Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.
Second Year

CS222: Systems Programming
Spring 2019
Term Project

Pass 1 of sic/xe Assembler project

Contents:

by group (7):

1. **Mahmoud Hamed Sharshar(46)**
2. **Mahmoud Fathy Aboeleneen(47)**
3. **Karim Ibrahim Mostafa(30)**
4. **Abd_elmonem Badre el-sawy (26)**

Requirements specification

- handling source lines that are instructions, storage declaration, comments, and assembler directives.
- handle all storage directives (BYTE, WORD, RESW, and RESB).
- Generate The symbol table.
- A meaningful error message should be printed below the line in which the error occurred.
- The capability of decoding 1,2,3 and 4-byte instructions.
- free-formatted assembly language programs.
- Generation of addresses for all statements in the source code.
- The source program in a format that determines each field of instruction like line number, address, label,opCode, operand, and comment.

Design

The program divided into two parts :

- **Parser.h file**: contains class pass 1 in which methods to perform basic operations of pass 1. The methods are as following:
    - **excutePass1** method:
        - Takes a path of the source file as argument.
        - Read source file line by line and perform some operation to write it in the list file.
    - **constructOpTable** method:
        - build operation table for this assember.
    - **writeIntermediateFile method:**
        - print parts of statement and error in the list file in the proper format.
    - **parseLine** method **:**
        - extract opCode, label, operand, and comment from each statement using regex.
        - Takes line in the source file as an argument.
        - Return parts of the statement.

- **Pass1.cpp file:** contains the main program that creates the object of the pass1 class to assemble specific file.

Main data structures

- **LabelInfo** structure:
  - Hold basic information of any label such as its address and flag to indicate an error.
- **opCodeInfo** structure:
  - Hold information of each operation code such as machine language equivalent code format, and the number of operands required for this opCode.
- **statParts** structure:
  - Represent parts of each statement in source file like a label, opcode, operand,comment and commentOnly that is a flag to indicate this is comment only or not.
- Map data structure(built in): used to
  - Store operation code as key and its information as value (opCodeTable).
  - Store label as key and its information as value(symbol table).

Algorithms description

- Steps of the pass1 algorithm (**excutePass1** method):
  - open source file for reading statement
  - Read first line and loop until hit first instruction.
  - If the first statement is (start statement), specify starting address to the operand.
  - Read the next instruction.
  - Look up the label in the symbol table and if exist, show an error message and if not, insert it into the symbol table.
  - Look up operation code in the operation table.
  - If exist check format instruction and addressing mode.
  - If not, compare opcode with one of directives.

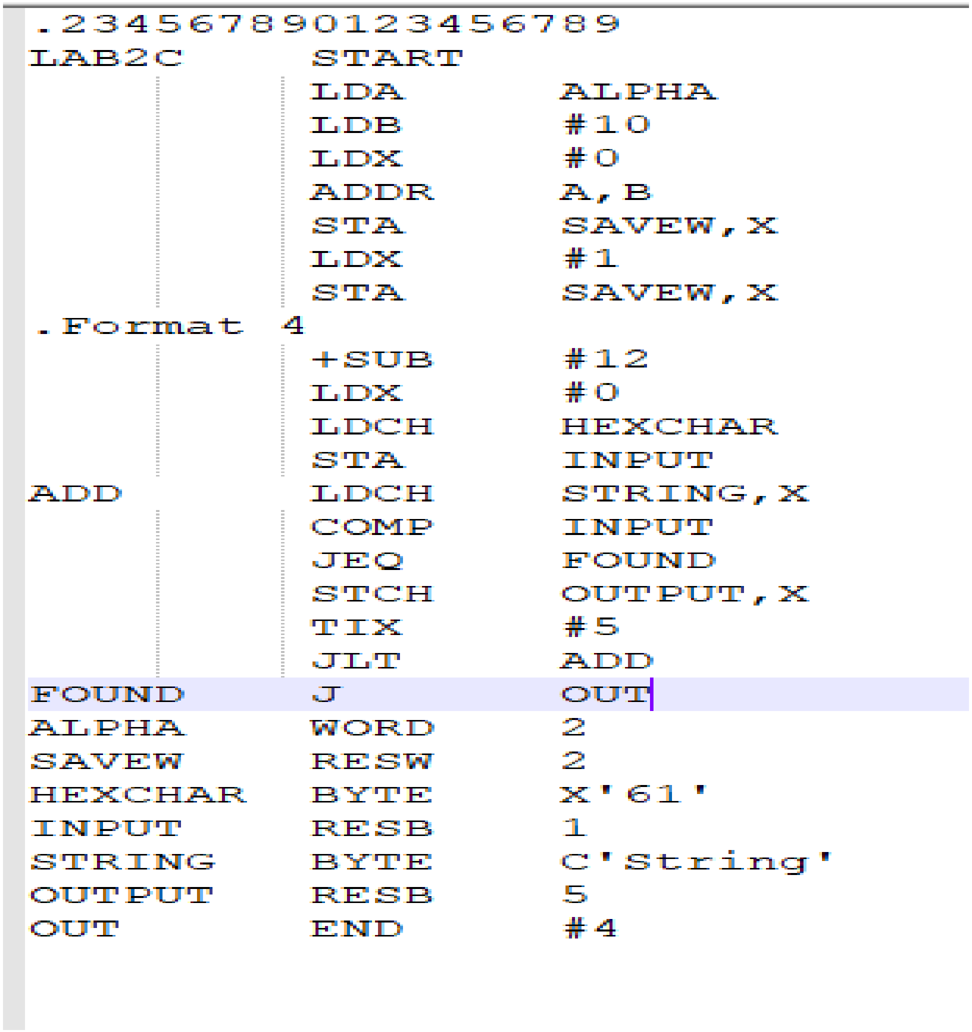
    - Write statements in list file in a proper format.
    - Repeat previous steps until opCode equal to end statement or file ends.

Assumptions

- Any comment must be preceded with the (.) character.
- If the program doesn’t begin with **(start statement)**, it will give a warning not an error and assign the starting address to zero by default.
- If the program doesn’t end with **(end statement)**, it will give a warning not an error.
- For **(EQU statement)**, integers and previously defined symbols are allowed to be operand.
- For **(ORG statement),** previously defined symbols is only allowed to be operand.

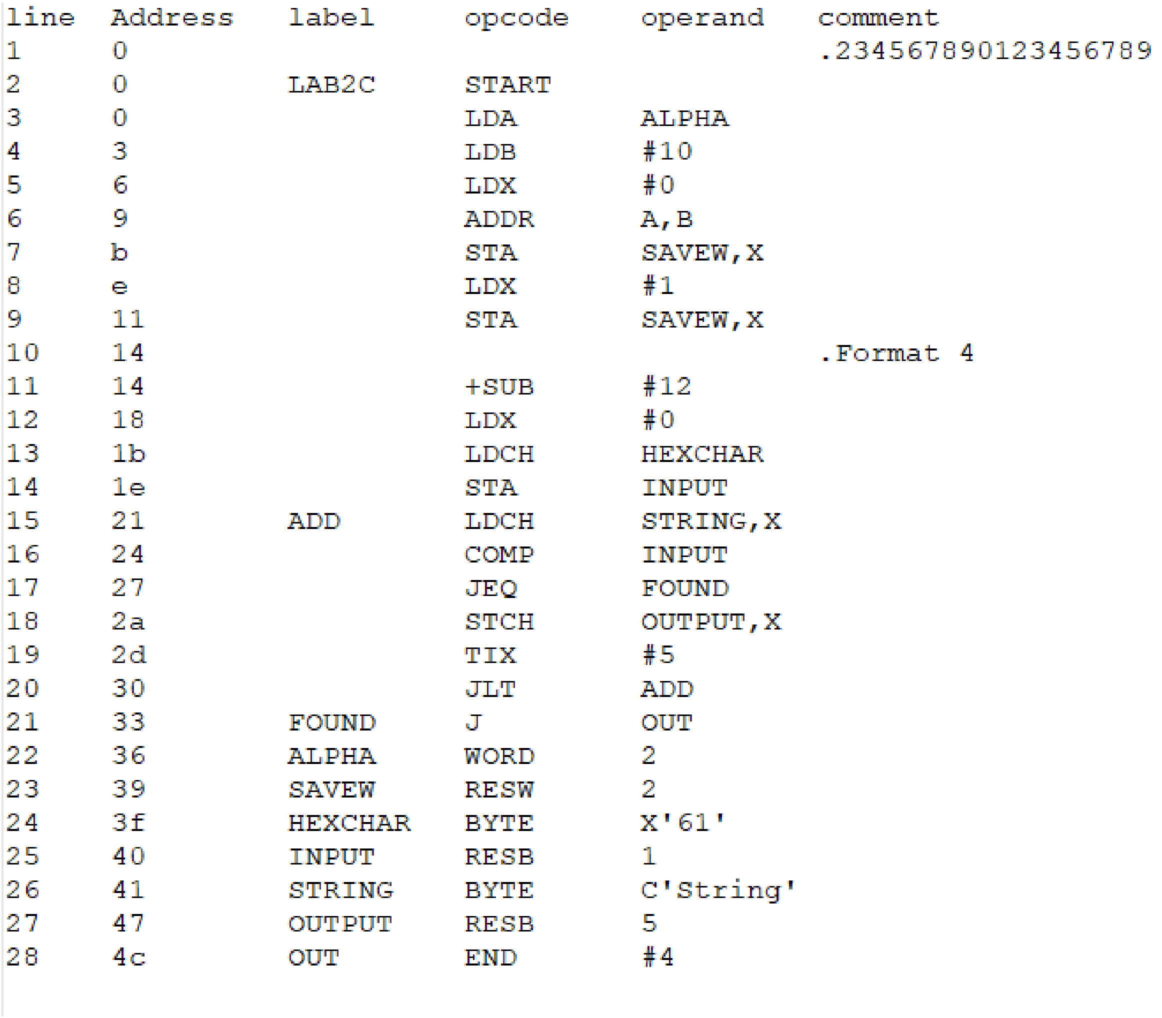
Sample runs

Ex1: Source code:

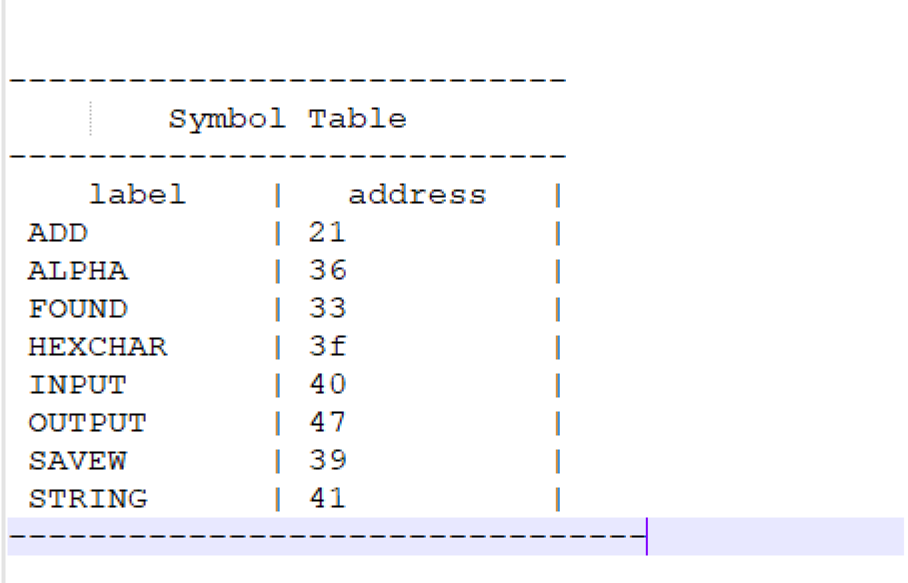


```
.234567890123456789
LAB2C    START
         LDA     ALPHA
         LDB     #10
         LDX     #0
         ADDR    A,B
         STA     SAVEW,X
         LDX     #1
         STA     SAVEW,X
.Format 4
         +SUB    #12
         LDX     #0
         LDCH    HEXCHAR
         STA     INPUT
ADD      LDCH    STRING,X
         COMP    INPUT
         JEQ     FOUND
         STCH    OUTPUT,X
         TIX     #5
         JLT     ADD
FOUND    J       OUT
ALPHA    WORD    2
SAVEW    RESW    2
HEXCHAR  BYTE    X'61'
INPUT    RESB    1
STRING   BYTE    C'String'
OUTPUT   RESB    5
OUT      END     #4
```

List file :

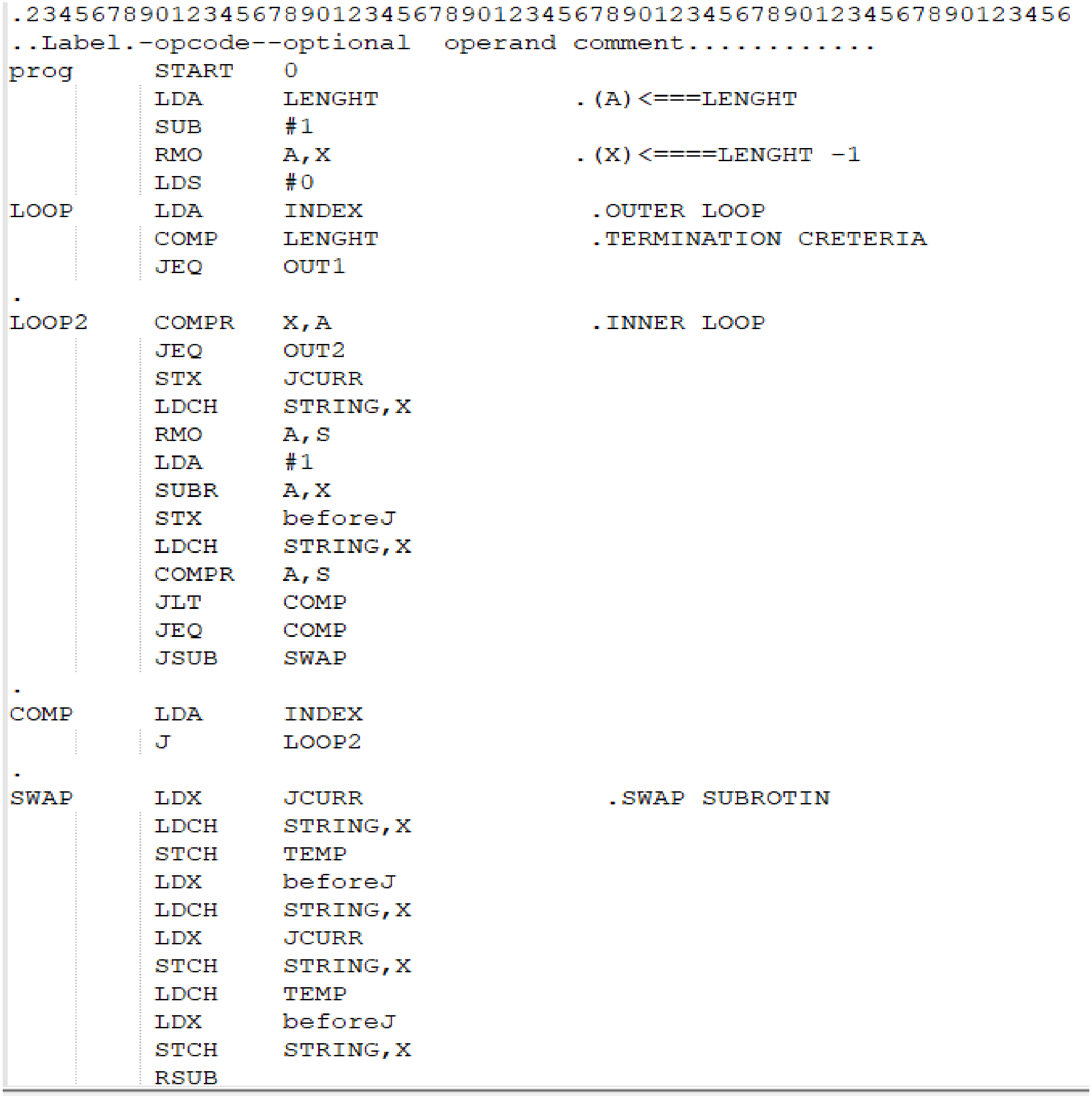


```
line   Address    label      opcode     operand    comment
1      0                                           .23456789012345678 9
2      0          LAB2C      START
3      0                     LDA        ALPHA
4      3                     LDB        #10
5      6                     LDX        #0
6      9                     ADDR       A,B
7      b                     STA        SAVEW,X
8      e                     LDX        #1
9      11                    STA        SAVEW,X
10     14                                          .Format 4
11     14                    +SUB       #12
12     18                    LDX        #0
13     1b                    LDCH       HEXCHAR
14     1e                    STA        INPUT
15     21         ADD        LDCH       STRING,X
16     24                    COMP       INPUT
17     27                    JEQ        FOUND
18     2a                    STCH       OUTPUT,X
19     2d                    TIX        #5
20     30                    JLT        ADD
21     33         FOUND      J          OUT
22     36         ALPHA      WORD       2
23     39         SAVEW      RESW       2
24     3f         HEXCHAR    BYTE       X'61'
25     40         INPUT      RESB       1
26     41         STRING     BYTE       C'String'
27     47         OUTPUT     RESB       5
28     4c         OUT        END        #4
```

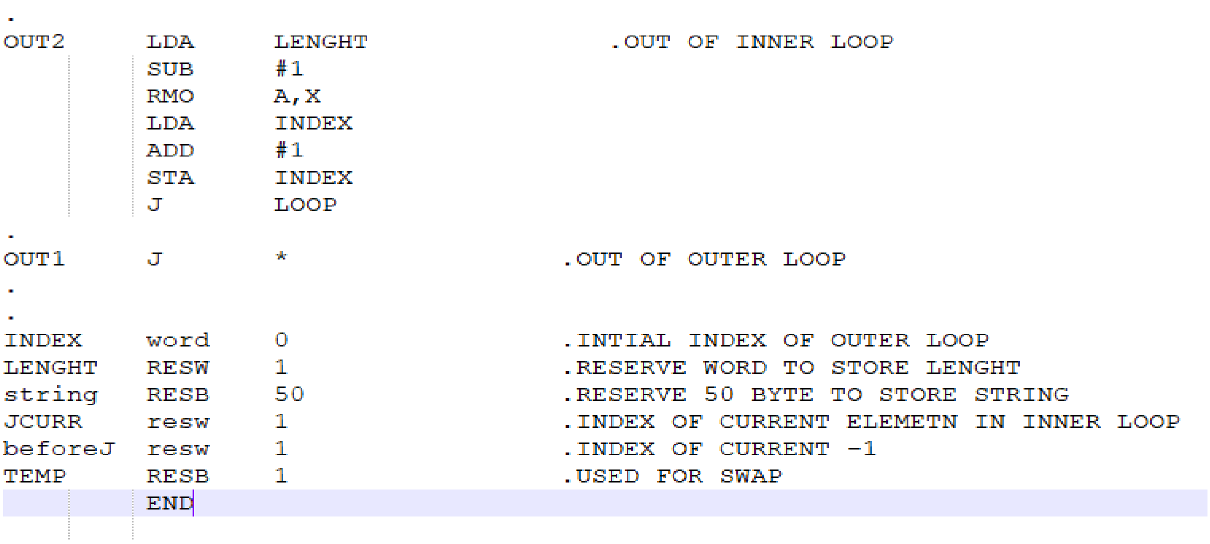


```
------------------------------
        Symbol Table
------------------------------
    label     |    address   |
 ADD          | 21           |
 ALPHA        | 36           |
 FOUND        | 33           |
 HEXCHAR      | 3f           |
 INPUT        | 40           |
 OUTPUT       | 47           |
 SAVEW        | 39           |
 STRING       | 41           |
-----------------------------------
```

Ex 2: Source file

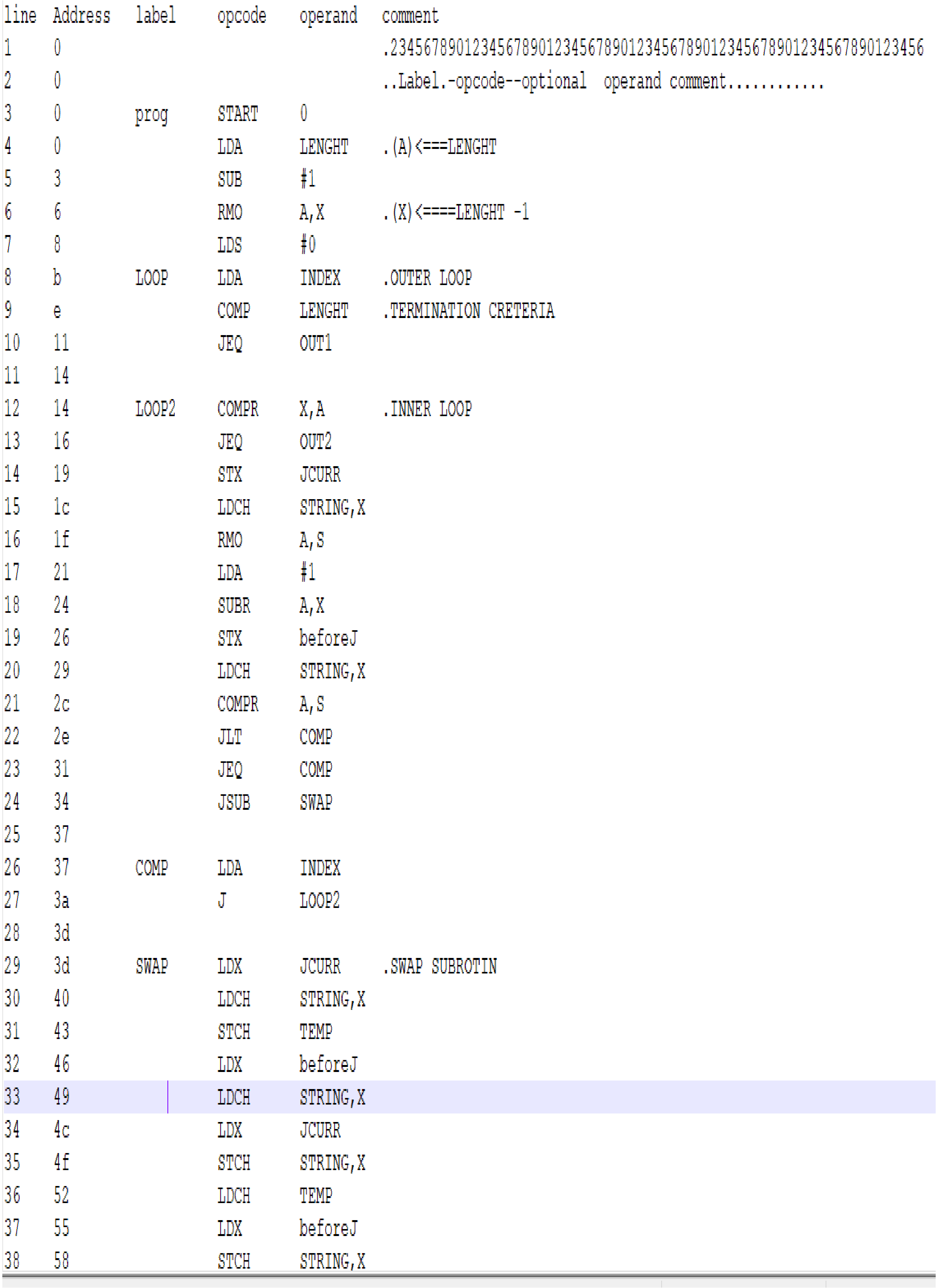


```
.23456789012345678901234567890123456789012345678901234567890123456
..Label.-opcode--optional  operand comment...........
prog     START    0
         LDA      LENGHT                .(A)<===LENGHT
         SUB      #1
         RMO      A,X                   .(X)<====LENGHT -1
         LDS      #0
LOOP     LDA      INDEX                  .OUTER LOOP
         COMP     LENGHT                 .TERMINATION CRETERIA
         JEQ      OUT1
.
LOOP2    COMPR    X,A                    .INNER LOOP
         JEQ      OUT2
         STX      JCURR
         LDCH     STRING,X
         RMO      A,S
         LDA      #1
         SUBR     A,X
         STX      beforeJ
         LDCH     STRING,X
         COMPR    A,S
         JLT      COMP
         JEQ      COMP
         JSUB     SWAP
.
COMP     LDA      INDEX
         J        LOOP2
.
SWAP     LDX      JCURR                   .SWAP SUBROTIN
         LDCH     STRING,X
         STCH     TEMP
         LDX      beforeJ
         LDCH     STRING,X
         LDX      JCURR
         STCH     STRING,X
         LDCH     TEMP
         LDX      beforeJ
         STCH     STRING,X
         RSUB
```

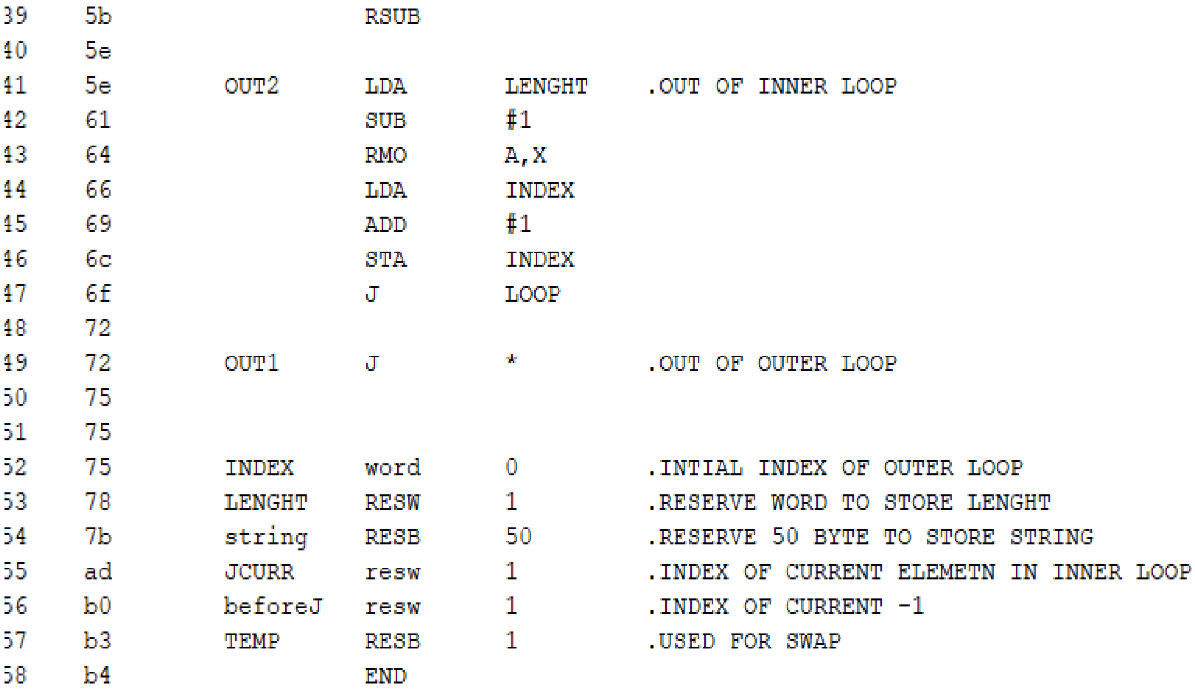


```
.
OUT2     LDA      LENGHT                  .OUT OF INNER LOOP
         SUB      #1
         RMO      A,X
         LDA      INDEX
         ADD      #1
         STA      INDEX
         J        LOOP
.
OUT1     J        *                   .OUT OF OUTER LOOP
.
.
INDEX    word     0                   .INTIAL INDEX OF OUTER LOOP
LENGHT   RESW     1                   .RESERVE WORD TO STORE LENGHT
string   RESB     50                  .RESERVE 50 BYTE TO STORE STRING
JCURR    resw     1                   .INDEX OF CURRENT ELEMETN IN INNER LOOP
beforeJ  resw     1                   .INDEX OF CURRENT -1
TEMP     RESB     1                   .USED FOR SWAP
         END
```

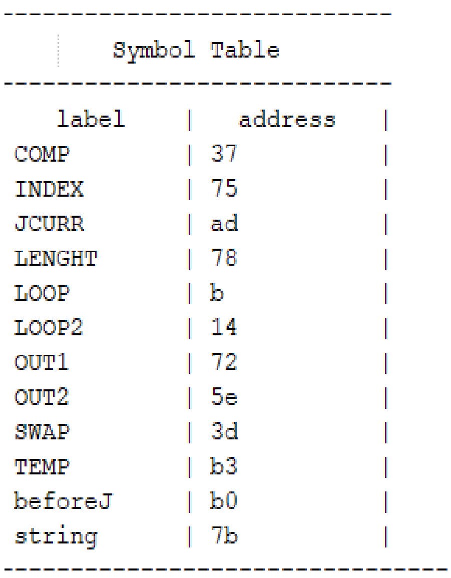
List file:



```
line Address  label     opcode    operand   comment
1    0                                      .234567890123456789012345678901234567890123456789012345678901234567890123456
2    0                                      ..Label.-opcode--optional  operand comment...........
3    0        prog      START     0
4    0                  LDA       LENGHT    .(A)<===LENGHT
5    3                  SUB       #1
6    6                  RMO       A,X       .(X)<====LENGHT -1
7    8                  LDS       #0
8    b        LOOP      LDA       INDEX     .OUTER LOOP
9    e                  COMP      LENGHT    .TERMINATION CRETERIA
10   11                 JEQ       OUT1
11   14
12   14       LOOP2     COMPR     X,A       .INNER LOOP
13   16                 JEQ       OUT2
14   19                 STX       JCURR
15   1c                 LDCH      STRING,X
16   1f                 RMO       A,S
17   21                 LDA       #1
18   24                 SUBR      A,X
19   26                 STX       beforeJ
20   29                 LDCH      STRING,X
21   2c                 COMPR     A,S
22   2e                 JLT       COMP
23   31                 JEQ       COMP
24   34                 JSUB      SWAP
25   37
26   37       COMP      LDA       INDEX
27   3a                 J         LOOP2
28   3d
29   3d       SWAP      LDX       JCURR     .SWAP SUBROTIN
30   40                 LDCH      STRING,X
31   43                 STCH      TEMP
32   46                 LDX       beforeJ
33   49                 LDCH      STRING,X
34   4c                 LDX       JCURR
35   4f                 STCH      STRING,X
36   52                 LDCH      TEMP
37   55                 LDX       beforeJ
38   58                 STCH      STRING,X
```



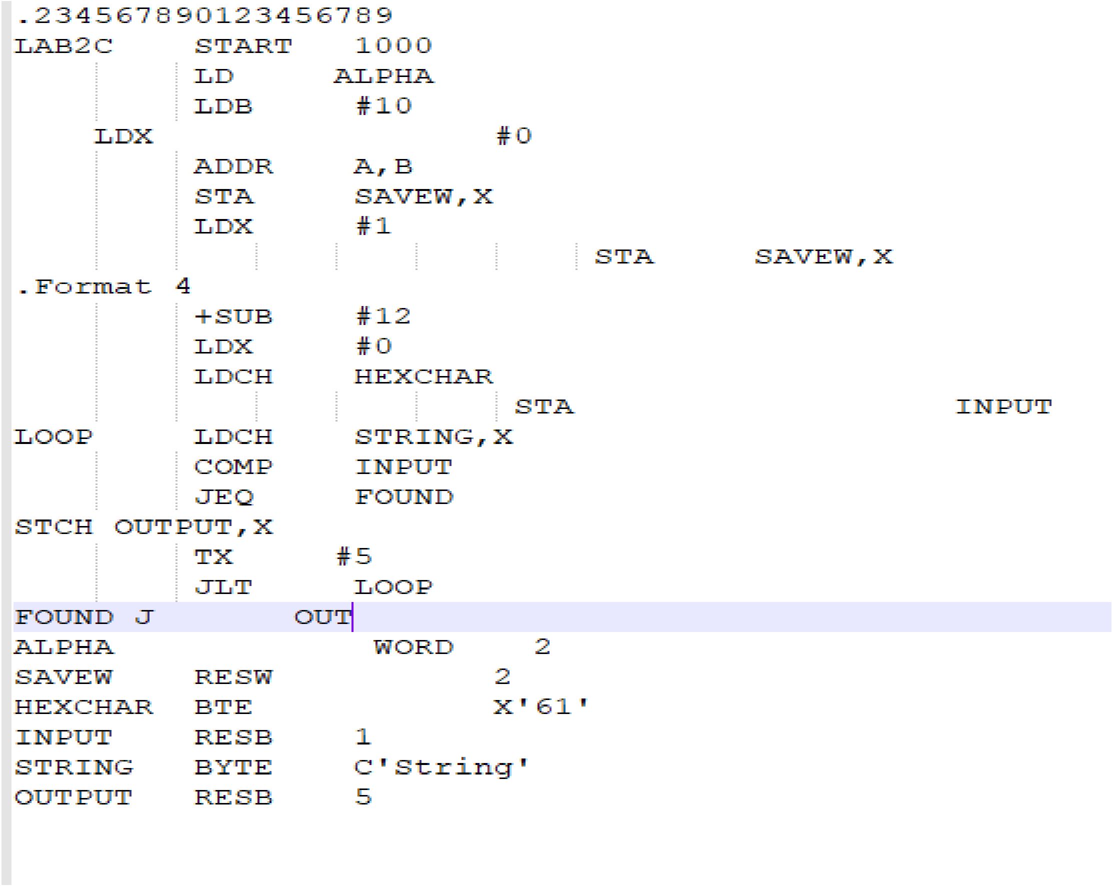
```
39    5b                    RSUB
40    5e
41    5e          OUT2      LDA       LENGHT     .OUT OF INNER LOOP
42    61                    SUB       #1
43    64                    RMO       A,X
44    66                    LDA       INDEX
45    69                    ADD       #1
46    6c                    STA       INDEX
47    6f                    J         LOOP
48    72
49    72          OUT1      J         *          .OUT OF OUTER LOOP
50    75
51    75
52    75          INDEX     word      0          .INTIAL INDEX OF OUTER LOOP
53    78          LENGHT    RESW      1          .RESERVE WORD TO STORE LENGHT
54    7b          string    RESB      50         .RESERVE 50 BYTE TO STORE STRING
55    ad          JCURR     resw      1          .INDEX OF CURRENT ELEMETN IN INNER LOOP
56    b0          beforeJ   resw      1          .INDEX OF CURRENT -1
57    b3          TEMP      RESB      1          .USED FOR SWAP
58    b4                    END
```



```
----------------------------
       Symbol Table
----------------------------
   label      |   address   |
 COMP         | 37          |
 INDEX        | 75          |
 JCURR        | ad          |
 LENGHT       | 78          |
 LOOP         | b           |
 LOOP2        | 14          |
 OUT1         | 72          |
 OUT2         | 5e          |
 SWAP         | 3d          |
 TEMP         | b3          |
 beforeJ      | b0          |
 string       | 7b          |
--------------------------------
```

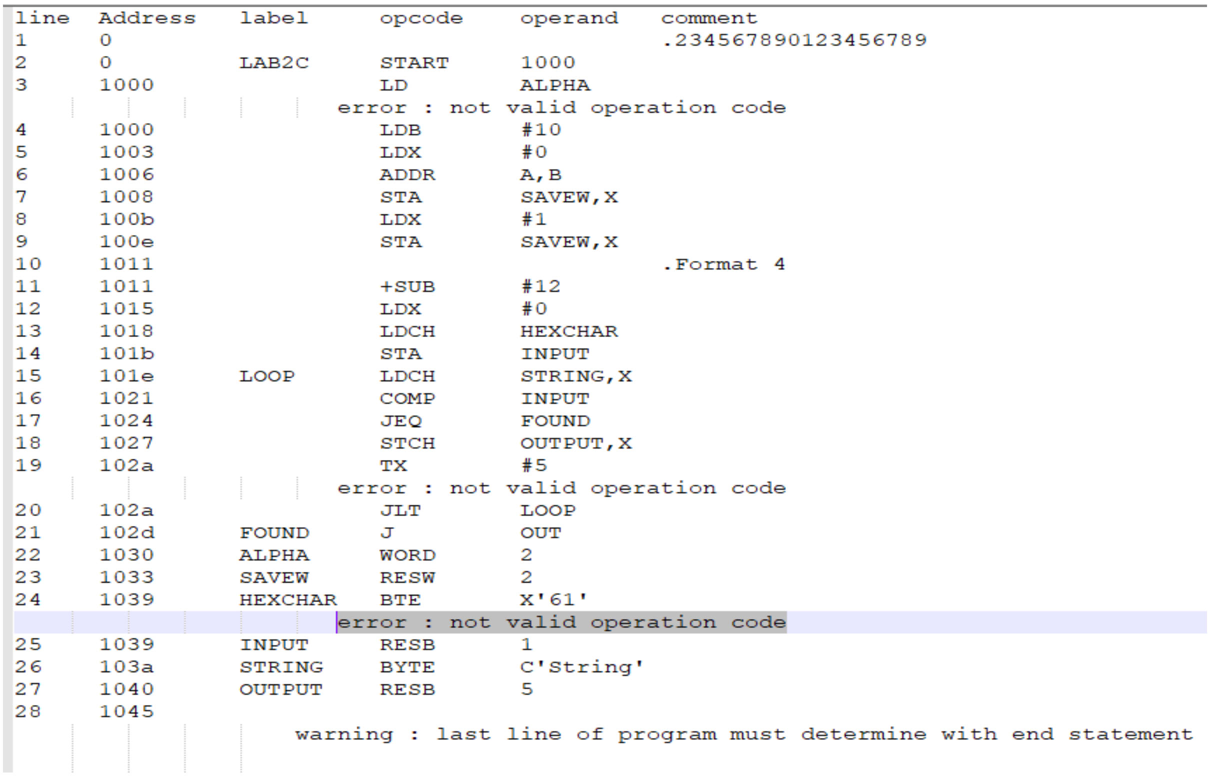
Ex3 : free format and errors indicator:

Source file :



```
.234567890123456789
LAB2C    START   1000
         LD     ALPHA
         LDB     #10
    LDX                 #0
         ADDR    A,B
         STA     SAVEW,X
         LDX     #1
                             STA     SAVEW,X
.Format 4
         +SUB    #12
         LDX     #0
         LDCH    HEXCHAR
                         STA                   INPUT
LOOP     LDCH    STRING,X
         COMP    INPUT
         JEQ     FOUND
STCH OUTPUT,X
         TX     #5
         JLT     LOOP
FOUND J       OUT
ALPHA             WORD    2
SAVEW    RESW           2
HEXCHAR  BTE            X'61'
INPUT    RESB    1
STRING   BYTE    C'String'
OUTPUT   RESB    5
```

List file:



```
line  Address   label     opcode    operand   comment
1     0                                       .234567890123456789
2     0         LAB2C     START     1000
3     1000                LD        ALPHA
                       error : not valid operation code
4     1000                LDB       #10
5     1003                LDX       #0
6     1006                ADDR      A,B
7     1008                STA       SAVEW,X
8     100b                LDX       #1
9     100e                STA       SAVEW,X
10    1011                                    .Format 4
11    1011                +SUB      #12
12    1015                LDX       #0
13    1018                LDCH      HEXCHAR
14    101b                STA       INPUT
15    101e      LOOP      LDCH      STRING,X
16    1021                COMP      INPUT
17    1024                JEQ       FOUND
18    1027                STCH      OUTPUT,X
19    102a                TX        #5
                       error : not valid operation code
20    102a                JLT       LOOP
21    102d      FOUND     J         OUT
22    1030      ALPHA     WORD      2
23    1033      SAVEW     RESW      2
24    1039      HEXCHAR   BTE       X'61'
                       error : not valid operation code
25    1039      INPUT     RESB      1
26    103a      STRING    BYTE      C'String'
27    1040      OUTPUT    RESB      5
28    1045
                    warning : last line of program must determine with end statement
```