Project:	Process the load-constant instruction, **ldc**, and create separate constant tables for character strings, sets, integers, and floating-point numbers. Print the **ldc** instructions together with the instructions printed in project **p04.3** in the trace file having the suffix ".atrc" After the entire P-Code source file has been processed, print the various constants followed by a list of instructions that have been recognized thus far in the assembler listing file having the suffix ".alst"	
Program Files:	**File**	**Description**
	makepasm	File **makepasm** contains instructions for program **pasm**. Instructions are written for the *Unix* utility ***make***. Program **pasm** is contained in file **pasm**.
	mkpasm	File **mkpasm** is a Unix script file that removes old file created in the last creation of executable file **pas** and invokes file **makepasm** to create a new executable file **pasm**. File makepasm is given below. rm *.o rm pasmlex.cpp rm pasm make -f makepasm
Command Line:	Project **p04.4** can be invoked with zero or one program parameters. The first program parameter is the input file name. Sample command lines together with corresponding actions by program **pasm** are shown below. Boldfaced type indicates data entered at the keyboard by the user. $ **pasm** Enter the input file name: **p00.pasm** $ **pasm ldc.pcd**	
Input File:	The input file contains a P-Code Assembler program. The input file name must have the suffix "**.pcd**" Please refer to figure 1 for an example of the format of an input file.	
Output Files:	Two output files are produced. Both output files have the same prefix as the input **source.pcd** and the suffix **.pcd** is replaced by the suffix **.atrc** in the trace file and **.alst** in the assembler listing file. For example, if the input file was named **hello.pcd**, output files would be named **hello.atrc** and **hello.alst**. $ **pasm ldc.pcd** File **ldc.atrc** is produced as shown in Figure 2. File **ldc.alst** is produced as shown in Figure 3.	

```
ldc     b       0
ldc     b       1
ldc     c       'a'
ldc     c       '!'
ldc     c       ''''
ldc     i       54
ldc     i       -17
ldc     r       1.0
ldc     r       2e10
ldc     r       -1.602e-19
ldc     s       'Mary had a little lamb.'
ldc     s       'a'
```

Figure 1. Example input file **ldc.pcd.**

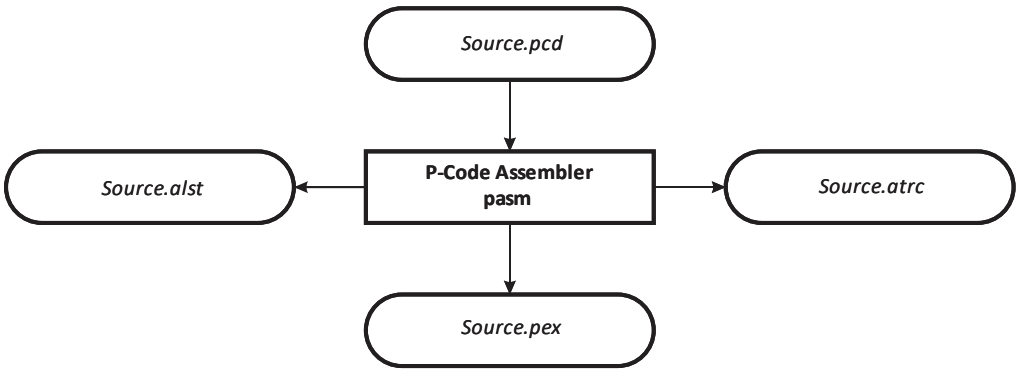

Figure 2. P-Code Assembler Block Diagram

```
Token=(316,LDC_O,ldc)
Token=(329,B_T,b)
#081 type->B
Token=(260,INTLIT,0)
068 class2-operation->ldc type INTLIT
ldc(32)   b(01)     0(0000)
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#002 list->item
Token=(316,LDC_O,ldc)
Token=(329,B_T,b)
#081 type->B
Token=(260,INTLIT,1)
068 class2-operation->ldc type INTLIT
ldc(32)   b(01)     1(0001)
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item
```

Figure 3. File ldc.atrc

```
Token=(316,LDC_O,ldc)
Token=(330,C_T,c)
#082 type->C
Token=(263,CHRLIT,'a')
070 class2-operation->ldc type CHRLIT

014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(316,LDC_O,ldc)
Token=(330,C_T,c)
#082 type->C
Token=(263,CHRLIT,'!')
070 class2-operation->ldc type CHRLIT

014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(316,LDC_O,ldc)
Token=(330,C_T,c)
#082 type->C
Token=(263,CHRLIT,'''')
070 class2-operation->ldc type CHRLIT

014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(316,LDC_O,ldc)
Token=(331,I_T,i)
#083 type->I
Token=(260,INTLIT,54)
068 class2-operation->ldc type INTLIT
ldc(32)   i(03)     2(0002)
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(316,LDC_O,ldc)
Token=(331,I_T,i)
#083 type->I
Token=(260,INTLIT,-17)
068 class2-operation->ldc type INTLIT
ldc(32)   i(03)     3(0003)
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item
```

Figure 3. File ldc.atrc (continued)

```
Token=(316,LDC_O,ldc)
Token=(332,R_T,r)
#084 type->R
Token=(261,REALIT,1.0)
069 class2-operation->ldc type REALIT
ldc(32)   r(04)     0(0000)
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(316,LDC_O,ldc)
Token=(332,R_T,r)
#084 type->R
Token=(261,REALIT,2e10)
069 class2-operation->ldc type REALIT
ldc(32)   r(04)     1(0001)
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(316,LDC_O,ldc)
Token=(332,R_T,r)
#084 type->R
Token=(261,REALIT,-1.602e-19)
069 class2-operation->ldc type REALIT
ldc(32)   r(04)     2(0002)
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(316,LDC_O,ldc)
Token=(333,S_T,s)
#085 type->S
Token=(262,STRLIT,'Mary had a little lamb.')
071 class2-operation->ldc type STRLIT
STRLIT = Mary had a little lamb.
ndx = 0


014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(316,LDC_O,ldc)
Token=(333,S_T,s)
#085 type->S
Token=(263,CHRLIT,'a')
070 class2-operation->ldc type CHRLIT
```

Figure 3. File ldc.atrc (continued)

```
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

#001 program->list
```

Figure 3. File ldc.atrc (continued)

```
String Constants
Index   Constant

    0 "Mary had a little lamb."

Set Constants
Index Constant

Integer Constants
Index   Constant
    0                    0
    1                    1
    2                   54
    3                  -17

Real Constants
Index   Constant
    0         1.000000e+00
    1         2.000000e+10
    2        -1.602000e-19

P-Code Instruction Array
  PC OP       R1         R2
   0 ldc(32)   b(01)     0(0000)
   1 ldc(32)   b(01)     1(0001)
   2 ldc(32)   c(02)    97(0061)
   3 ldc(32)   c(02)    33(0021)
   4 ldc(32)   c(02)    39(0027)
   5 ldc(32)   i(03)     2(0002)
   6 ldc(32)   i(03)     3(0003)
   7 ldc(32)   r(04)     0(0000)
   8 ldc(32)   r(04)     1(0001)
   9 ldc(32)   r(04)     2(0002)
  10 ldc(32)   s(05)     0(0000)
  11 ldc(32)   s(05)    97(0061)
```

Figure 4. File ldc.alst