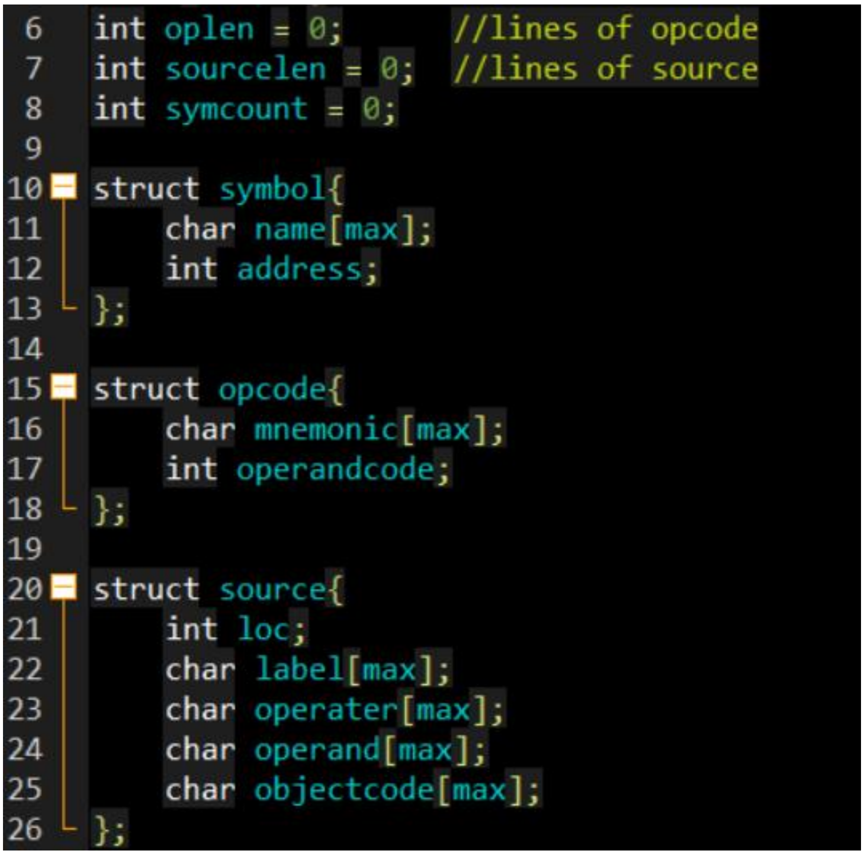
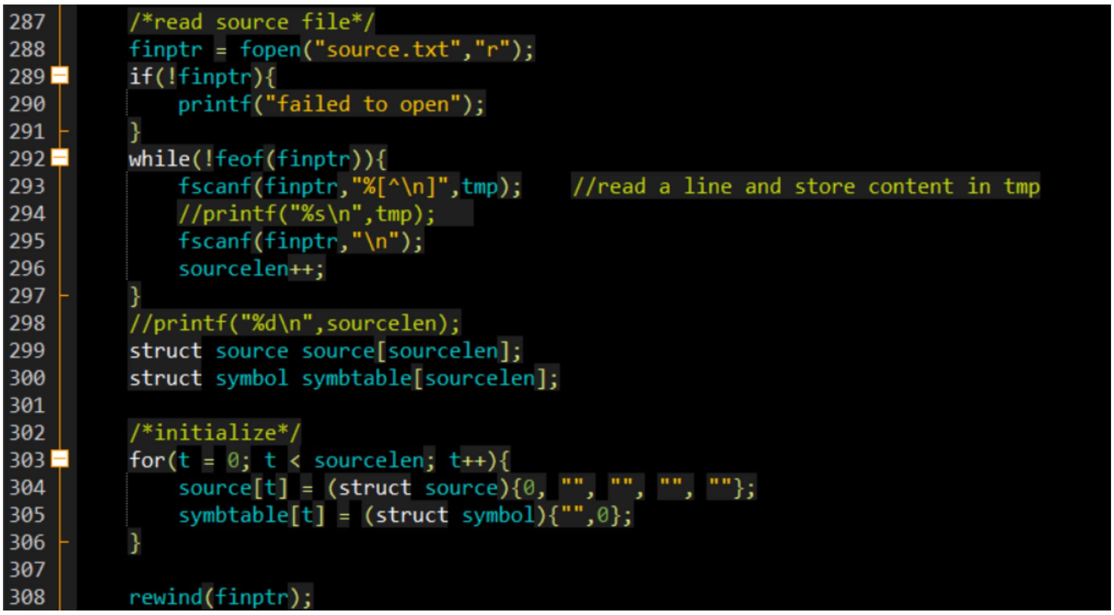
結構分別記錄 symbol, opcode, source program 所需使用之變數。



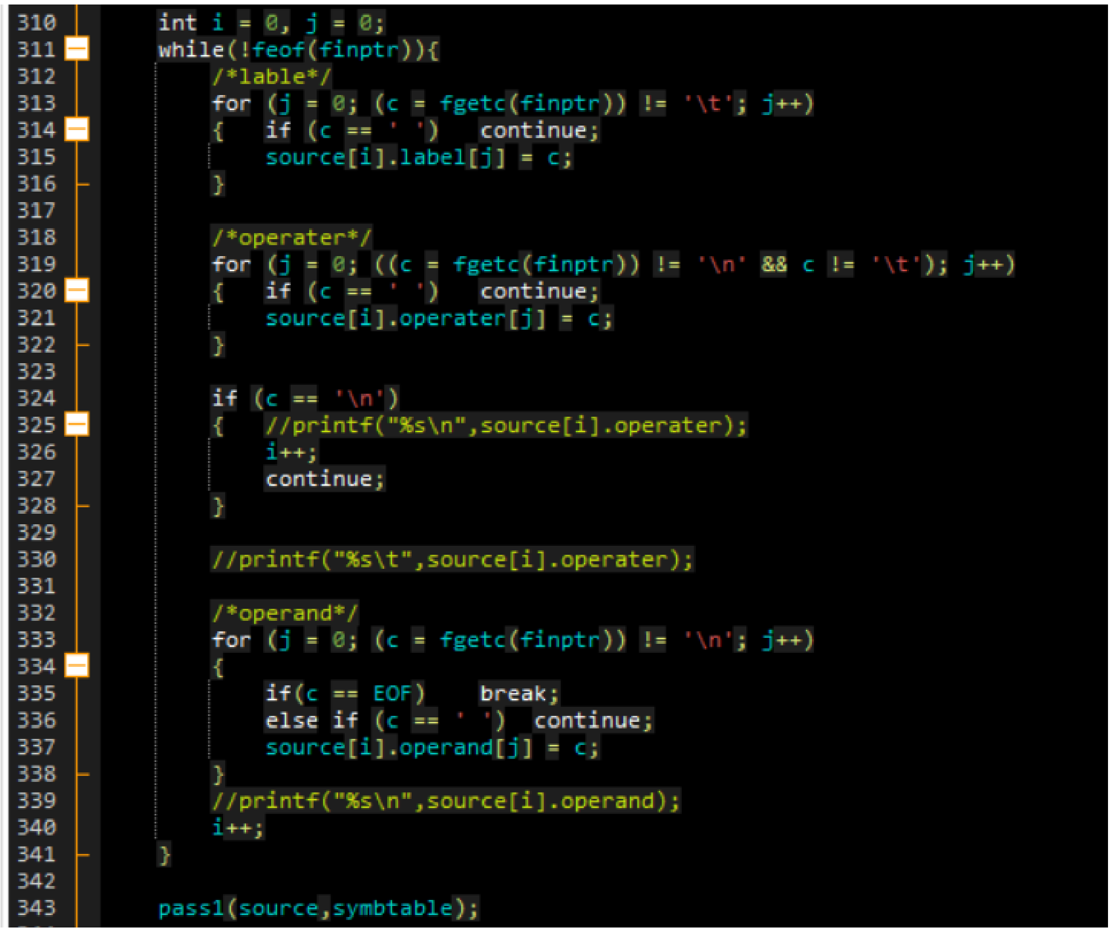
```c
int oplen = 0;      //lines of opcode
int sourcelen = 0;  //lines of source
int symcount = 0;

struct symbol{
    char name[max];
    int address;
};

struct opcode{
    char mnemonic[max];
    int operandcode;
};

struct source{
    int loc;
    char label[max];
    char operater[max];
    char operand[max];
    char objectcode[max];
};
```

讀取 source 檔後，數出有幾行，並令指標 finptr 回到檔案開頭，同時初始化結構 source 和結構 symbtable。



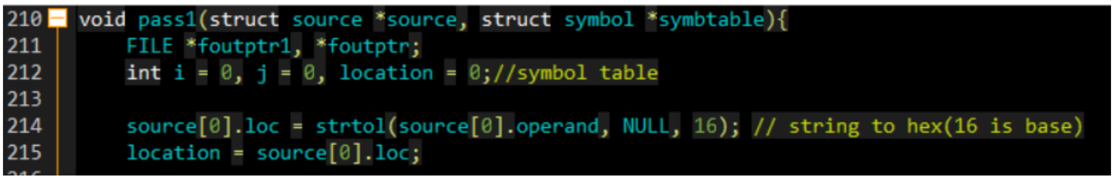
```c
    /*read source file*/
    finptr = fopen("source.txt","r");
    if(!finptr){
        printf("failed to open");
    }
    while(!feof(finptr)){
        fscanf(finptr,"%[^\n]",tmp);     //read a line and store content in tmp
        //printf("%s\n",tmp);
        fscanf(finptr,"\n");
        sourcelen++;
    }
    //printf("%d\n",sourcelen);
    struct source source[sourcelen];
    struct symbol symbtable[sourcelen];

    /*initialize*/
    for(t = 0; t < sourcelen; t++){
        source[t] = (struct source){0, "", "", "", ""};
        symbtable[t] = (struct symbol){"",0};
    }

    rewind(finptr);
```

讀到檔案最後才停止。以’\t’當作變數的分隔，若讀到空白字元則略過，其他輸入至結構 source 裡。然後呼叫函式 pass1。



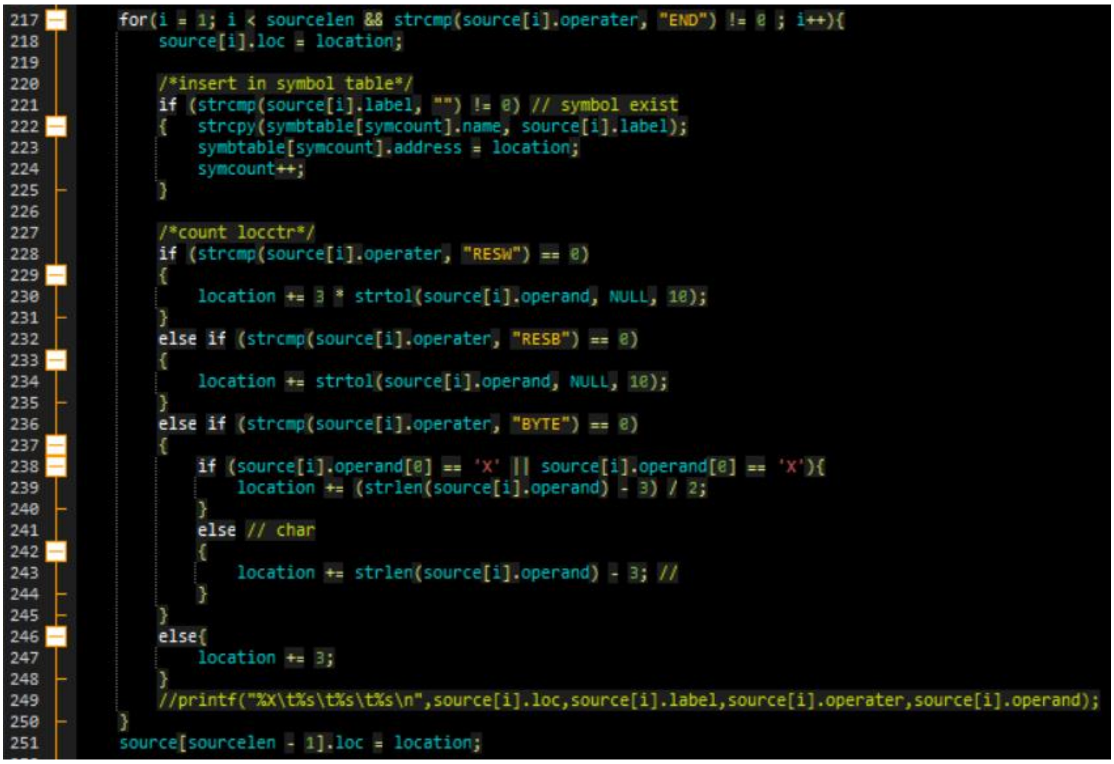
```c
int i = 0, j = 0;
while(!feof(finptr)){
    /*lable*/
    for (j = 0; (c = fgetc(finptr)) != '\t'; j++)
    {   if (c == ' ')   continue;
        source[i].label[j] = c;
    }

    /*operater*/
    for (j = 0; ((c = fgetc(finptr)) != '\n' && c != '\t'); j++)
    {   if (c == ' ')   continue;
        source[i].operater[j] = c;
    }

    if (c == '\n')
    {   //printf("%s\n",source[i].operater);
        i++;
        continue;
    }

    //printf("%s\t",source[i].operater);

    /*operand*/
    for (j = 0; (c = fgetc(finptr)) != '\n'; j++)
    {
        if(c == EOF)    break;
        else if (c == ' ')  continue;
        source[i].operand[j] = c;
    }
    //printf("%s\n",source[i].operand);
    i++;
}

pass1(source,symbtable);
```

用 strol 將 source 裡面 operand 用 16 進位的方式存在 source location 中，location 紀錄現在位置。



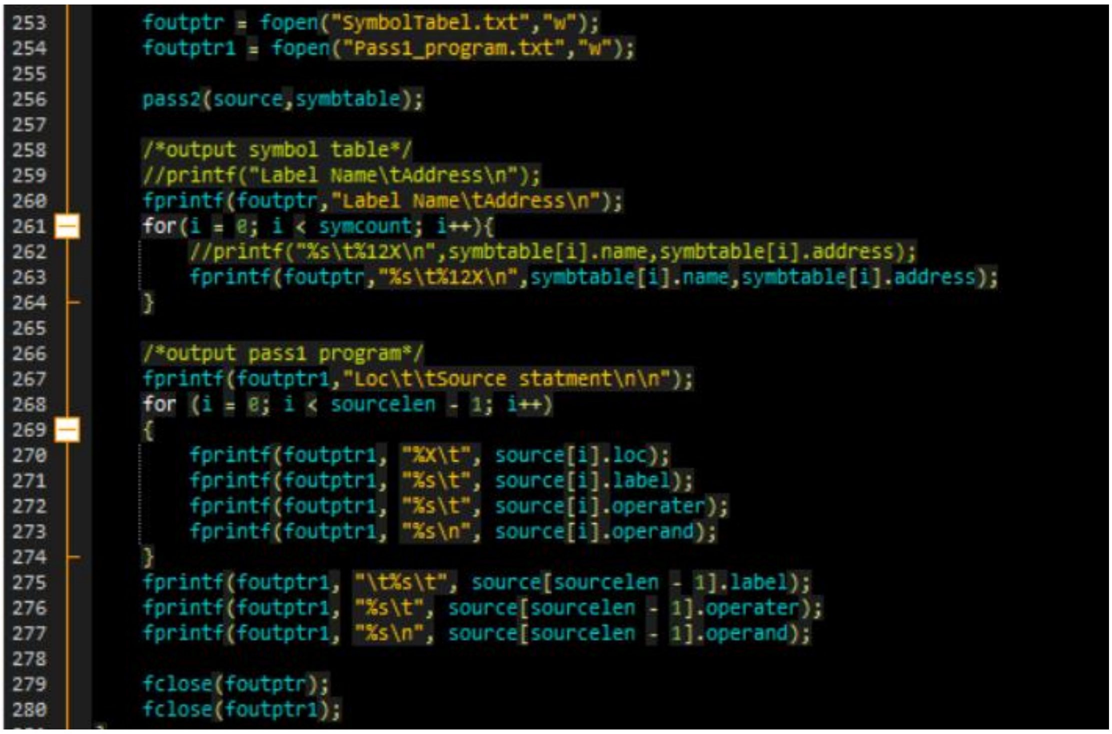
```c
void pass1(struct source *source, struct symbol *symbtable){
    FILE *foutptr1, *foutptr;
    int i = 0, j = 0, location = 0;//symbol table

    source[0].loc = strtol(source[0].operand, NULL, 16); // string to hex(16 is base)
    location = source[0].loc;
```

當沒有讀到 END 時繼續讀資料，先看 symbol 是否存在，再付於位置。
比對 operater 為何種狀況，計算下一個 location 位置。

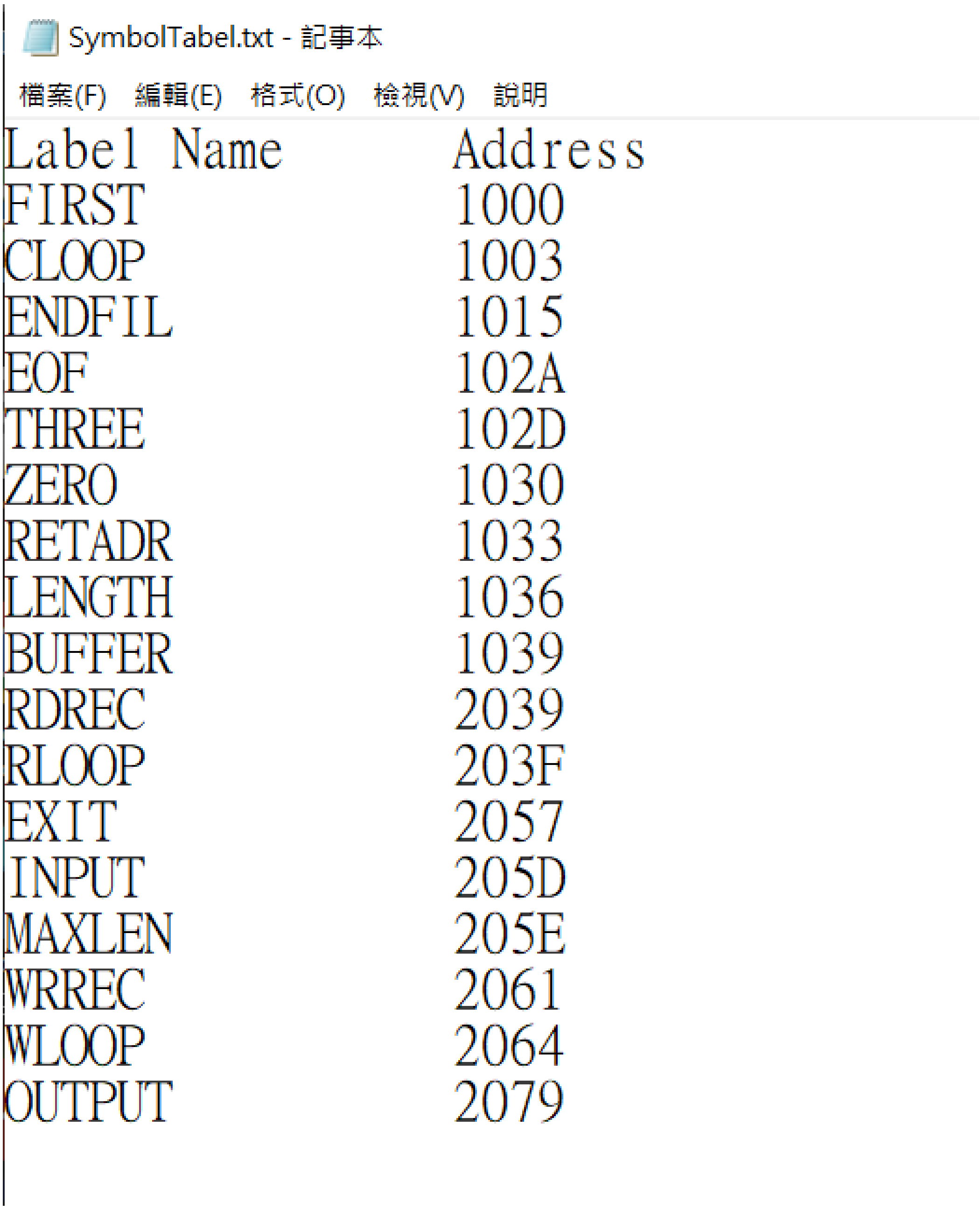


```
    for(i = 1; i < sourcelen && strcmp(source[i].operater, "END") != 0 ; i++){
        source[i].loc = location;

        /*insert in symbol table*/
        if (strcmp(source[i].label, "") != 0) // symbol exist
        {   strcpy(symbtable[symcount].name, source[i].label);
            symbtable[symcount].address = location;
            symcount++;
        }

        /*count locctr*/
        if (strcmp(source[i].operater, "RESW") == 0)
        {
            location += 3 * strtol(source[i].operand, NULL, 10);
        }
        else if (strcmp(source[i].operater, "RESB") == 0)
        {
            location += strtol(source[i].operand, NULL, 10);
        }
        else if (strcmp(source[i].operater, "BYTE") == 0)
        {
            if (source[i].operand[0] == 'X' || source[i].operand[0] == 'X'){
                location += (strlen(source[i].operand) - 3) / 2;
            }
            else // char
            {
                location += strlen(source[i].operand) - 3; //
            }
        }
        else{
            location += 3;
        }
        //printf("%X\t%s\t%s\t%s\n",source[i].loc,source[i].label,source[i].operater,source[i].operand);
    }
    source[sourcelen - 1].loc = location;
```

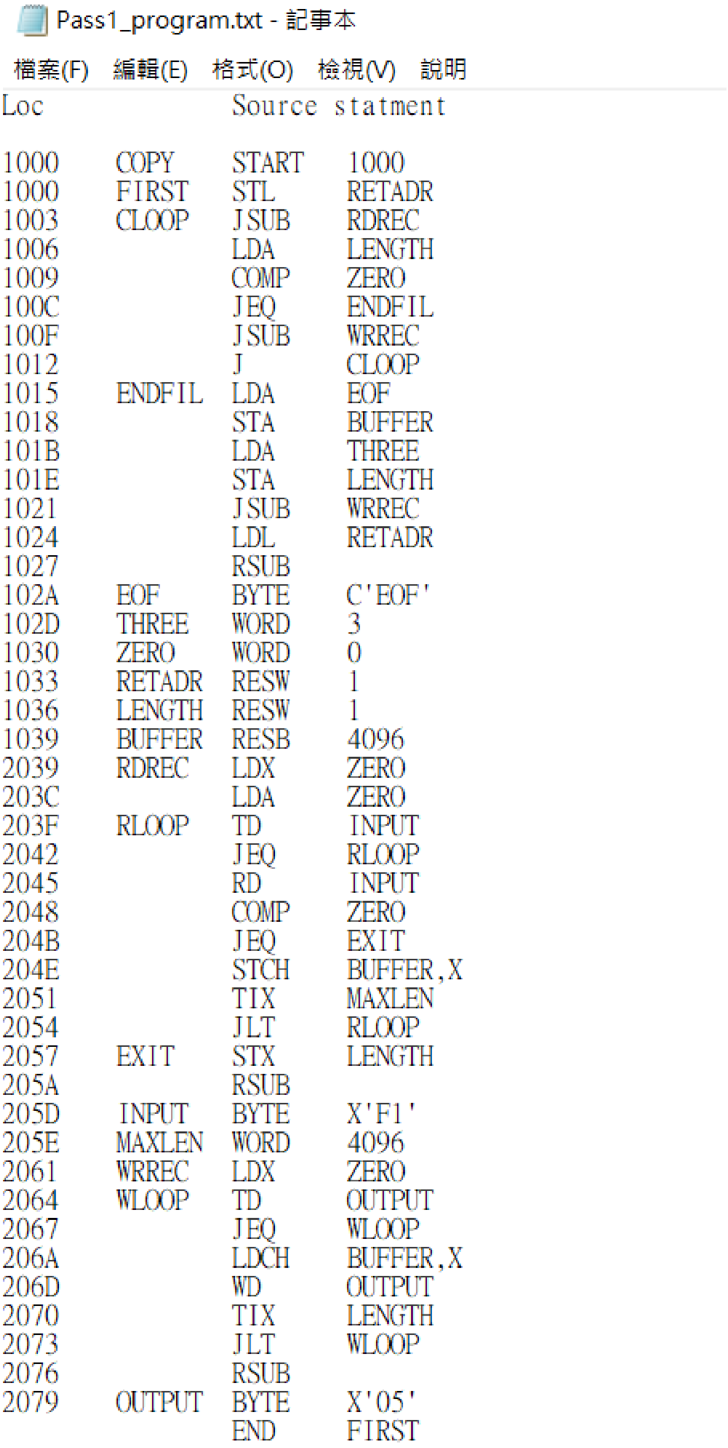
將資料寫入檔案中，並呼叫 pass2 函式。



```
    foutptr = fopen("SymbolTabel.txt","w");
    foutptr1 = fopen("Pass1_program.txt","w");

    pass2(source,symbtable);

    /*output symbol table*/
    //printf("Label Name\tAddress\n");
    fprintf(foutptr,"Label Name\tAddress\n");
    for(i = 0; i < symcount; i++){
        //printf("%s\t%12X\n",symbtable[i].name,symbtable[i].address);
        fprintf(foutptr,"%s\t%12X\n",symbtable[i].name,symbtable[i].address);
    }

    /*output pass1 program*/
    fprintf(foutptr1,"Loc\t\tSource statment\n\n");
    for (i = 0; i < sourcelen - 1; i++)
    {
        fprintf(foutptr1, "%X\t", source[i].loc);
        fprintf(foutptr1, "%s\t", source[i].label);
        fprintf(foutptr1, "%s\t", source[i].operater);
        fprintf(foutptr1, "%s\n", source[i].operand);
    }
    fprintf(foutptr1, "\t%s\t", source[sourcelen - 1].label);
    fprintf(foutptr1, "%s\t", source[sourcelen - 1].operater);
    fprintf(foutptr1, "%s\n", source[sourcelen - 1].operand);

    fclose(foutptr);
    fclose(foutptr1);
```

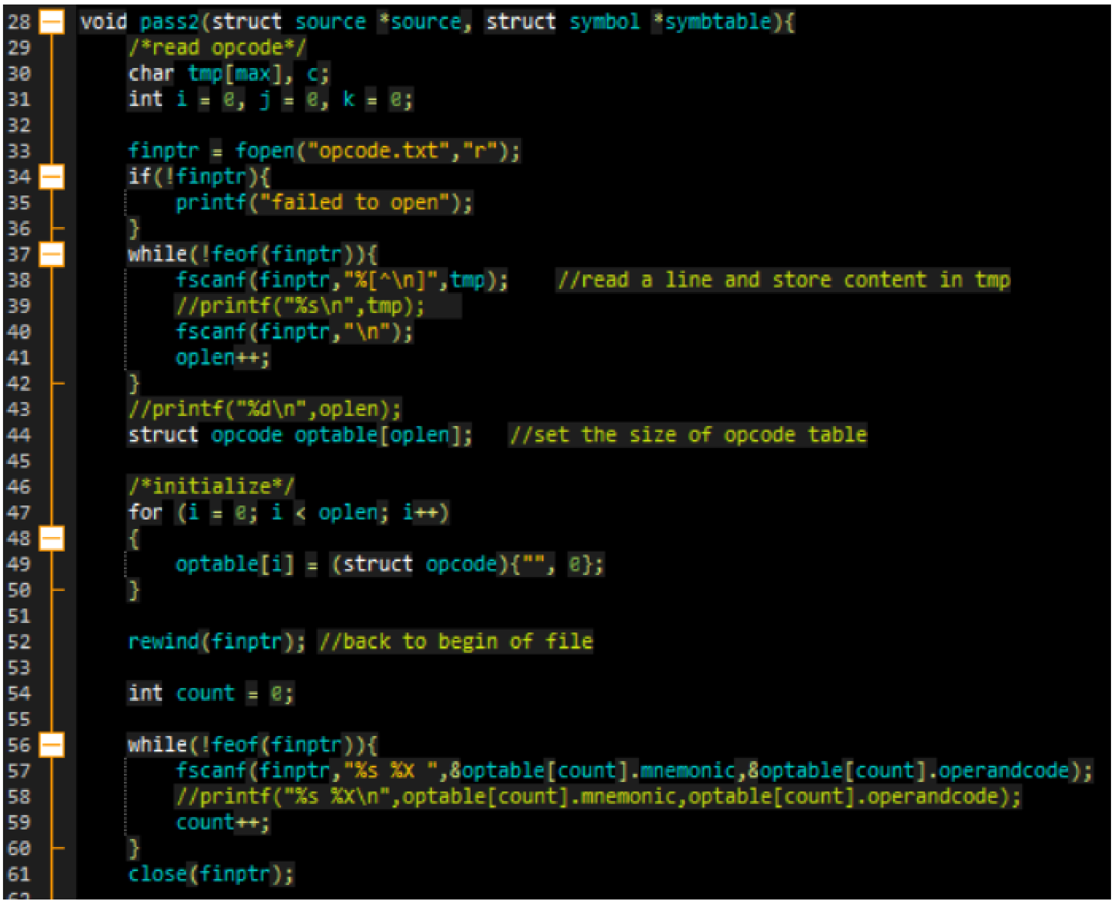


```
Label Name      Address
FIRST           1000
CLOOP           1003
ENDFIL          1015
EOF             102A
THREE           102D
ZERO            1030
RETADR          1033
LENGTH          1036
BUFFER          1039
RDREC           2039
RLOOP           203F
EXIT            2057
INPUT           205D
MAXLEN          205E
WRREC           2061
WLOOP           2064
OUTPUT          2079
```


Pass1_program.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

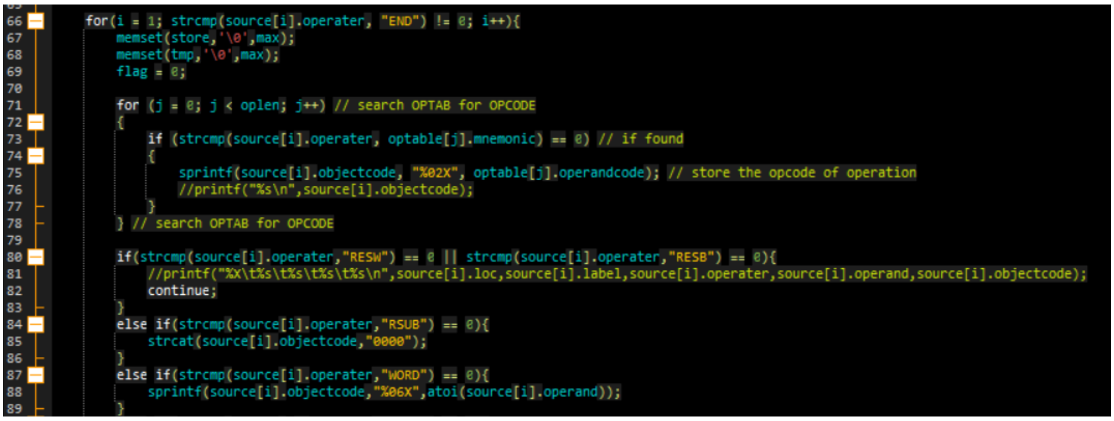
```
Loc             Source statment

1000    COPY    START   1000
1000    FIRST   STL     RETADR
1003    CLOOP   JSUB    RDREC
1006            LDA     LENGTH
1009            COMP    ZERO
100C            JEQ     ENDFIL
100F            JSUB    WRREC
1012            J       CLOOP
1015    ENDFIL  LDA     EOF
1018            STA     BUFFER
101B            LDA     THREE
101E            STA     LENGTH
1021            JSUB    WRREC
1024            LDL     RETADR
1027            RSUB
102A    EOF     BYTE    C'EOF'
102D    THREE   WORD    3
1030    ZERO    WORD    0
1033    RETADR  RESW    1
1036    LENGTH  RESW    1
1039    BUFFER  RESB    4096
2039    RDREC   LDX     ZERO
203C            LDA     ZERO
203F    RLOOP   TD      INPUT
2042            JEQ     RLOOP
2045            RD      INPUT
2048            COMP    ZERO
204B            JEQ     EXIT
204E            STCH    BUFFER,X
2051            TIX     MAXLEN
2054            JLT     RLOOP
2057    EXIT    STX     LENGTH
205A            RSUB
205D    INPUT   BYTE    X'F1'
205E    MAXLEN  WORD    4096
2061    WRREC   LDX     ZERO
2064    WLOOP   TD      OUTPUT
2067            JEQ     WLOOP
206A            LDCH    BUFFER,X
206D            WD      OUTPUT
2070            TIX     LENGTH
2073            JLT     WLOOP
2076            RSUB
2079    OUTPUT  BYTE    X'05'
                END     FIRST
```

讀取 opcode 檔案並計算有幾行，初始化結構 optable，並回到檔案最開始，並將 opcode 檔案裡的變數，儲存至 optable。

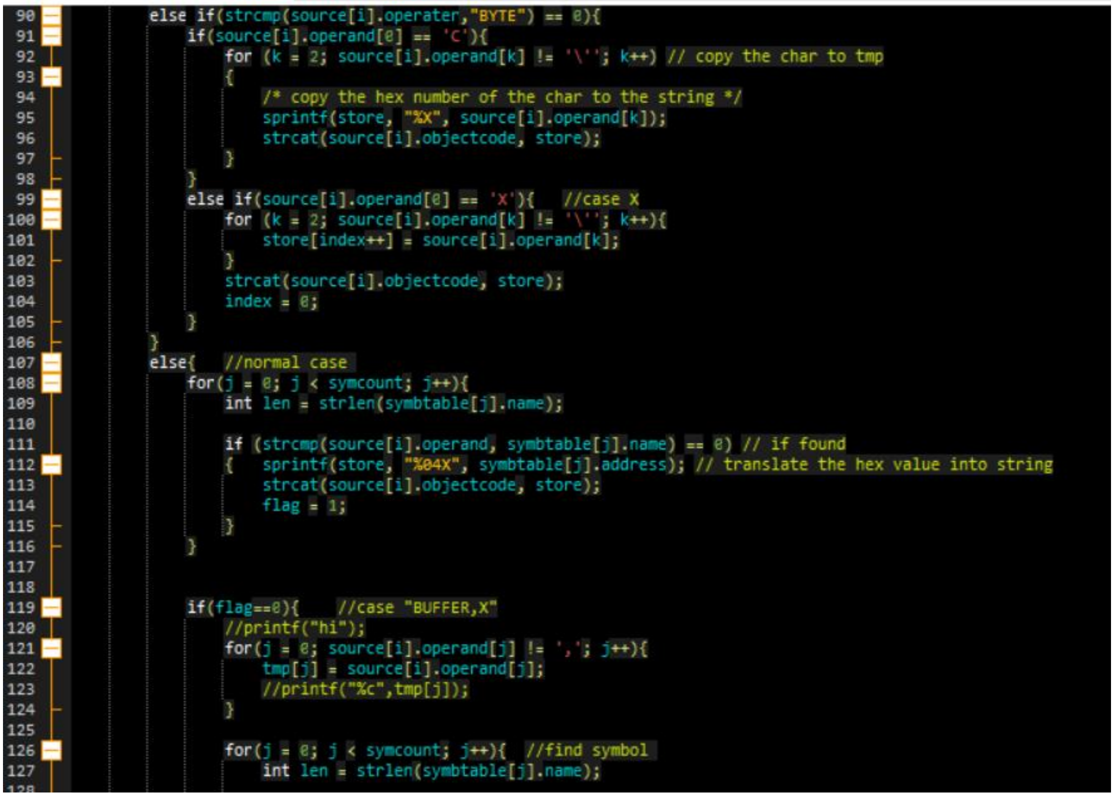


```
void pass2(struct source *source, struct symbol *symbtable){
    /*read opcode*/
    char tmp[max], c;
    int i = 0, j = 0, k = 0;

    finptr = fopen("opcode.txt","r");
    if(!finptr){
        printf("failed to open");
    }
    while(!feof(finptr)){
        fscanf(finptr,"%[^\n]",tmp);     //read a line and store content in tmp
        //printf("%s\n",tmp);
        fscanf(finptr,"\n");
        oplen++;
    }
    //printf("%d\n",oplen);
    struct opcode optable[oplen];   //set the size of opcode table

    /*initialize*/
    for (i = 0; i < oplen; i++)
    {
        optable[i] = (struct opcode){"", 0};
    }

    rewind(finptr); //back to begin of file

    int count = 0;

    while(!feof(finptr)){
        fscanf(finptr,"%s %X ",&optable[count].mnemonic,&optable[count].operandcode);
        //printf("%s %X\n",optable[count].mnemonic,optable[count].operandcode);
        count++;
    }
    close(finptr);
```

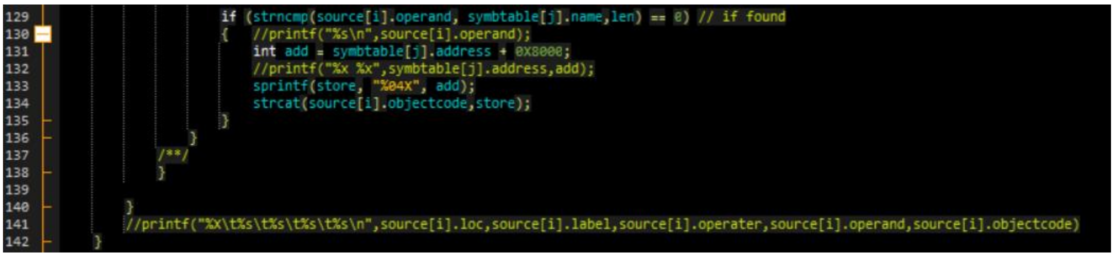
依照不同的狀況，填入不同的值，sprintf 是將字串存入。



```
    for(i = 1; strcmp(source[i].operater, "END") != 0; i++){
        memset(store,'\0',max);
        memset(tmp,'\0',max);
        flag = 0;

        for (j = 0; j < oplen; j++) // search OPTAB for OPCODE
        {
            if (strcmp(source[i].operater, optable[j].mnemonic) == 0) // if found
            {
                sprintf(source[i].objectcode, "%02X", optable[j].operandcode); // store the opcode of operation
                //printf("%s\n",source[i].objectcode);
            }
        } // search OPTAB for OPCODE

        if(strcmp(source[i].operater,"RESW") == 0 || strcmp(source[i].operater,"RESB") == 0){
            //printf("%X\t%s\t%s\t%s\t%s\n",source[i].loc,source[i].label,source[i].operater,source[i].operand,source[i].objectcode);
            continue;
        }
        else if(strcmp(source[i].operater,"RSUB") == 0){
            strcat(source[i].objectcode,"0000");
        }
        else if(strcmp(source[i].operater,"WORD") == 0){
            sprintf(source[i].objectcode,"%06X",atoi(source[i].operand));
        }
```

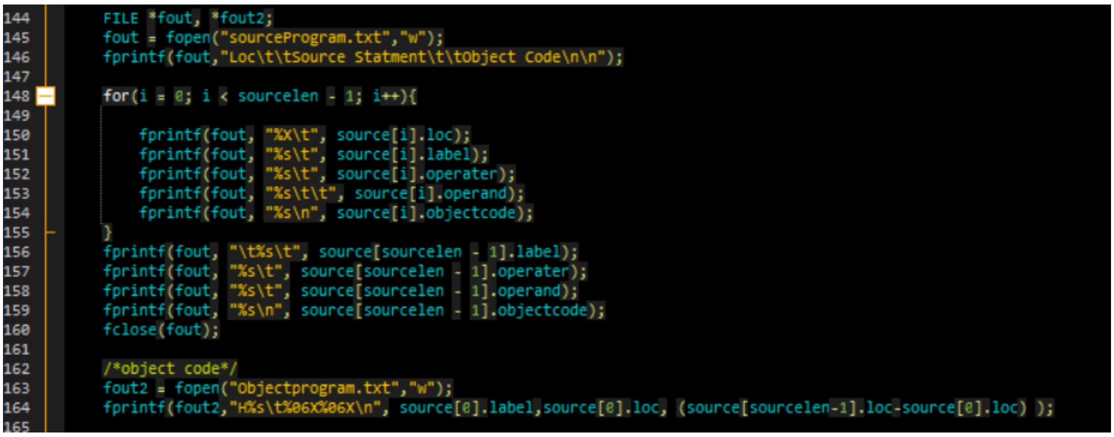


```
        else if(strcmp(source[i].operater,"BYTE") == 0){
            if(source[i].operand[0] == 'C'){
                for (k = 2; source[i].operand[k] != '\''; k++) // copy the char to tmp
                {
                    /* copy the hex number of the char to the string */
                    sprintf(store, "%X", source[i].operand[k]);
                    strcat(source[i].objectcode, store);
                }
            }
            else if(source[i].operand[0] == 'X'){    //case X
                for (k = 2; source[i].operand[k] != '\''; k++){
                    store[index++] = source[i].operand[k];
                }
                strcat(source[i].objectcode, store);
                index = 0;
            }
        }
        else{    //normal case
            for(j = 0; j < symcount; j++){
                int len = strlen(symbtable[j].name);

                if (strcmp(source[i].operand, symbtable[j].name) == 0) // if found
                {   sprintf(store, "%04X", symbtable[j].address); // translate the hex value into string
                    strcat(source[i].objectcode, store);
                    flag = 1;
                }
            }


            if(flag==0){     //case "BUFFER,X"
                //printf("hi");
                for(j = 0; source[i].operand[j] != ','; j++){
                    tmp[j] = source[i].operand[j];
                    //printf("%c",tmp[j]);
                }

                for(j = 0; j < symcount; j++){  //find symbol
                    int len = strlen(symbtable[j].name);
```

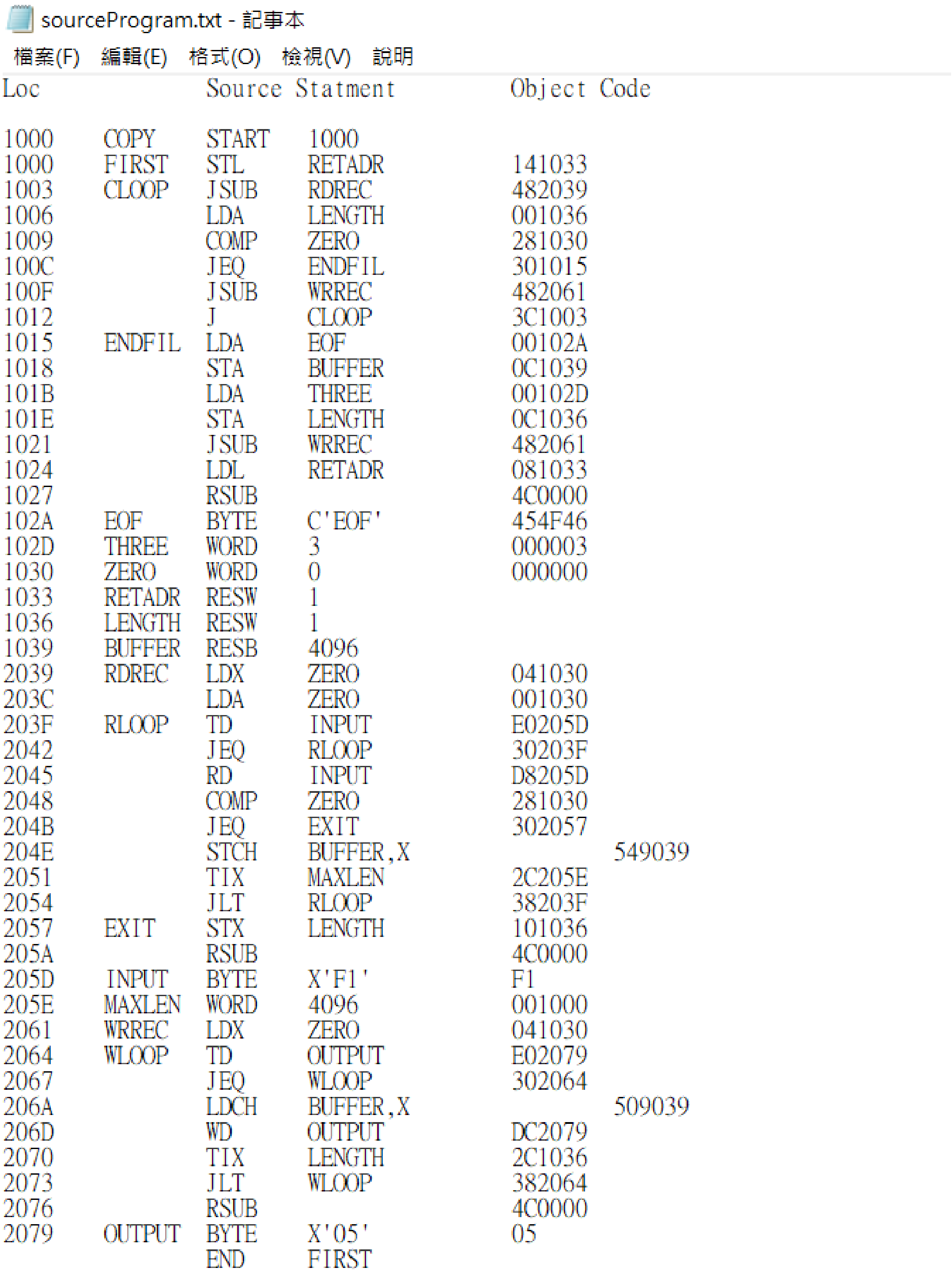


```
                    if (strncmp(source[i].operand, symbtable[j].name,len) == 0) // if found
                    {   //printf("%s\n",source[i].operand);
                        int add = symbtable[j].address + 0X8000;
                        //printf("%x %x",symbtable[j].address,add);
                        sprintf(store, "%04X", add);
                        strcat(source[i].objectcode,store);
                    }
                }
            /**/
            }

        }
        //printf("%X\t%s\t%s\t%s\t%s\n",source[i].loc,source[i].label,source[i].operater,source[i].operand,source[i].objectcode)
    }
```

將計算好的最終 source program 輸出。



```
    FILE *fout, *fout2;
    fout = fopen("sourceProgram.txt","w");
    fprintf(fout,"Loc\t\tSource Statment\t\tObject Code\n\n");

    for(i = 0; i < sourcelen - 1; i++){

        fprintf(fout, "%X\t", source[i].loc);
        fprintf(fout, "%s\t", source[i].label);
        fprintf(fout, "%s\t", source[i].operater);
        fprintf(fout, "%s\t\t", source[i].operand);
        fprintf(fout, "%s\n", source[i].objectcode);
    }
    fprintf(fout, "\t%s\t", source[sourcelen - 1].label);
    fprintf(fout, "%s\t", source[sourcelen - 1].operater);
    fprintf(fout, "%s\t", source[sourcelen - 1].operand);
    fprintf(fout, "%s\n", source[sourcelen - 1].objectcode);
    fclose(fout);

    /*object code*/
    fout2 = fopen("Objectprogram.txt","w");
    fprintf(fout2,"H%s\t%06X%06X\n", source[0].label,source[0].loc, (source[sourcelen-1].loc-source[0].loc) );
```


sourceProgram.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
Loc             Source Statment          Object Code

1000    COPY    START   1000
1000    FIRST   STL     RETADR           141033
1003    CLOOP   JSUB    RDREC            482039
1006            LDA     LENGTH           001036
1009            COMP    ZERO             281030
100C            JEQ     ENDFIL           301015
100F            JSUB    WRREC            482061
1012            J       CLOOP            3C1003
1015    ENDFIL  LDA     EOF              00102A
1018            STA     BUFFER           0C1039
101B            LDA     THREE            00102D
101E            STA     LENGTH           0C1036
1021            JSUB    WRREC            482061
1024            LDL     RETADR           081033
1027            RSUB                     4C0000
102A    EOF     BYTE    C'EOF'           454F46
102D    THREE   WORD    3                000003
1030    ZERO    WORD    0                000000
1033    RETADR  RESW    1
1036    LENGTH  RESW    1
1039    BUFFER  RESB    4096
2039    RDREC   LDX     ZERO             041030
203C            LDA     ZERO             001030
203F    RLOOP   TD      INPUT            E0205D
2042            JEQ     RLOOP            30203F
2045            RD      INPUT            D8205D
2048            COMP    ZERO             281030
204B            JEQ     EXIT             302057
204E            STCH    BUFFER,X                 549039
2051            TIX     MAXLEN           2C205E
2054            JLT     RLOOP            38203F
2057    EXIT    STX     LENGTH           101036
205A            RSUB                     4C0000
205D    INPUT   BYTE    X'F1'            F1
205E    MAXLEN  WORD    4096             001000
2061    WRREC   LDX     ZERO             041030
2064    WLOOP   TD      OUTPUT           E02079
2067            JEQ     WLOOP            302064
206A            LDCH    BUFFER,X                 509039
206D            WD      OUTPUT           DC2079
2070            TIX     LENGTH           2C1036
2073            JLT     WLOOP            382064
2076            RSUB                     4C0000
2079    OUTPUT  BYTE    X'05'            05
                END     FIRST
```

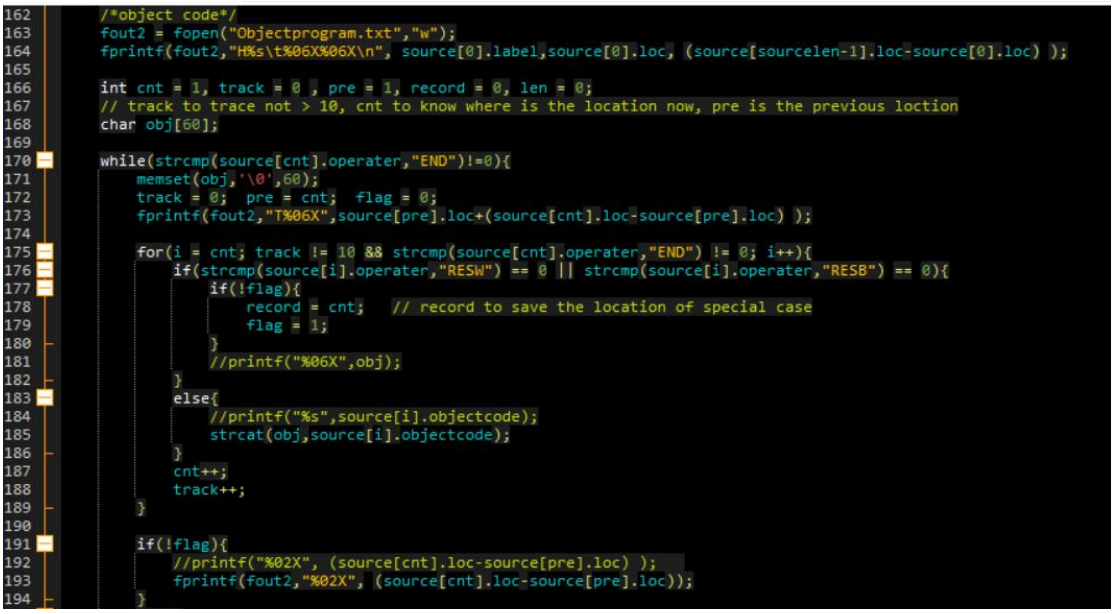
H 一開始由最後的位置減掉最初位置。

T 位置：由現在位置減掉前一個位置。

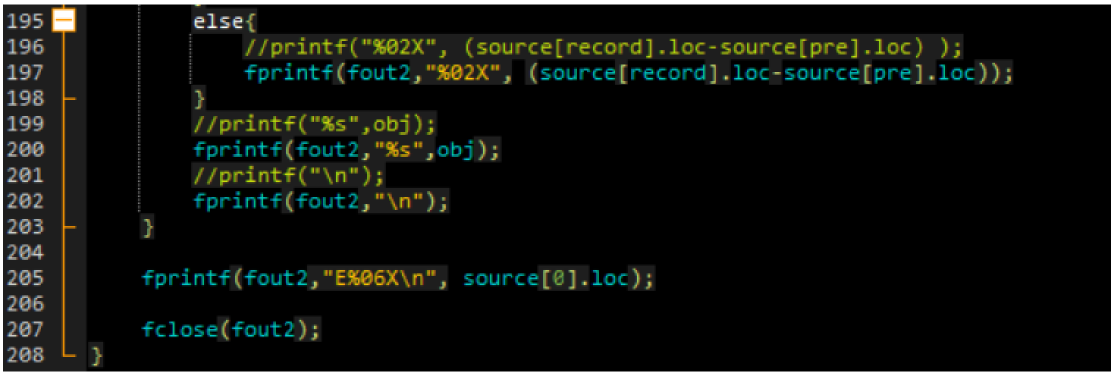
Cnt 紀錄現在已經到哪個位置，track 紀錄現在經過幾個 opcode。

Record 紀錄特殊案例的前一個位置，pre 記錄一般案例的前一位置。

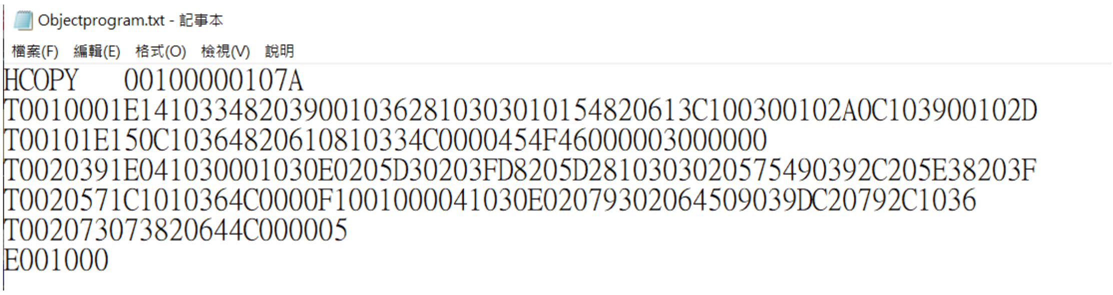
並將最終結果輸出至 Objectprogram.txt 中



```c
    /*object code*/
    fout2 = fopen("Objectprogram.txt","w");
    fprintf(fout2,"H%s\t%06X%06X\n", source[0].label,source[0].loc, (source[sourcelen-1].loc-source[0].loc) );

    int cnt = 1, track = 0 , pre = 1, record = 0, len = 0;
    // track to trace not > 10, cnt to know where is the location now, pre is the previous loction
    char obj[60];

    while(strcmp(source[cnt].operater,"END")!=0){
        memset(obj,'\0',60);
        track = 0;  pre = cnt;  flag = 0;
        fprintf(fout2,"T%06X",source[pre].loc+(source[cnt].loc-source[pre].loc) );

        for(i = cnt; track != 10 && strcmp(source[cnt].operater,"END") != 0; i++){
            if(strcmp(source[i].operater,"RESW") == 0 || strcmp(source[i].operater,"RESB") == 0){
                if(!flag){
                    record = cnt;   // record to save the location of special case
                    flag = 1;
                }
                //printf("%06X",obj);
            }
            else{
                //printf("%s",source[i].objectcode);
                strcat(obj,source[i].objectcode);
            }
            cnt++;
            track++;
        }

        if(!flag){
            //printf("%02X", (source[cnt].loc-source[pre].loc) );
            fprintf(fout2,"%02X", (source[cnt].loc-source[pre].loc));
        }
```



```c
        else{
            //printf("%02X", (source[record].loc-source[pre].loc) );
            fprintf(fout2,"%02X", (source[record].loc-source[pre].loc));
        }
        //printf("%s",obj);
        fprintf(fout2,"%s",obj);
        //printf("\n");
        fprintf(fout2,"\n");
    }

    fprintf(fout2,"E%06X\n", source[0].loc);

    fclose(fout2);
}
```


Objectprogram.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
HCOPY	00100000107A
T0010001E141033482039001036281030301015482061 3C100300102A0C103900102D
T00101E150C1036482061081033 4C0000454F46000003000000
T0020391E041030001030E0205D30203FD8205D28103030205754903 92C205E38203F
T0020571C1010364C0000F1001000041030E02079302064509039DC20792C1036
T002073073820644C000005
E001000
```