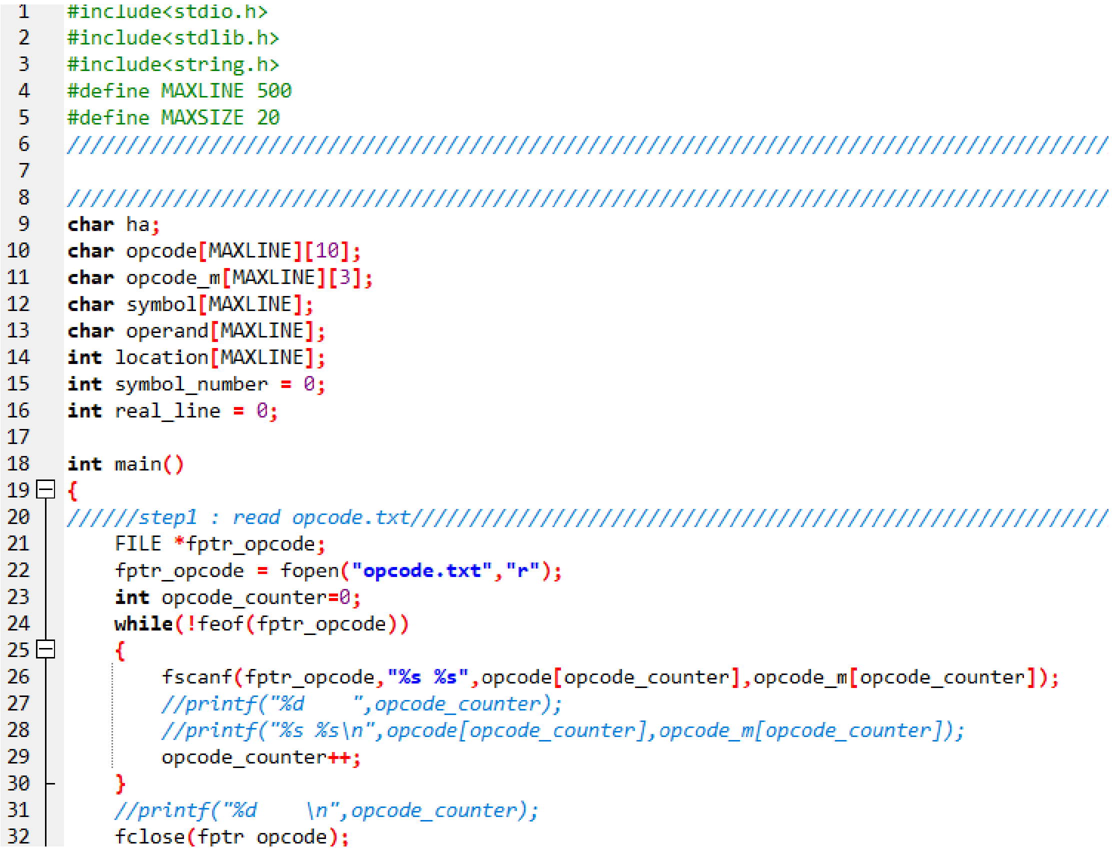
Pass1

1.

用 2 維陣列讀 opcode

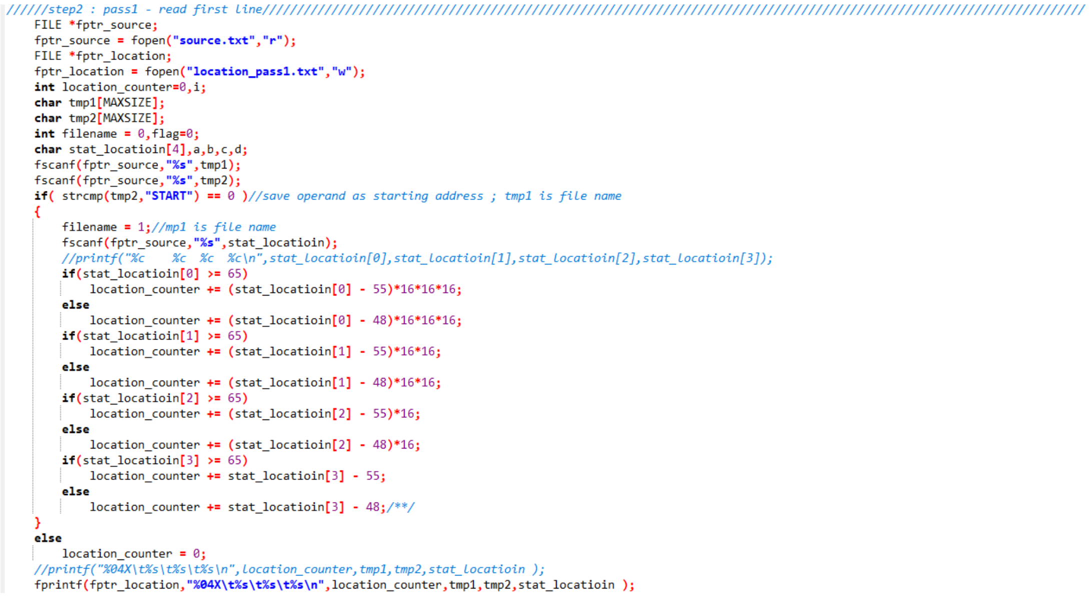


```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAXLINE 500
#define MAXSIZE 20
/////////////////////////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////////////////////////
char ha;
char opcode[MAXLINE][10];
char opcode_m[MAXLINE][3];
char symbol[MAXLINE];
char operand[MAXLINE];
int location[MAXLINE];
int symbol_number = 0;
int real_line = 0;

int main()
{
//////step1 : read opcode.txt//////////////////////////////////////////////////////////
    FILE *fptr_opcode;
    fptr_opcode = fopen("opcode.txt","r");
    int opcode_counter=0;
    while(!feof(fptr_opcode))
    {
        fscanf(fptr_opcode,"%s %s",opcode[opcode_counter],opcode_m[opcode_counter]);
        //printf("%d     ",opcode_counter);
        //printf("%s %s\n",opcode[opcode_counter],opcode_m[opcode_counter]);
        opcode_counter++;
    }
    //printf("%d     \n",opcode_counter);
    fclose(fptr_opcode);
```

2.

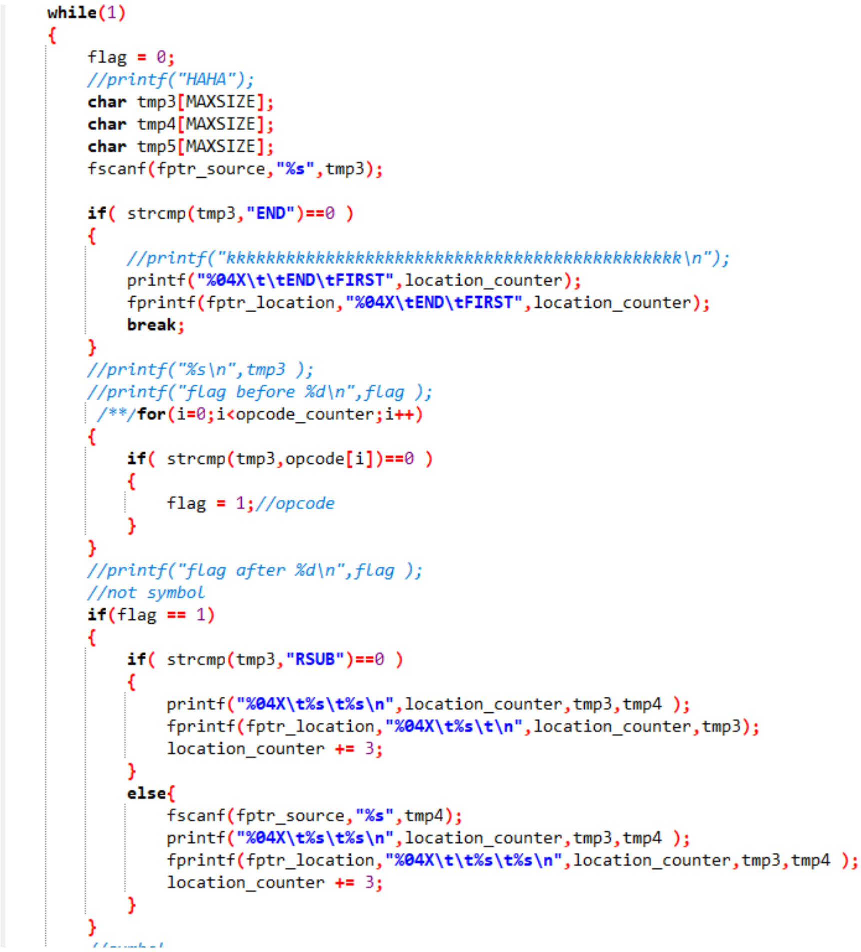
讀第一行，若 opcode == start，紀錄 filename,紀錄 location counter

若不是，

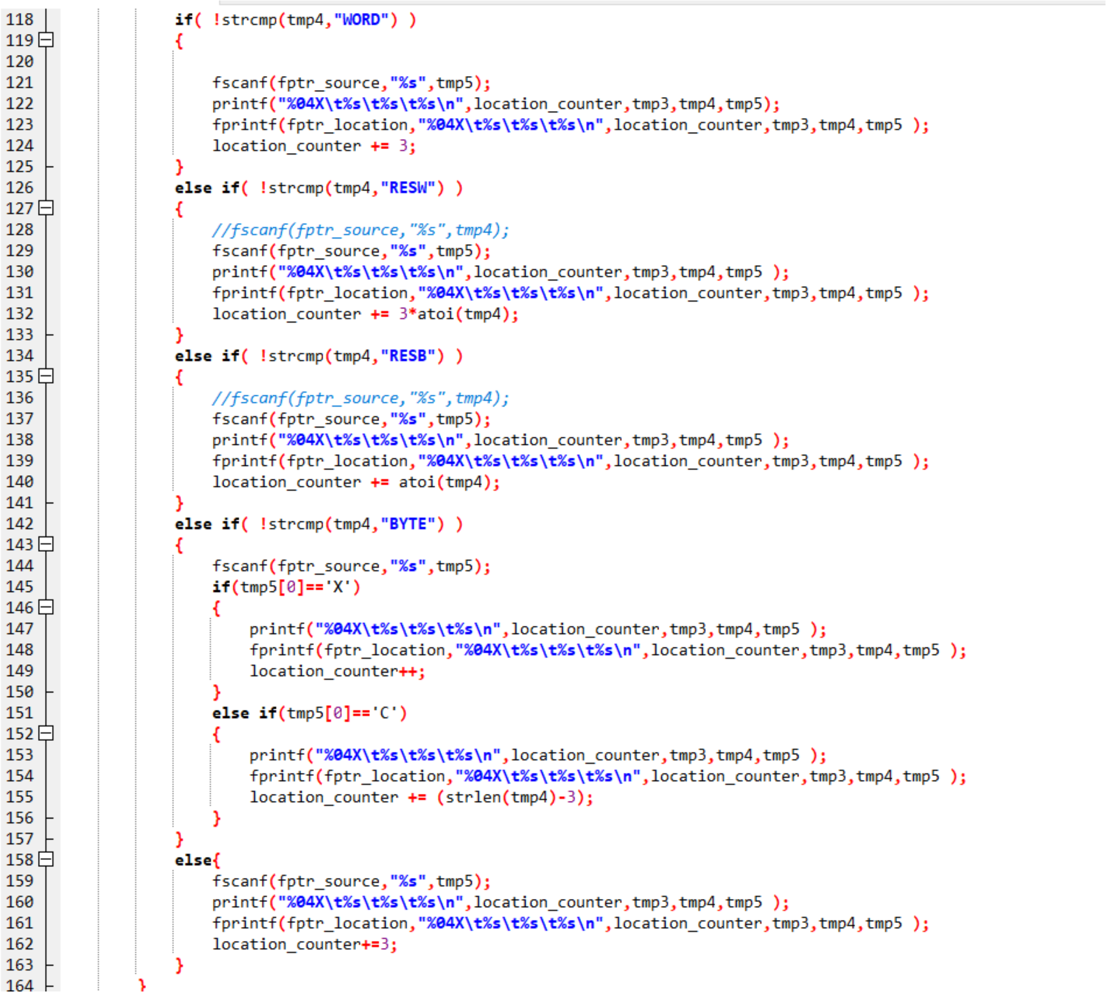


```c
//////step2 : pass1 - read first line//////////////////////////////////////////////////////////////////////////////////////////////////
    FILE *fptr_source;
    fptr_source = fopen("source.txt","r");
    FILE *fptr_location;
    fptr_location = fopen("location_pass1.txt","w");
    int location_counter=0,i;
    char tmp1[MAXSIZE];
    char tmp2[MAXSIZE];
    int filename = 0,flag=0;
    char stat_locatioin[4],a,b,c,d;
    fscanf(fptr_source,"%s",tmp1);
    fscanf(fptr_source,"%s",tmp2);
    if( strcmp(tmp2,"START") == 0 )//save operand as starting address ; tmp1 is file name
    {
        filename = 1;//mp1 is file name
        fscanf(fptr_source,"%s",stat_locatioin);
        //printf("%c    %c  %c  %c\n",stat_locatioin[0],stat_locatioin[1],stat_locatioin[2],stat_locatioin[3]);
        if(stat_locatioin[0] >= 65)
            location_counter += (stat_locatioin[0] - 55)*16*16*16;
        else
            location_counter += (stat_locatioin[0] - 48)*16*16*16;
        if(stat_locatioin[1] >= 65)
            location_counter += (stat_locatioin[1] - 55)*16*16;
        else
            location_counter += (stat_locatioin[1] - 48)*16*16;
        if(stat_locatioin[2] >= 65)
            location_counter += (stat_locatioin[2] - 55)*16;
        else
            location_counter += (stat_locatioin[2] - 48)*16;
        if(stat_locatioin[3] >= 65)
            location_counter += stat_locatioin[3] - 55;
        else
            location_counter += stat_locatioin[3] - 48;/**/
    }
    else
        location_counter = 0;
    //printf("%04X\t%s\t%s\t%s\n",location_counter,tmp1,tmp2,stat_locatioin );
    fprintf(fptr_location,"%04X\t%s\t%s\t%s\n",location_counter,tmp1,tmp2,stat_locatioin );
```

3 整理其他航，做出 SYMTAB



```
while(1)
{
    flag = 0;
    //printf("HAHA");
    char tmp3[MAXSIZE];
    char tmp4[MAXSIZE];
    char tmp5[MAXSIZE];
    fscanf(fptr_source,"%s",tmp3);

    if( strcmp(tmp3,"END")==0 )
    {
        //printf("kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk\n");
        printf("%04X\t\tEND\tFIRST",location_counter);
        fprintf(fptr_location,"%04X\tEND\tFIRST",location_counter);
        break;
    }
    //printf("%s\n",tmp3 );
    //printf("flag before %d\n",flag );
     /**/for(i=0;i<opcode_counter;i++)
    {
        if( strcmp(tmp3,opcode[i])==0 )
        {
            flag = 1;//opcode
        }
    }
    //printf("flag after %d\n",flag );
    //not symbol
    if(flag == 1)
    {
        if( strcmp(tmp3,"RSUB")==0 )
        {
            printf("%04X\t%s\t%s\n",location_counter,tmp3,tmp4 );
            fprintf(fptr_location,"%04X\t%s\t\n",location_counter,tmp3);
            location_counter += 3;
        }
        else{
            fscanf(fptr_source,"%s",tmp4);
            printf("%04X\t%s\t%s\n",location_counter,tmp3,tmp4 );
            fprintf(fptr_location,"%04X\t\t%s\t%s\n",location_counter,tmp3,tmp4 );
            location_counter += 3;
        }
    }
    //symbol
```



```
    if( !strcmp(tmp4,"WORD") )
    {

        fscanf(fptr_source,"%s",tmp5);
        printf("%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5);
        fprintf(fptr_location,"%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
        location_counter += 3;
    }
    else if( !strcmp(tmp4,"RESW") )
    {
        //fscanf(fptr_source,"%s",tmp4);
        fscanf(fptr_source,"%s",tmp5);
        printf("%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
        fprintf(fptr_location,"%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
        location_counter += 3*atoi(tmp4);
    }
    else if( !strcmp(tmp4,"RESB") )
    {
        //fscanf(fptr_source,"%s",tmp4);
        fscanf(fptr_source,"%s",tmp5);
        printf("%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
        fprintf(fptr_location,"%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
        location_counter += atoi(tmp4);
    }
    else if( !strcmp(tmp4,"BYTE") )
    {
        fscanf(fptr_source,"%s",tmp5);
        if(tmp5[0]=='X')
        {
            printf("%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
            fprintf(fptr_location,"%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
            location_counter++;
        }
        else if(tmp5[0]=='C')
        {
            printf("%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
            fprintf(fptr_location,"%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
            location_counter += (strlen(tmp4)-3);
        }
    }
    else{
        fscanf(fptr_source,"%s",tmp5);
        printf("%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
        fprintf(fptr_location,"%04X\t%s\t%s\t%s\n",location_counter,tmp3,tmp4,tmp5 );
        location_counter+=3;
    }
}
```