

Lab 2 Report

Here are screenshots of my initial run of PHONEADD

```
Menu List Mode Functions Utilities Help
ISPFS Command Shell
Enter TSO commands below:
==> CALL COBOL.LOAD(PHONEADD)

Place cursor on choice and press enter to Retrieve command
=> CALL COBOL.LOAD(PHONEADD)
=> ALLOC DA(PHBOOK) FI(PHONBOOK) MOD REUSE
=> CALL COBOL.LOAD(PHONELST)
=> LISTDS PHONE MEMBERS
=> COPY FROM('KC02791.PHONE') TO('KC03C55.PHONE')
=>
=>
=>
=>
Last Name ?
URSO_

Number ?
904-373-6346
Enter Another ?
N_
```

Here is the output of PHONELST with some additional entries in it

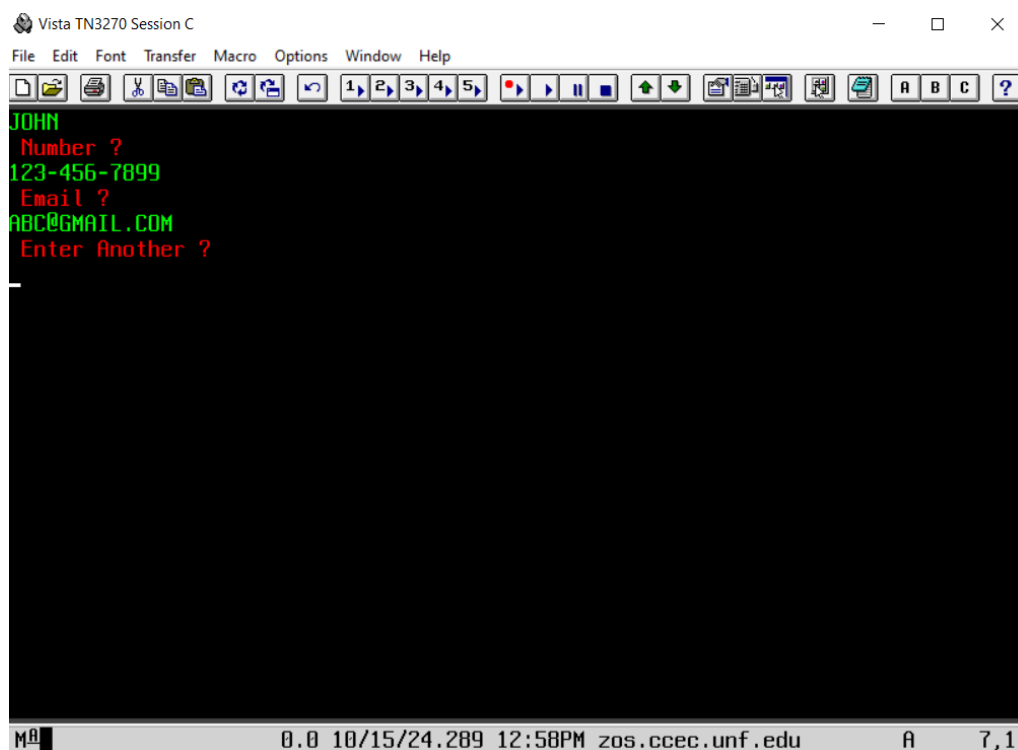
```
Menu List Mode Functions Utilities Help
ISP Command Shell
Enter TSO commands below:
==> CALL COBOL.LOAD(PHONELST)

Place cursor on choice and press enter to Retrieve command

=> CALL COBOL.LOAD(PHONEADD)
=> CALL COBOL.LOAD(PHONELST)
=> ALLOC DA(PHBOOK) FI(PHONBOOK) MOD REUSE
=> LISTDS PHONE MEMBERS
=> COPY FROM('KC02791.PHONE') TO('KC03C55.PHONE')
=>
=>
=>
=>
Last Name:URSO                      First:                      NO:904-373-6346
Last Name:LAW                       First:                      NO:904-703-2657
***
```

My first names were not entered correctly because I was pressing enter to submit at the wrong time but this problem will be fixed as I grow the data file.

Here is my PHONEADD program now that I added email support



```
Vista TN3270 Session C
File Edit Font Transfer Macro Options Window Help
JOHN
Number ?
123-456-7899
Email ?
ABC@GMAIL.COM
Enter Another ?

0.0 10/15/24.289 12:58PM zos.ccec.unf.edu A 7,1
```

And here is the output of PHONELST with a more records added

```
Vista TN3270 Session C
File Edit Font Transfer Macro Options Window Help
[Toolbar]

Last Name:LAW           First:                NO:904-703-2657
      EMAIL:
Last Name:SMITH         First:JOHN            NO:123-456-7890
      EMAIL:
Last Name:URSO          First:ANDREW          NO:904-373-6346
      EMAIL:  N01422082@UNF.EDU
Last Name:URSO          First:ANDREW          NO:904-373-6346
      EMAIL:  ANDREWPURSO@GMAIL.COM

*** _

M0 0.0 10/15/24.289 12:57PM zos.ccec.unf.edu A 18,6
```

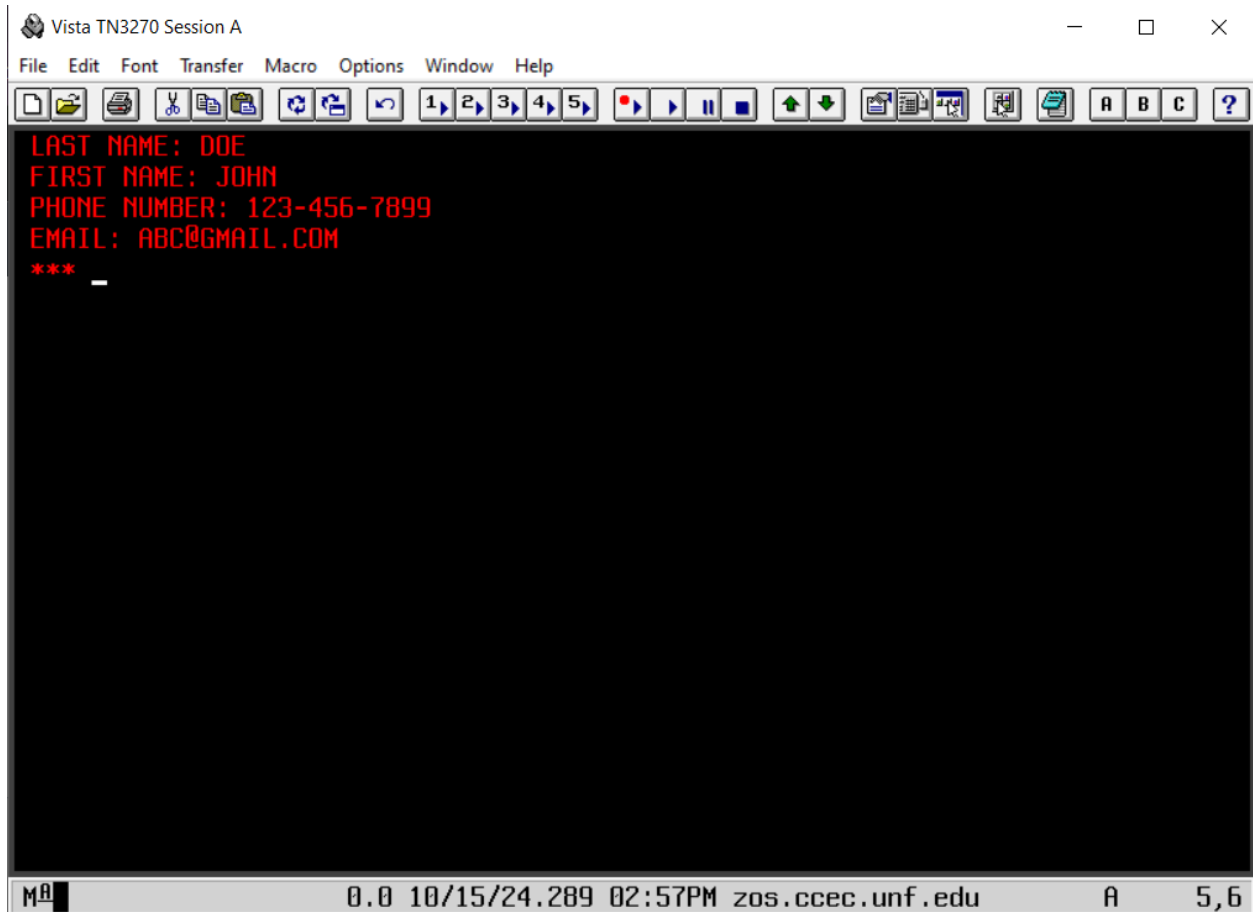
Lastly here is an example of my PHONESRCH program running

```
Vista TN3270 Session A
Menu List Mode Functions Utilities Help
[Toolbar]

ISPF Command Shell
Enter TSO commands below:
==> CALL COBOL.LOAD(PHONESCH)

Place cursor on choice and press enter to Retrieve command
=> CALL COBOL.LOAD(PHONESCH)
=> ALLOC DA(PHBOOK) FI(PHONBOOK) MOD REUSE
=> COPY FROM('KC02791.PHONE') TO('KC03C55.PHONE')
=>
=>
=>
=>
=>
ENTER A LAST NAME TO BE SEARCHED
DOE _

M0 0.2 10/15/24.289 02:56PM zos.ccec.unf.edu A 22,4
```



It searches the record by last name, and then outputs all records containing this last name.

Here is my source code for PHONEADD

```
* 2. Dataset size on our Mainframe default to a size of 80  
* which is required for our record sizes  
* using default DS Allocation in OUR z/OS Environment  
DATA DIVISION.  
FILE SECTION.  
FD  PHONE-FILE RECORDING MODE F  
   LABEL RECORDS ARE STANDARD.  
01  PHONE-RECORD.  
    05  PHONE-LAST-NAME  PIC X(20).  
    05  PHONE-FIRST-NAME PIC X(20).  
    05  PHONE-NUMBER     PIC X(15).  
    05  PHONE-EMAIL      PIC X(25).
```

```

C:\Users\pcaadmin>.vscode>extensions>zowe.vscode-extension-for-zowe-2.18.0>resources>temp>_D_>zosmf_UNF_ZOS>K03C55.PHONE(PHONEADD)
48      *      using default DS Allocation in OUR z/OS Environment
49
50      MA      *
51
52      WORKING-STORAGE SECTION.
53
54      01 PROMPT-1    PIC X(9) VALUE "Last Name".
55      01 PROMPT-2    PIC X(10) VALUE "First Name".
56      01 PROMPT-3    PIC X(6) VALUE "Number".
57      01 PROMPT-4    PIC X(5) VALUE "Email".
58
59
60      01 YES-NO      PIC X.
61      01 ENTRY-OK   PIC X.
62
63
64      PROCEDURE DIVISION.
65      MAIN-LOGIC SECTION.
66      PROGRAM-BEGIN.
67          PERFORM OPENING-PROCEDURE.
68          MOVE "Y" TO YES-NO.
69          PERFORM ADD-RECORDS
70              UNTIL YES-NO = "N".
71          PERFORM CLOSING-PROCEDURE.
72      PROGRAM-DONE.
73          STOP RUN.
74
75      * -----
76      * Open PHONE-FILE for APPENDING. If the file does not exist
77      * it is created (due to our SELECT OPTIONAL)
78      * -----
79      OPENING-PROCEDURE.
80          OPEN EXTEND PHONE-FILE.
81
82      CLOSING-PROCEDURE.
83          CLOSE PHONE-FILE.
84
85      ADD-RECORDS.
86          MOVE "N" TO ENTRY-OK.
87          PERFORM GET-FIELDS
88              UNTIL ENTRY-OK = "Y".
89          PERFORM ADD-THIS-RECORD.
90          PERFORM GO-AGAIN.
91
92      GET-FIELDS.
93          MOVE SPACE TO PHONE-RECORD.
94          DISPLAY PROMPT-1 " ? ".
95          ACCEPT PHONE-LAST-NAME.
96          DISPLAY PROMPT-2 " ? ".
97          ACCEPT PHONE-FIRST-NAME.
98          DISPLAY PROMPT-3 " ? ".
99          ACCEPT PHONE-NUMBER.
100         DISPLAY PROMPT-4 " ? ".
101         ACCEPT PHONE-EMAIL.
102         PERFORM VALIDATE-FIELDS.
103
104     VALIDATE-FIELDS.
105         MOVE "Y" TO ENTRY-OK.
106         IF PHONE-LAST-NAME = SPACE
107             DISPLAY "LAST NAME IS REQUIRED"
108             MOVE "N" TO ENTRY-OK.
109
110
111

```

For this program, I need to add a prompt for email and a variable for email in the PHONE-RECORD. Lastly, I displayed the prompt and accepted the variable in the get-fields procedure.

Here is my source code for PHONLIST

```
10 *****
11 FD  PHONE-FILE RECORDING MODE F
12 LABEL RECORDS ARE STANDARD.
13 @1  PHONE-RECORD.
14      @5  PHONE-LAST-NAME  PIC X(20).
15      @5  PHONE-FIRST-NAME PIC X(20).
16      @5  PHONE-NUMBER     PIC X(15).
17      @5  PHONE-EMAIL      PIC X(25).
18
19 WORKING-STORAGE SECTION.
20 @1  FIELDS-TO-DISPLAY.
21      @5  PROMPT-1          PIC X(10) VALUE "Last Name:".
22      @5  DISPLAY-LAST-NAME PIC X(20).
23      @5  PROMPT-2          PIC X(6) VALUE "First:".
24      @5  DISPLAY-FIRST-NAME PIC X(20).
25      @5  PROMPT-3          PIC X(13) VALUE "MO:".
26      @5  DISPLAY-NUMBER    PIC X(15).
27      @5  PROMPT-4          PIC X(10) VALUE "EMAIL:".
28      @5  DISPLAY-EMAIL     PIC X(25).
29
30 @1  END-OF-FILE  PIC X.
31
32 @1  SCREEN-LINES PIC 99.
33 @1  DUMMYIN      PIC X.
34
35 PROCEDURE DIVISION.
36 MAIN-LOGIC SECTION.
37 PROGRAM-BEGIN.
38
39     PERFORM OPENING-PROCEDURE.
40     MOVE ZEROS TO SCREEN-LINES.
41     MOVE "N" TO END-OF-FILE.
42     PERFORM READ-NEXT-RECORD.
43     PERFORM DISPLAY-RECORDS
44         UNTIL END-OF-FILE = "Y".
45     PERFORM CLOSING-PROCEDURE.
46
47 PROGRAM-DONE.
48     STOP RUN.
49
50 OPENING-PROCEDURE.
51     OPEN INPUT PHONE-FILE.
52
53 CLOSING-PROCEDURE.
54     CLOSE PHONE-FILE.
55
56 DISPLAY-RECORDS.
57     PERFORM DISPLAY-FIELDS.
58     PERFORM READ-NEXT-RECORD.
59
60 DISPLAY-FIELDS.
61     IF SCREEN-LINES = 25
62         PERFORM PRESS-ENTER.
63     MOVE PHONE-LAST-NAME TO DISPLAY-LAST-NAME.
64     MOVE PHONE-FIRST-NAME TO DISPLAY-FIRST-NAME.
65     MOVE PHONE-NUMBER TO DISPLAY-NUMBER.
66     MOVE PHONE-EMAIL TO DISPLAY-EMAIL.
67     DISPLAY FIELDS-TO-DISPLAY.
68
69 * Read the next record from the phone file
70 * If we reach the end, we sent END-OF-FILE
71 * to Y
72 READ-NEXT-RECORD.
73     READ PHONE-FILE NEXT RECORD
74     AT END
75         MOVE "Y" TO END-OF-FILE.
76
77
```

For this program, I added in a prompt for the email and a variable to read in the email from PHONE-RECORD so I could display it.

This is my PHONECMP JCL

```
> Users > pcaadmin > .vscode > extensions > zowe.vscode-extension-for-zowe-2.18.0 > resources > temp > _D_ > zosmf_UNF_ZOS > IF KC03C55.PHONE(PHONECMP)
1  /* -----
2  /* JCL Steps to Compile and Link Phonebook Programs
3  /* For LAB #2
4  /* Author(s): Paul Newton
5  /*           Scott Piersall
6  /* -----
7  //PHONECMP  JOB 1,NOTIFY=&SYSUID
8  //COBRUN   EXEC IGYMCL
9  /* Compile and Link PHONELST Program
10 //COBOL.SYSIN DD DSN=&SYSUID..PHONE(PHONEADD),DISP=SHR
11 //LKED.SYSLMOD DD DSN=&SYSUID..COBOL.LOAD(PHONEADD),DISP=SHR
12 /* Check the return code from the compiler
13 /* we only compile the next program if we get a clean
14 /* compile of PHONEADD
15 // IF RC=0 THEN
16 //COBRUN   EXEC IGYMCL
17 //COBOL.SYSIN DD DSN=&SYSUID..PHONE(PHONELST),DISP=SHR
18 //LKED.SYSLMOD DD DSN=&SYSUID..COBOL.LOAD(PHONELST),DISP=SHR
19 //COBRUN   EXEC IGYMCL
20 //COBOL.SYSIN DD DSN=&SYSUID..PHONE(PHONESCH),DISP=SHR
21 //LKED.SYSLMOD DD DSN=&SYSUID..COBOL.LOAD(PHONESCH),DISP=SHR
22 // ENDIF
23
```

To fix my JCL I just copied the statements that handled phone list, and replaced PHONELST with PHONESRCH, and placed the statements under PHONELST's compilation.

Lastly, here is my code for PHONESRCH

```

*
* Author: Scott Piersall
* -----
IDENTIFICATION DIVISION.
PROGRAM-ID. PHONESRCH.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT PHONE-FILE ASSIGN TO PHONBOOK
    ORGANIZATION IS SEQUENTIAL.

DATA DIVISION.
FILE SECTION.
FD  PHONE-FILE RECORDING MODE F
    LABEL RECORDS ARE STANDARD.
01  PHONE-RECORD.
    05  PHONE-LAST-NAME    PIC X(20).
    05  PHONE-FIRST-NAME   PIC X(20).
    05  PHONE-NUMBER       PIC X(15).
    05  PHONE-EMAIL        PIC X(25).

WORKING-STORAGE SECTION.
01  FIELDS-TO-DISPLAY.
    05  PROMPT-1          PIC X(10) VALUE "Last Name:".
    05  DISPLAY-LAST-NAME PIC X(20).
    05  PROMPT-2          PIC X(6) VALUE "First:".
    05  DISPLAY-FIRST-NAME PIC X(20).
    05  PROMPT-3          PIC X(3) VALUE "No:".
    05  DISPLAY-NUMBER     PIC X(150).
    05  PROMPT-4          PIC X(10) VALUE "Email:".
    05  DISPLAY-EMAIL      PIC X(25).

01  A-STRING              PIC X(50).
01  A-SEARCH              PIC X(15).
01  END-OF-FILE           PIC X.
01  ENTRY-FOUND           PIC X.
01  I PIC 99.
01  J PIC 99.

PROCEDURE DIVISION.
MAIN-LOGIC SECTION.
PROGRAM-BEGIN.
    MOVE 'N' TO END-OF-FILE.
    MOVE 'N' TO ENTRY-FOUND.

    DISPLAY "ENTER A LAST NAME TO BE SEARCHED".
    ACCEPT A-STRING.

    MOVE 0 TO I.
    INSPECT A-STRING TALLYING I FOR CHARACTERS

    IF I > 0
        OPEN INPUT PHONE-FILE
        PERFORM UNTIL END-OF-FILE = 'Y'
            READ PHONE-FILE INTO PHONE-RECORD
            AT END
                MOVE 'Y' TO END-OF-FILE
            NOT AT END
                IF PHONE-LAST-NAME = A-STRING
                    MOVE 'Y' TO ENTRY-FOUND
                    DISPLAY "RECORD FOUND: "
                    PERFORM DISPLAY-FIELDS
                END-IF
        END-READ
    END-PERFORM
    IF ENTRY-FOUND = 'N'
        DISPLAY "NO RECORD FOUND"
    ELSE
        CLOSE PHONE-FILE
        DISPLAY "NO SEARCH STRING ENTERED"
    END-IF.
    STOP RUN.

DISPLAY-FIELDS.
    DISPLAY "LAST NAME: " PHONE-LAST-NAME
    DISPLAY "FIRST NAME: " PHONE-FIRST-NAME
    DISPLAY "PHONE NUMBER: " PHONE-NUMBER
    DISPLAY "EMAIL: " PHONE-EMAIL.

```

In this code I modified your search program to read in a phone record instead of a sentence, and the check to see if the supplied last name matches any last names in the phone record.

Questions/Feedback

1. A PDS is a partitioned data set, this is very useful for mainframe developers as they allow for easy reading and manipulation of stored data. In a business use-case, this would be a great way to store credit card transactions.

2. Two additional options that can be passed in to ALLOC are SHR and OLD. These options determine whether or not exclusive control of the dataset is required.

Operands

ddname

Specifies the data definition name.

dsname

Specifies the fully qualified dataset name.

member

Specifies the member name in the dataset.

SHR

The dataset exists but exclusive control is not required.

OLD

The dataset exists and exclusive control is required.

3. LISTA is short for the clist command, unfortunately I could not figure out how to run the command
4. The unabbreviated form of the ALLOC command is ALLOCATE