address, label, opcode, operand);
        else
        {
            while (strcmp(opcode, mnemonic1[l]) != 0)
                l++;
            if (strcmp(operand, "COPY") == 0)
                fprintf(out, "%d\t%s\t%s\t%s\t%s0000\n", address, label, opcode,
operand, code1[l]);
            else
            {
                rewind(sym_r);
                fscanf(sym_r, "%s%d", symbol, &add);
                while (strcmp(operand, symbol) != 0)
                    fscanf(sym_r, "%s%d", symbol, &add);
                fprintf(out, "%d\t%s\t%s\t%s\t%s%d\n", address, label, opcode,
operand, code1[l], add);
                fprintf(ob_c, "^%s%d", code1[l], add);
            }
        }
        fscanf(int_file_r, "%d%s%s%s", &address, label, opcode, operand);
    }

    fprintf(out, "%d\t%s\t%s\t%s\n", address, label, opcode, operand);
    fprintf(ob_c, "\nE^00%d", st);
    printf("\nObject Program has been generated.");

    fclose(ob_c);
    fclose(int_file_r);
    fclose(sym_r);
    fclose(out);

    printf("\n\nObject Program:\n\n");
    ob_c = fopen("objcode.txt", "r");
    ch = fgetc(ob_c);
    while (ch != EOF)
    {
        printf("%c", ch);
        ch = fgetc(ob_c);
    }
    printf("\n\n");
    fclose(ob_c);

    return 0;
}
```

OUTPUT



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


The contents of Input file:

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



Length of the input program is 23.

The contents of Symbol Table:

ALPHA   2012
FIVE    2018
CHARZ   2021
C1      2022

Object Program has been generated.

Object Program:

H^**^002000^002023
T^002000^22^332018^442012^532021^572022^000005^5a
E^002000

anjana-anjali@anjana-anjali:~/Documents/program/ss_lab/pgm/single_pass$ 
```

Submitted by,

ANJANA DILEEPKUMAR
ROLL NO :13
S5 CSE