**Mainframe Assignment 1**

The objective of Assignment 1 is to get you acquainted with the mainframe interface, as well as with some basic user commands. The instructions below will lead you through exactly what you need to do to successfully complete Assignment 1. Read the instructions carefully!

**Table of Contents**

[1. Logging on to Marist Mainframe using 3270 Emulator 1](#_Toc346529741)

[2. Creating Your Assignment One Data Sets (5 Marks) 1](#_Toc346529742)

[3. Use ISPF facilities to locate data sets 8](#_Toc346529743)

[4. Create a new member in your data set (2 Marks) 13](#_Toc346529744)

[5. Enter three lines of text into your new data set member (3 Marks) 14](#_Toc346529745)

[6. Use the ISPF editor to manipulate text (5 Marks) 17](#_Toc346529746)

[7. Run an executable against your new data set member (5 Marks) 27](#_Toc346529747)

[8. Become an ISPF Edit Guru (15 Marks) 38](#_Toc346529748)

[9. ISPF Edit Macro (15 marks) 41](#_Toc346529749)

[10. Submitting Assignment 1 43](#_Toc346529750)

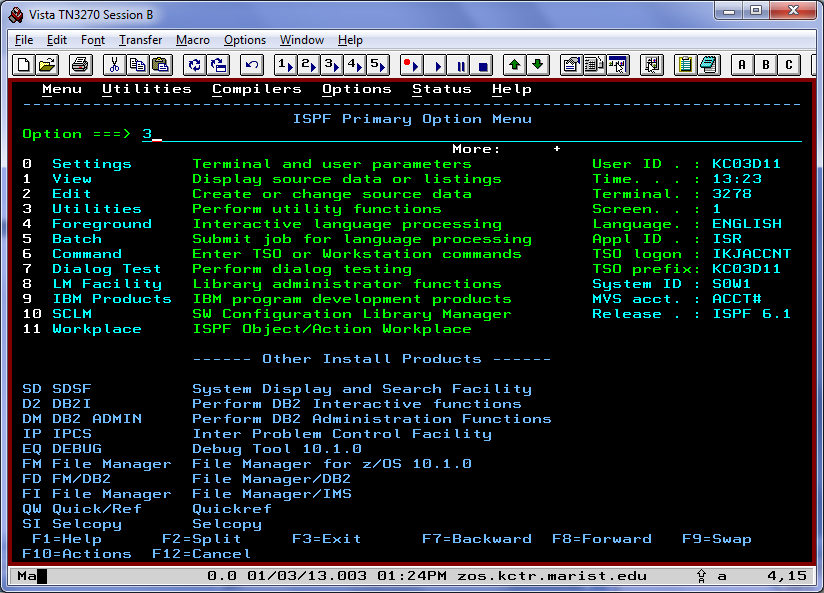
**1. Logging on to Marist Mainframe using 3270 Emulator**

To access a z/OS mainframe you'll need a 3270 terminal emulator. You can use the Vista TN3270 Emulator installed on all Georgian College PCs or see Blackboard for information about 3270 emulators that are available for your operating system.

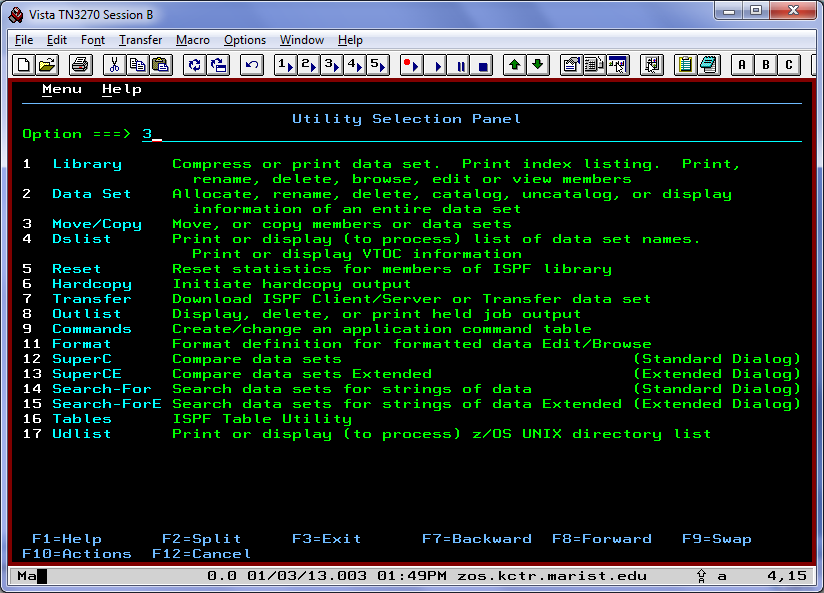
Also see the " How to Logon to the Marist Mainframe" item under the Assignment folder on Blackboard for detailed instructions on how to logon the Marist z/OS system.

**2. Creating Your Assignment One Data Sets (5 Marks)**

We now need to create (and copy) the data sets you will use to complete Assignment 1. The name of these data sets will be '***yourid***.ASSIGN1.WORK' and '***yourid***.ASSIGN1.REXX' where ***yourid*** is your Marist userid. The screen shots below describe how to create and copy '***yourid***.ASSIGN1.WORK'. Follow the same instructions to create and copy '***yourid***.ASSIGN1.REXX' but substitute 'KC02315.ASSIGN1.REXX' for 'KC02315.ASSIGN1.WORK'.

From the ISPF main screen: select option 3 (Utilities), and press Enter.

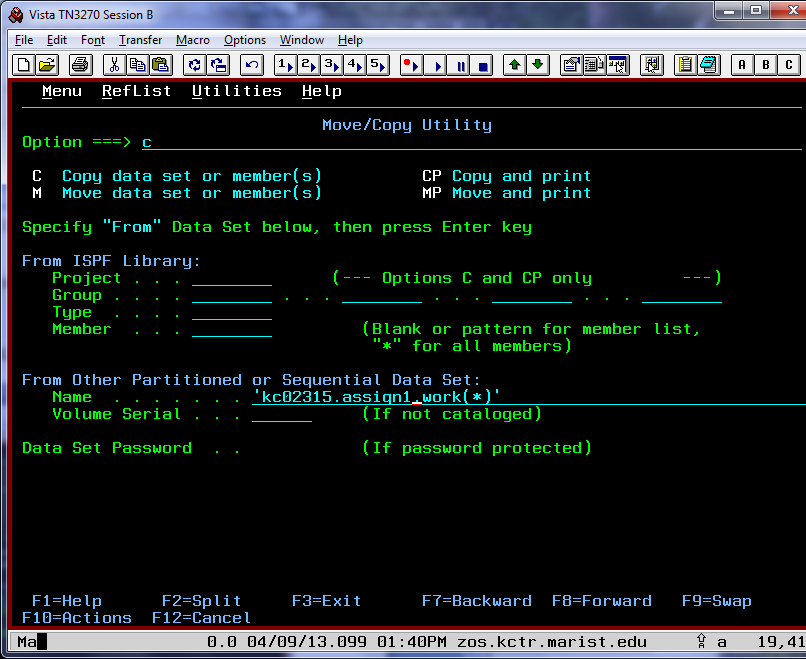
On the next screen: select option 3 (Move/Copy) and press Enter.



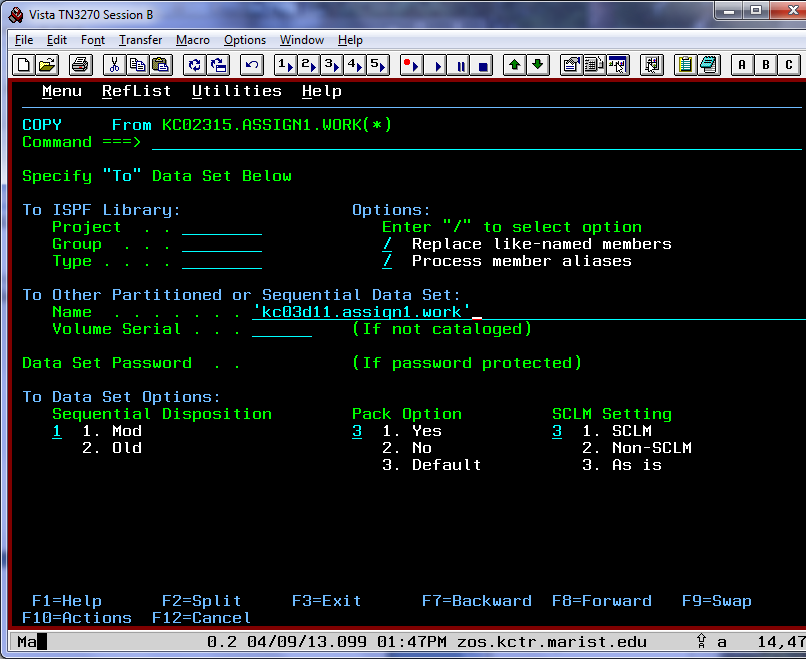
Specify C (Copy data set or member(s) and 'KC02315.ASSIGN1.WORK(\*)' in the

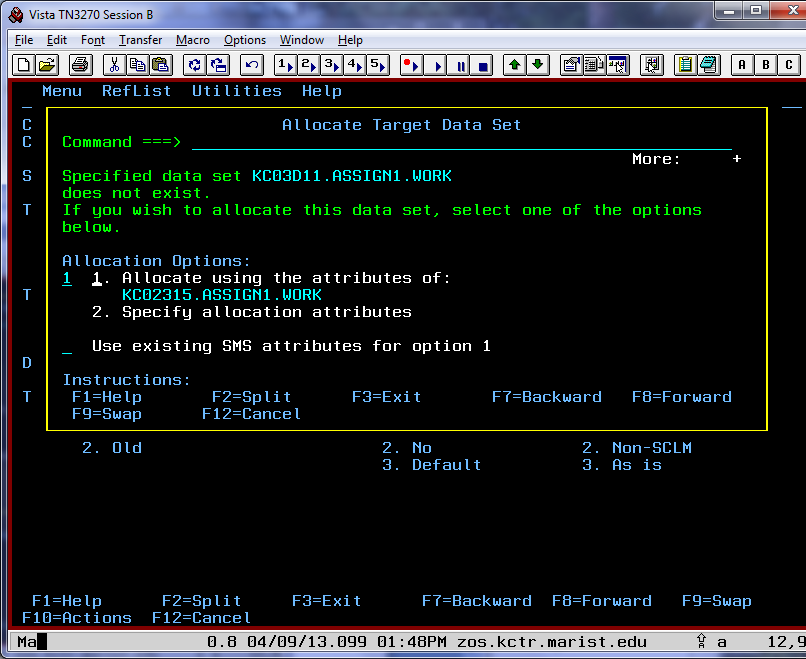
"Name . . . . . ." entry field and press enter.

Note: The asterisk (\*) tells the Copy utility to copy all the members of the data set.

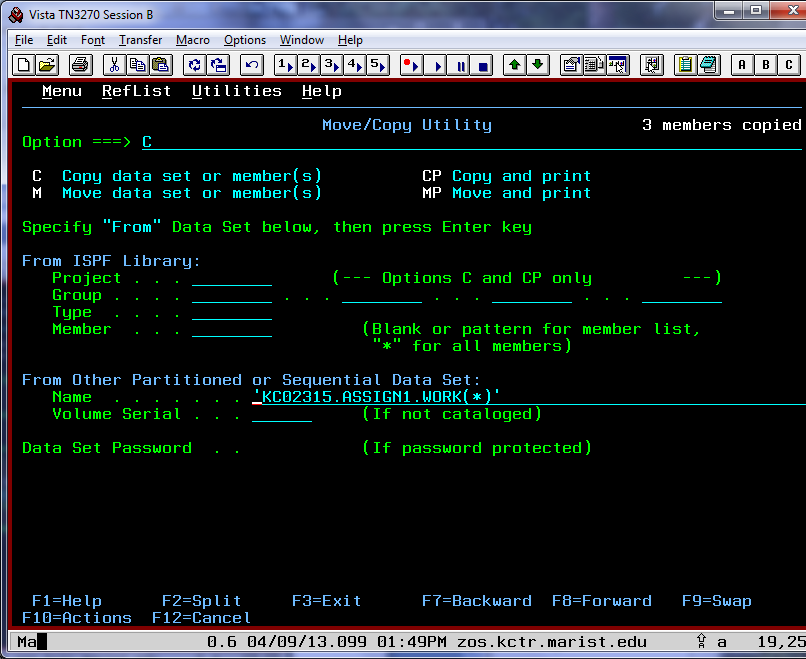


On the next screen specify '***yourid***.ASSIGN1.WORK' in the "Name . . . . . " entry field and press enter.



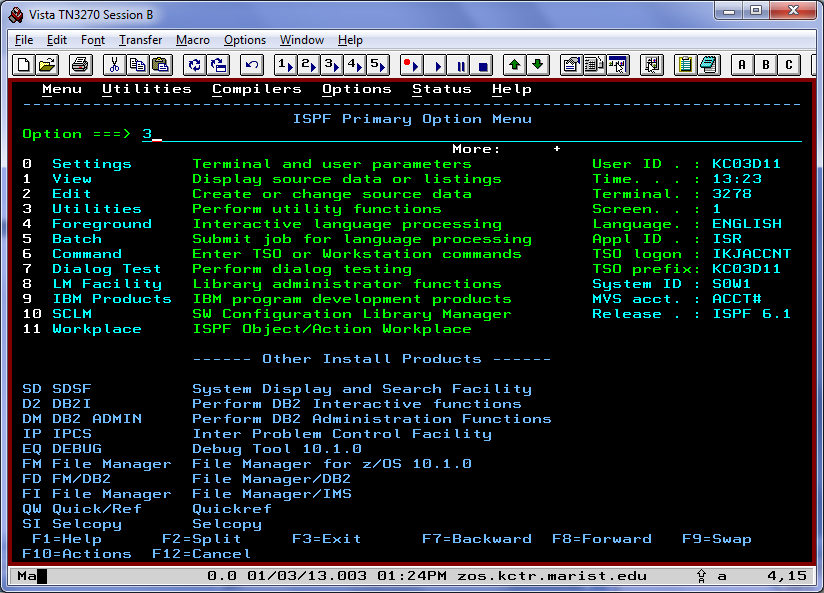
You will then be prompted to allocate (create) a new data set because 'yourid.ASSIGN1.WORK' does not exist yet. type 1 in the Allocation Options field (the cursor should already be there) and press enter. 

Finally you will return to the Move/Copy Utility menu and it will indicate that 3 members were copied. Now repeat this process for the ASSIGN1.REXX data set (it has 2 members).

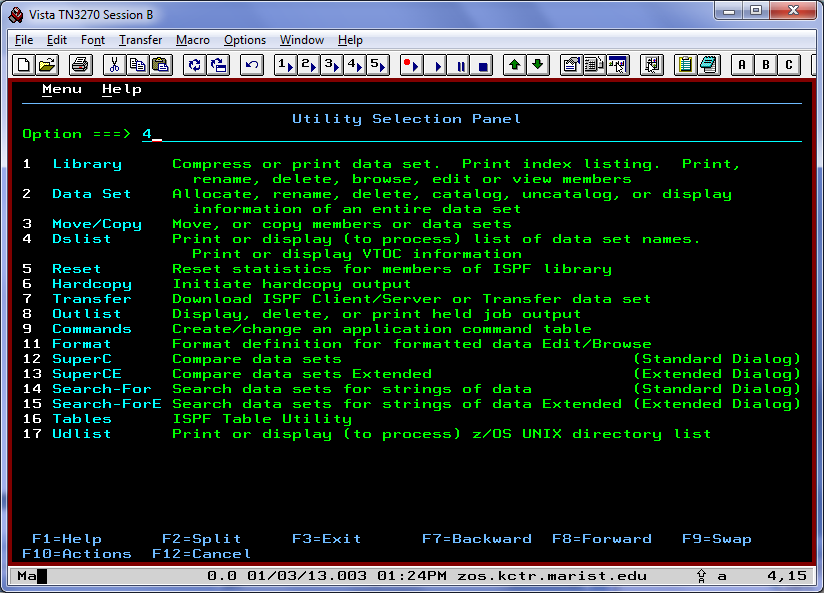


## 3. Use ISPF facilities to locate data sets

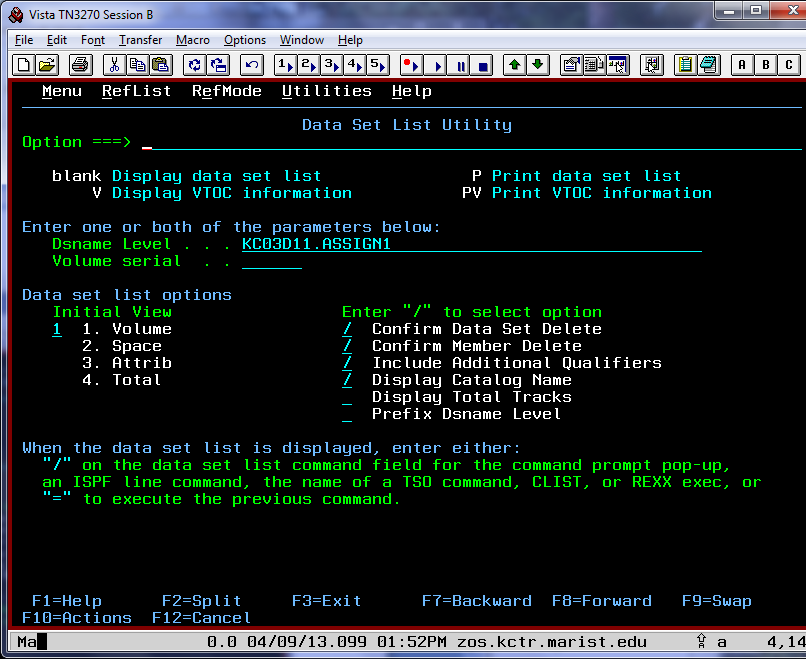
From the ISPF main screen: select option 3 (Utilities), and press Enter.



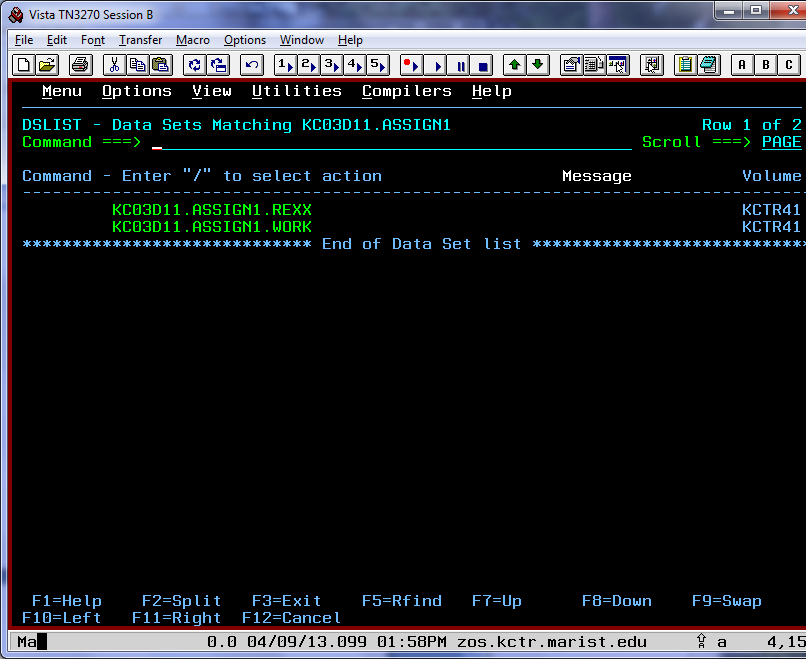
On the next screen: select option 4 (Dslist - short for data set list) and press Enter.



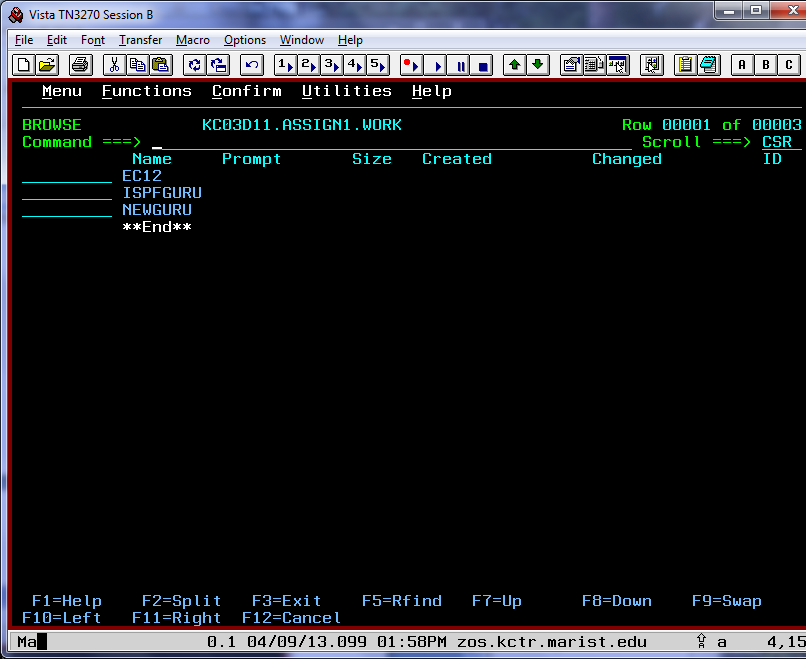
To display the assignment 1 datasets, enter yourid.ASSIGN1 (no quotes) on the line that says **Dsname Level** and press the enter key.



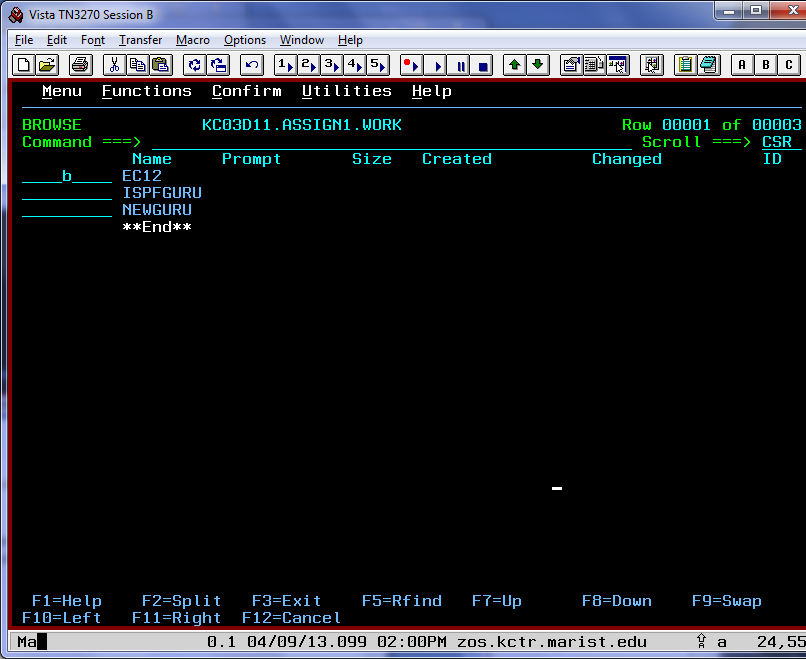
You'll be presented with a list of the two datasets required for assignment 1.



A *partitioned data set* (PDS) is the mainframe equivalent of a folder or a directory. Most of the data sets in the list you're looking at now are actually of the type PDSE (*partitioned data set extended)*, which is an improvement over the type PDS. We will discuss the technical differences between a PDS and a PDSE in class.  
  
A data set contains *members*, which you are probably used to calling *files*. Let's have a look at a member that contains some information about the newest mainframe, the IBM zEnterprise EC12. Tab down until your cursor is beside ***yourid***.ASSIGN1.WORK, then type a **B** (for browse):



and press Enter. You'll see that this data set contains three members, named EC12, ISPFGURU and NEWGURU.   
  
You can also browse members by entering a **B** on the field next to them. Try this on the member named EC12 (if you were going to **E**dit a member, you'd enter it using **E** instead — but we're not doing that just yet):



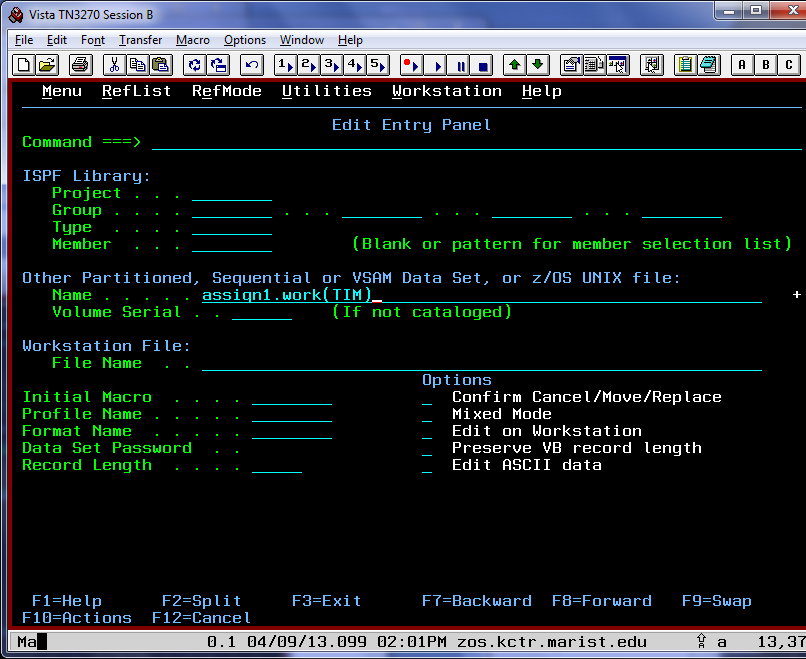
The EC12 member contains some quality information about your favorite refrigerator-sized computer and you've just learned how to navigate into data sets and members.

Now press F3 until you're back at the ISPF Primary Option menu. Let's create a new member for you to check out.

**4. Create a new member in your data set (2 Marks)**

A member (file) can be anything you'd like it to be - flat text, executables, program output - just like the files on your home computer. Let's create a simple flat text member.

From the ISPF Primary Option Menu, select option 2 (Edit). Let's name the new member after the person who's going to create it - you.   
  
Data sets are named with 1-8 character identifiers separated by periods, like this: ***yourid***.ASSIGN1.WORK. When you're creating or editing new members, ISPF automatically assumes that you want the first identifier (called the high level qualifier or HLQ) to be your user ID, unless you put single quotes around the entire data set name.   
  
Tab down until your cursor is on the **Name** field. The data set we're going to add a member to is your ***yourid***.ASSIGN1.WORK data set. Enter **ASSIGN1.WORK** as the data set name and then put the member name in parentheses directly following the data set name. Let's use your first name (or the first eight letters of your first name). For this example, we'll use the name Tim but you should use your own name:

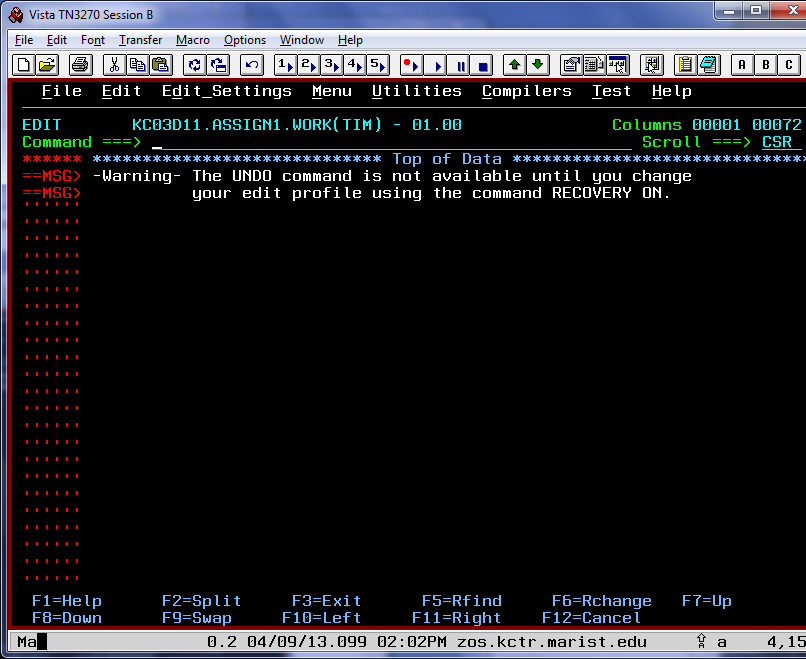


Press Enter to continue. The system creates a new member named *your\_name* in the data set ***yourid***.ASSIGN1.WORK and automatically opens it in an editor session. (In case you were wondering, entering '***yourid***.ASSIGN1.WORK (*your\_name*)' would have worked as well.)

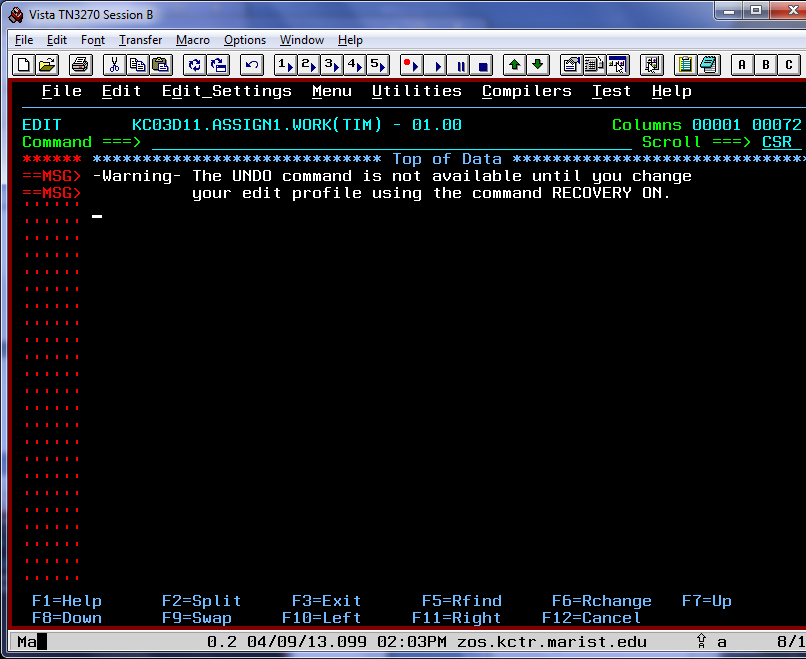
Now you're ready to learn the ISPF editor!

**5. Enter three lines of text into your new data set member (3 Marks)**

You are now in the ISPF editor, looking at a blank member:



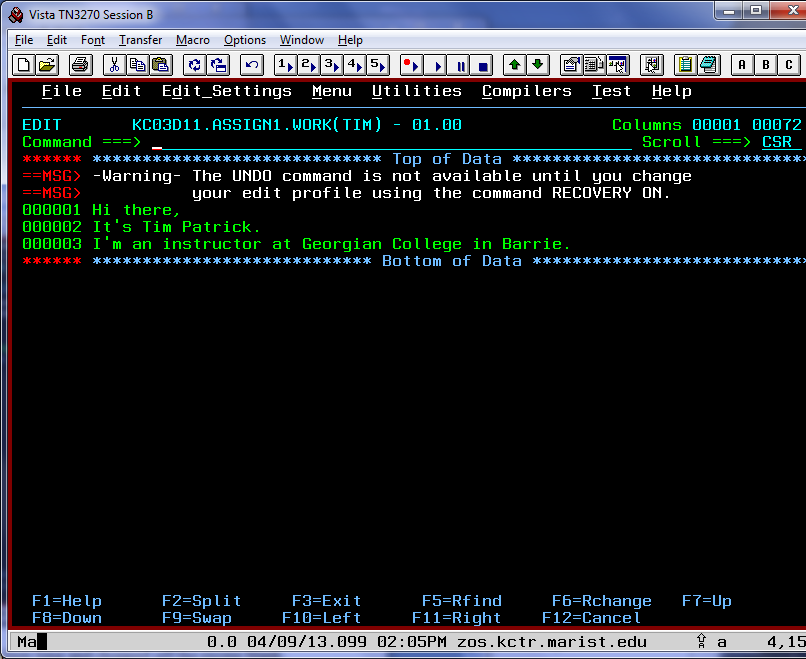
Tab your cursor to the first line of blank space (two lines under the first "-" in "-Warning-"):



Now type in three lines worth of whatever you'd like to say. The three lines of text can be whatever you want.

The important thing is that you type three lines of text here, using the tab key to move to the second line when you're done with the first, and so on. When you are finished entering the text, press Enter. (If at any point you mess up and would like a clean slate, enter **CAN** (short for cancel) on the command line. Your work will not be saved, and you will return to the previous screen. Press Enter on that screen, and you will have a fresh data set member again. You can also enter **HELP** at any time on the command line for more information about the ISPF editor.)

Once you're happy with your text, press Enter. Your screen should now have three lines of text:

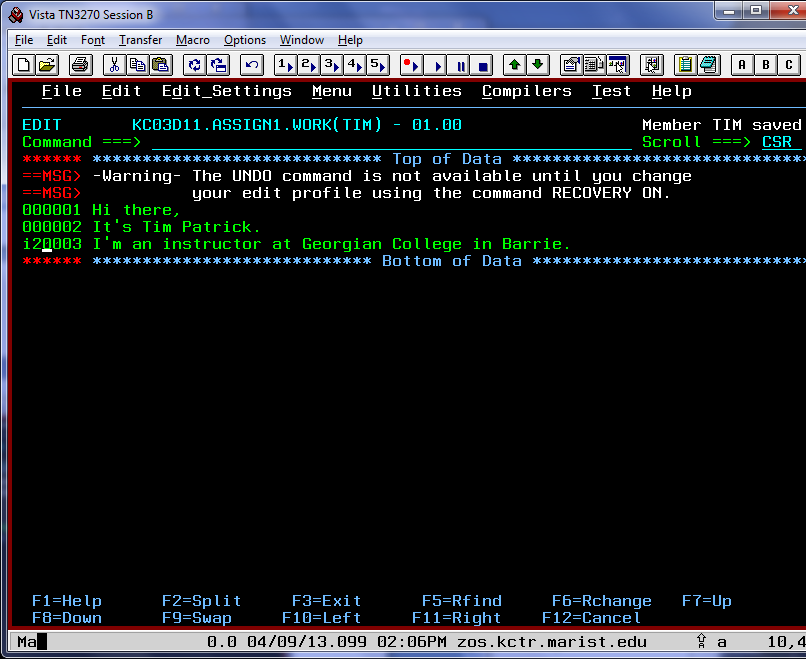


The ISPF editor has made the member exactly three lines long and chopped off the excess blank lines. To save your work, type SAVE on the command line and press Enter.

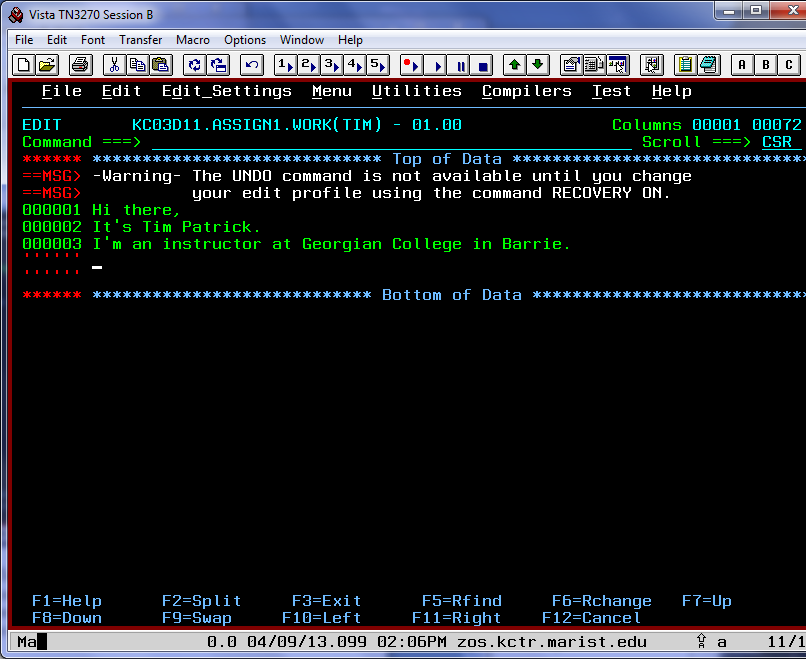
**6. Use the ISPF editor to manipulate text (5 Marks)**

Now let's practice some simple ISPF editor commands - the basic tools for editing and modifying members.

To insert more blank lines, use the **I** command (for insert). Line commands go in the numbered field beside the lines of text, right on top of the existing line numbers. Let's insert two lines by putting the **I2** command on the third line (go ahead and type right over the line numbers that are there):

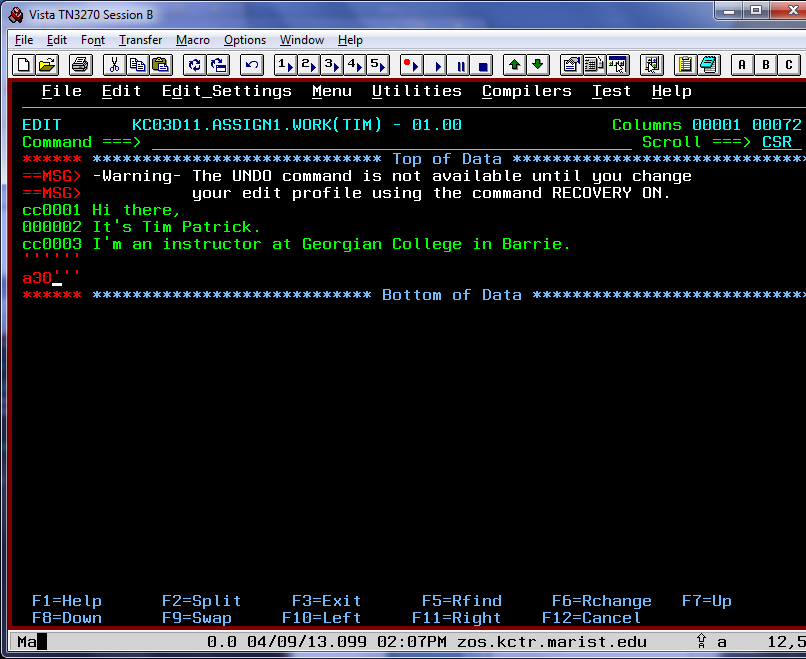


Now press Enter. Two blank lines have been created:

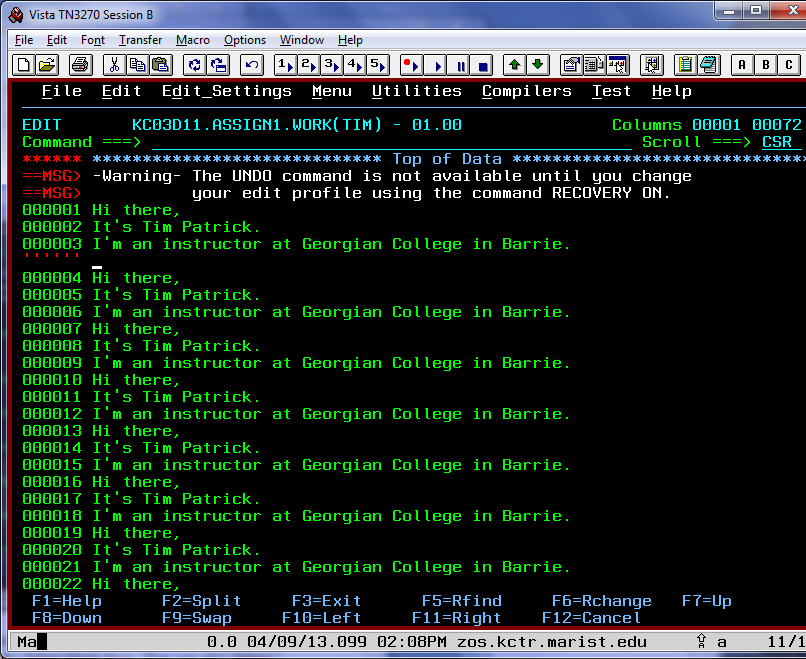


Let's now copy your brilliant text thirty times over. To do this, put a **CC** on the first line you want to copy, and a **CC** on the last line you want to copy (the first and third lines of your text, respectively). To tell the editor to paste the copied section thirty times, go to the last line, and put an **A30** in the last line's command field (the A stands for "paste After this line." A **B** would paste it before the line.)

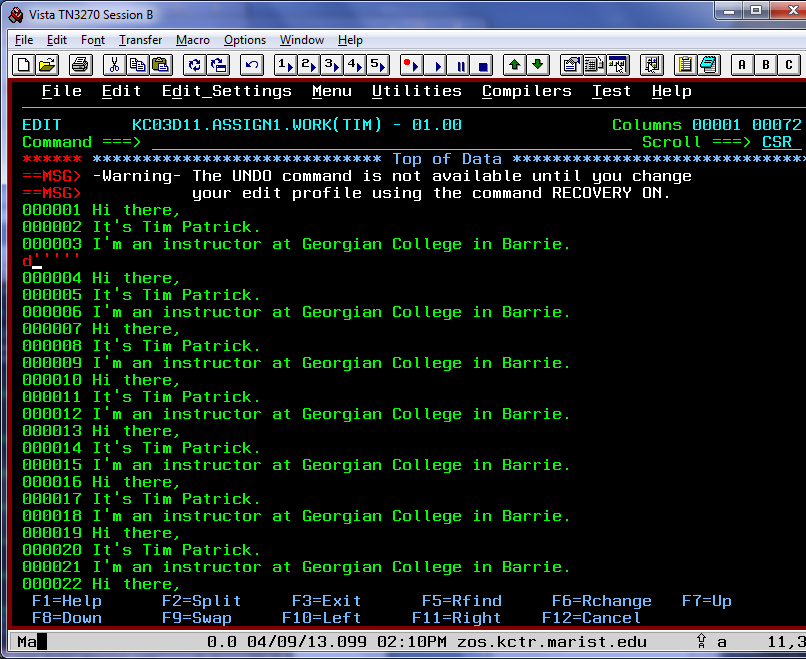
Your screen should now have the two CC commands and the A30 command correctly placed:



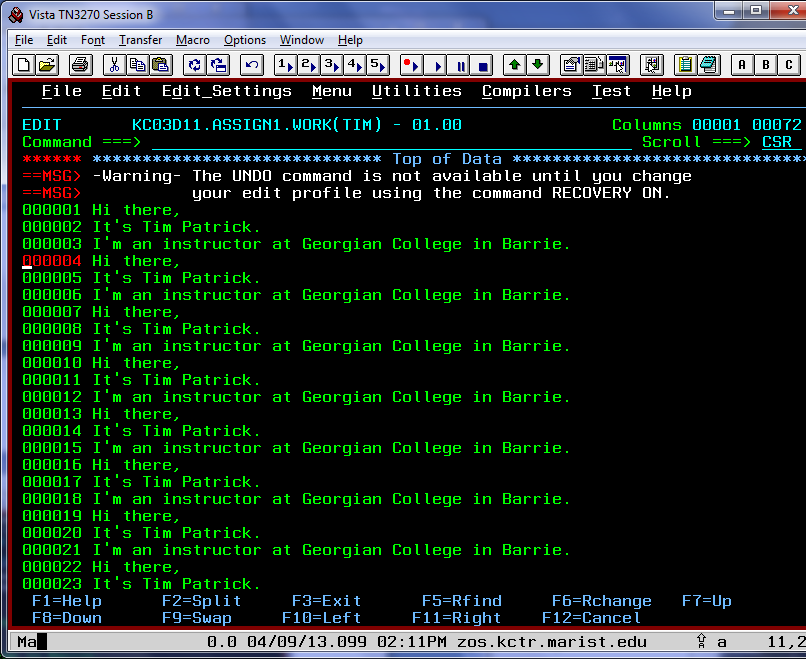
Now press Enter. The three lines you typed has now repeated thirty times, off the bottom of the screen:



Notice there's now only one blank line — ISPF doesn't like wasted resources. Paging up and down using **F7** and **F8**, respectively, will automatically delete blank lines. You can also delete the blank line by putting a D to the left of it and pressing enter, if you prefer. Delete the line now:

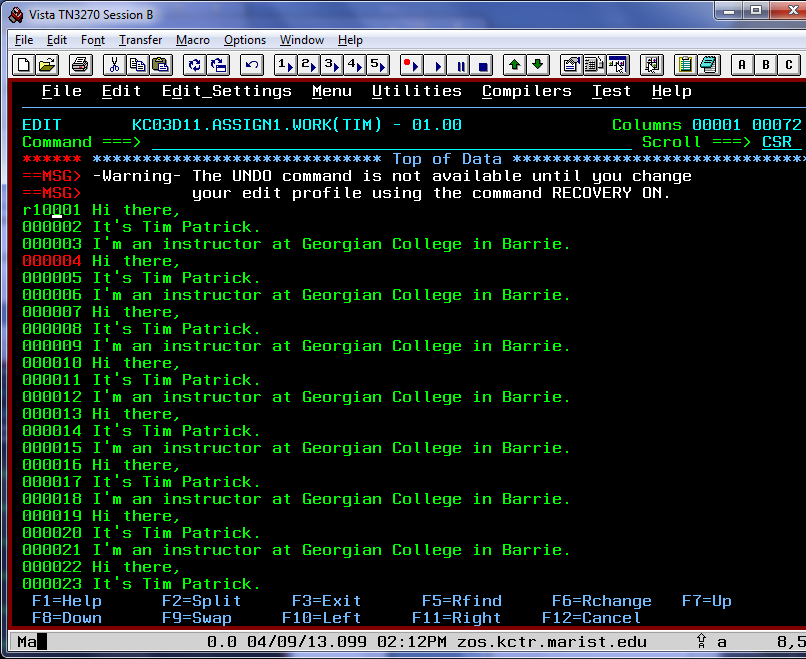


The line should now be gone:

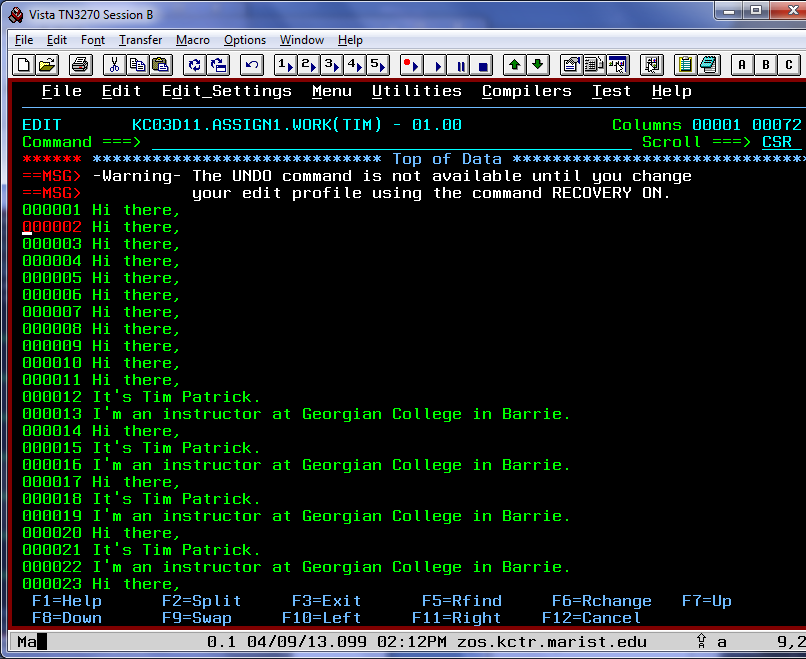


You can use **DD** the same way you used CC to delete sections of text as well.

Now let's repeat the first line of text 10 times. To do this, use the R (repeat) command. Type **R10** on the first line:

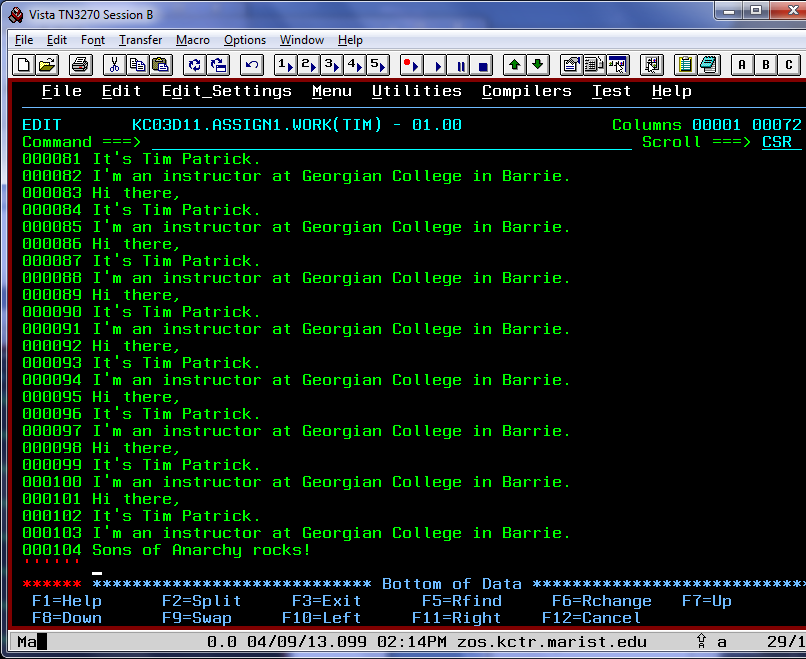


Now press Enter, and your first line repeats ten times, with new line numbers to match:

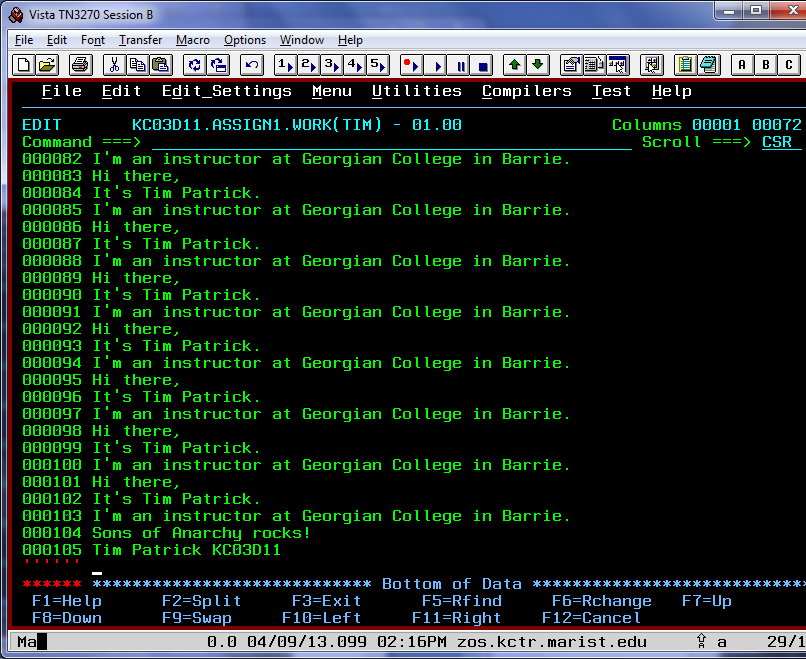


On the command line, enter the command **BOTTOM**. This takes you to the bottom of the member. Tab down and use the **I** command to insert one more line after the last line of the member.   
  
On the new line, let me know your favourite movie or TV show.

Press Enter after you've added the new line:



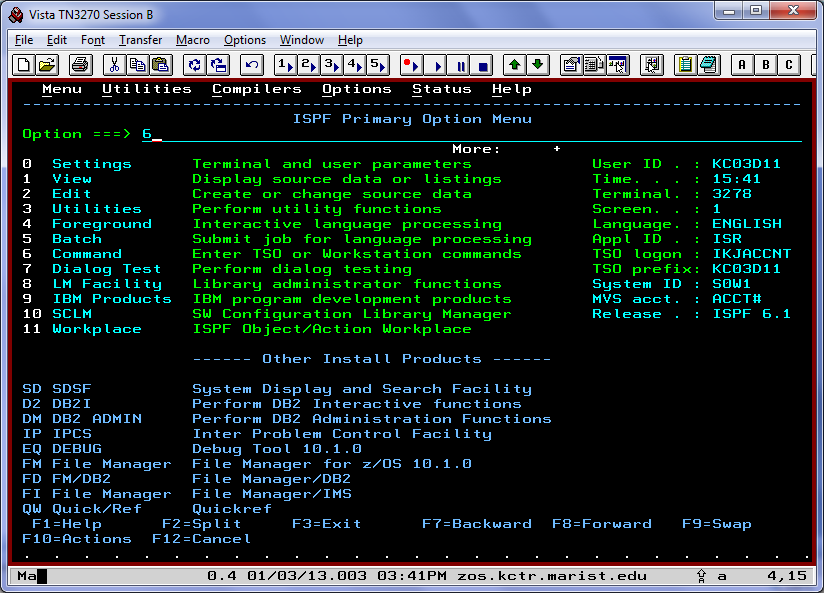
On the blank line at the bottom, type your first and last name, followed by your Marist userid, and press Enter:



F3 out of your data set member - it will be saved automatically (if you ever want to quit without saving, remember that **CAN** for cancel will do the trick).  
  
Now let's run an executable program against the member you just created.

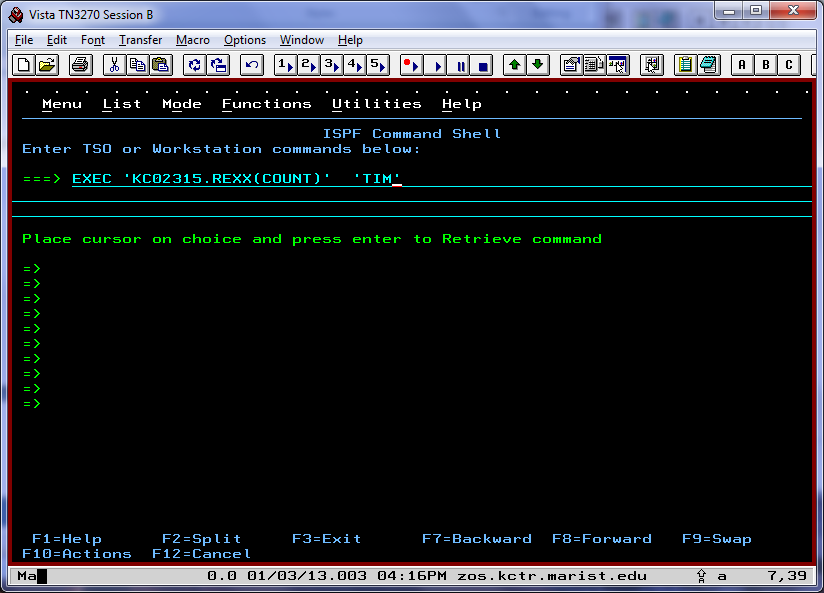
**7. Run an executable against your new data set member (5 Marks)**

From the ISPF Primary Option screen, select option 6: **Command**.

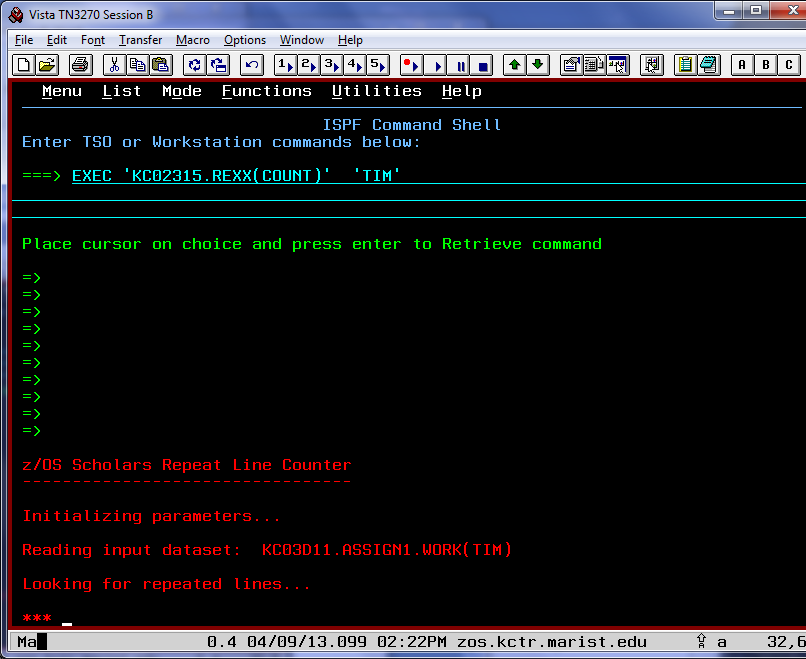


This screen lets you enter TSO commands from ISPF. We need to execute a REXX program (REXX is a powerful, user-friendly programming language first used on the mainframe) that is stored in the member KC02315.REXX(COUNT) against the member you created.

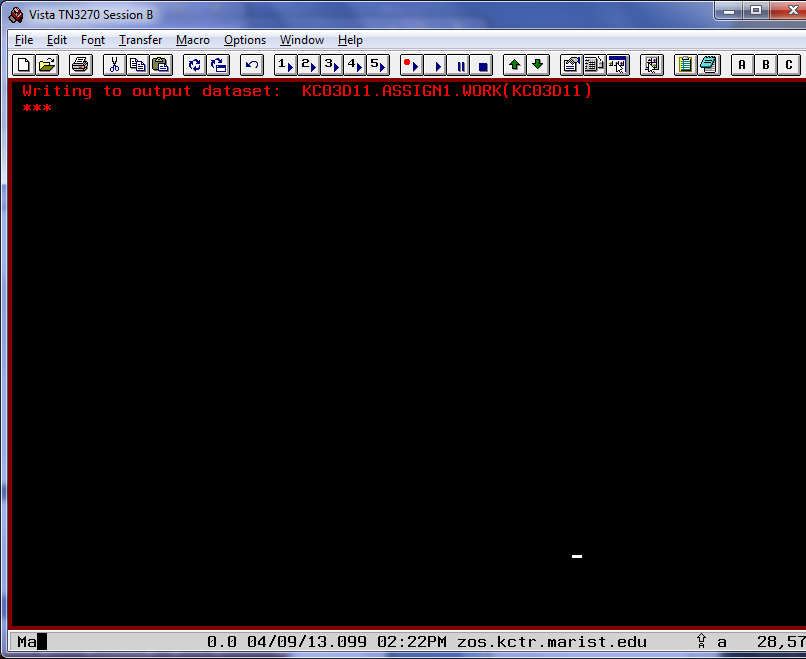
To execute the REXX program (or "exec" short for executable), enter on the command line:   
 **EXEC 'KC02315.REXX(COUNT)' '*your\_name'***   
  
where *your\_name* is either your first name or the first eight letters of your first name, whichever you named the new member in your ASSIGN1.WORK data set:



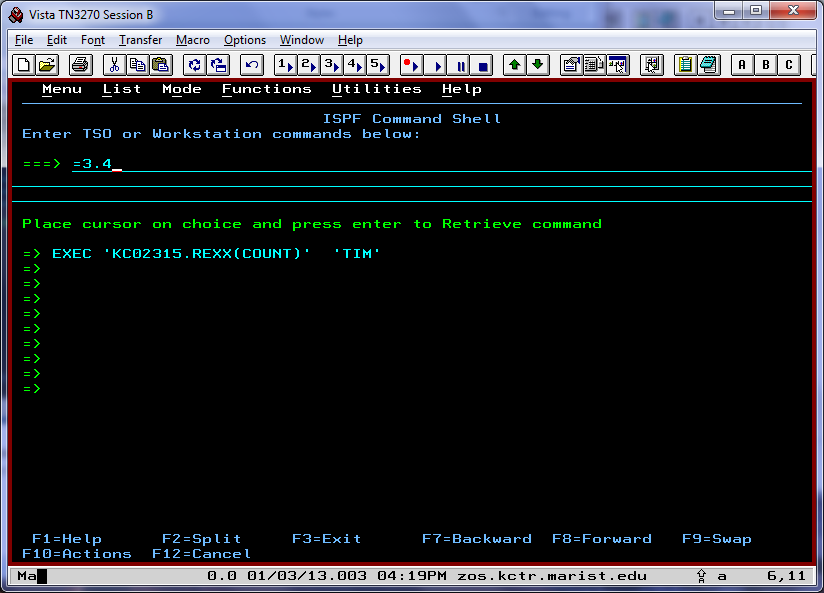
The program will begin running, giving you the following message:



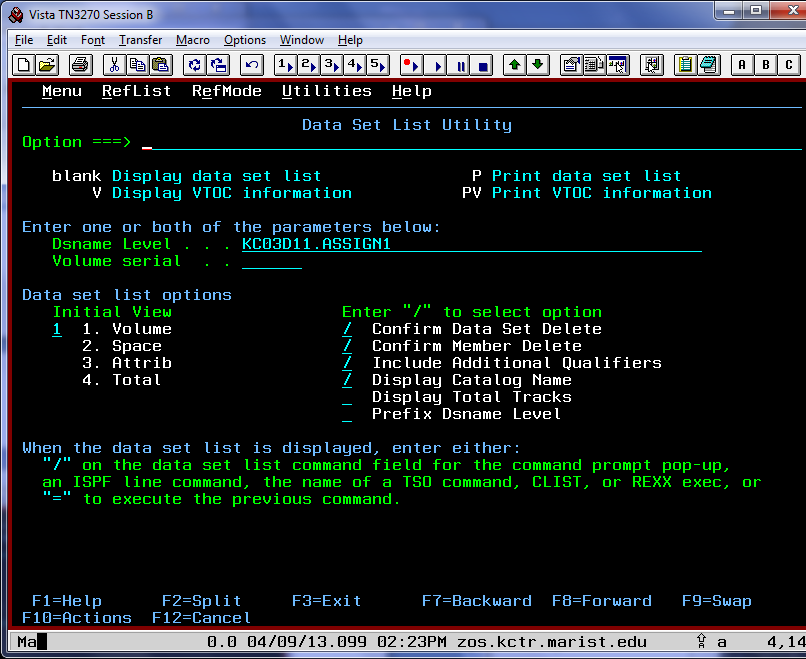
Remember, the \*\*\* means that the system is waiting for your input. Press Enter to continue executing the program.



Press Enter again to clear the system output. The program has now executed. Let's go see what it did! Instead of using F3 to back out to the ISPF Primary Option Menu, let's use a shortcut. On the command line, enter **=3.4** :

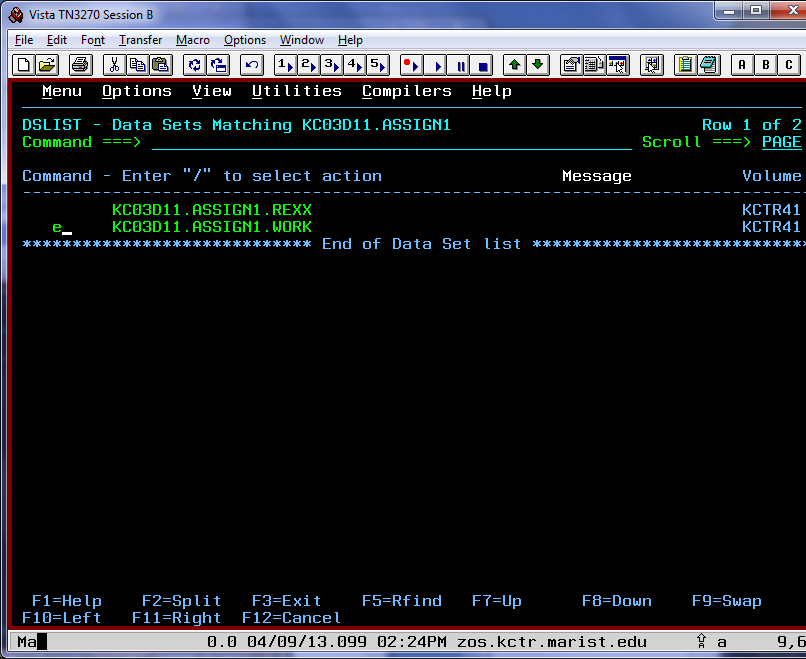


This is the same as choosing option 3 from the Primary Option Menu, then choosing 4 on the next screen. You should recognize the screen you are now looking at as the Data Set List Utility screen:

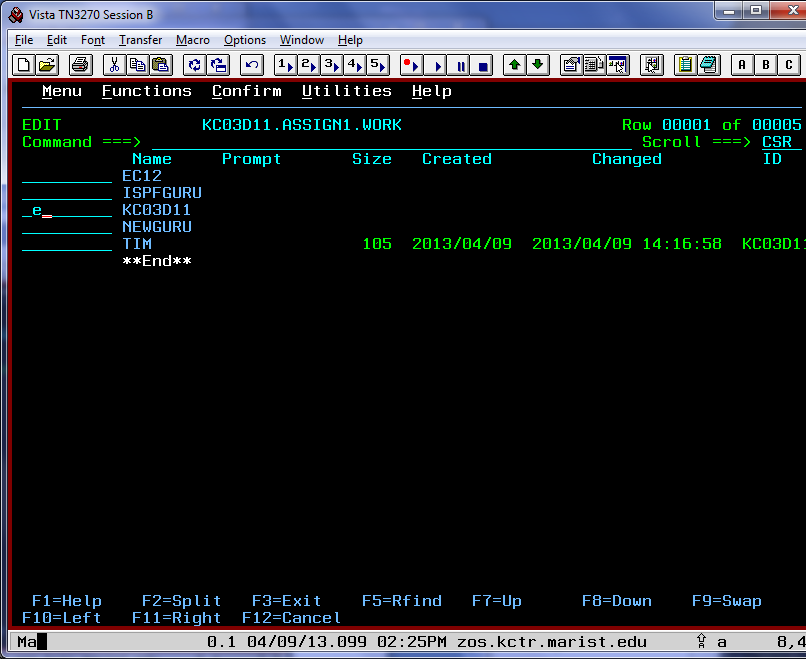


Make sure yourid.ASSIGN1 is specified on the **Dsname Level** field, and press Enter.

On the next screen, put an **E** for edit beside the data set ***yourid***.ASSIGN1.WORK:

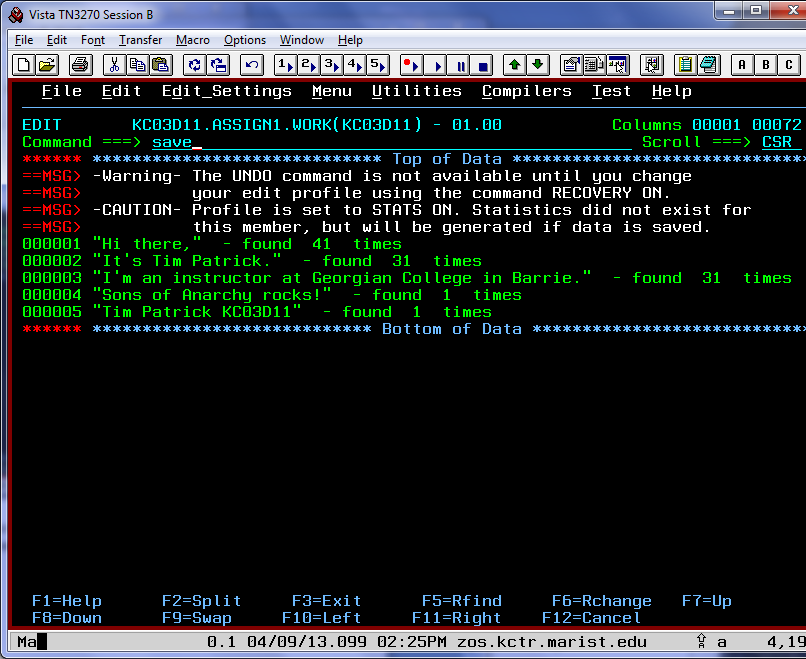


Press Enter. There is a new member in your data set now, named after your userid. Put an E beside it and press Enter to have a look.

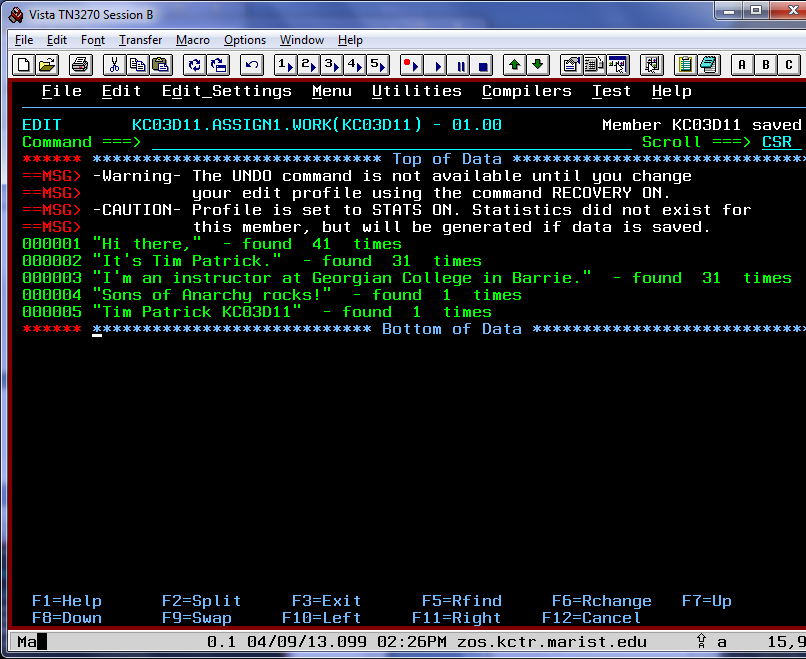


The REXX exec counted each occurrence of the lines of the new member you created and generated output to this new member accordingly.

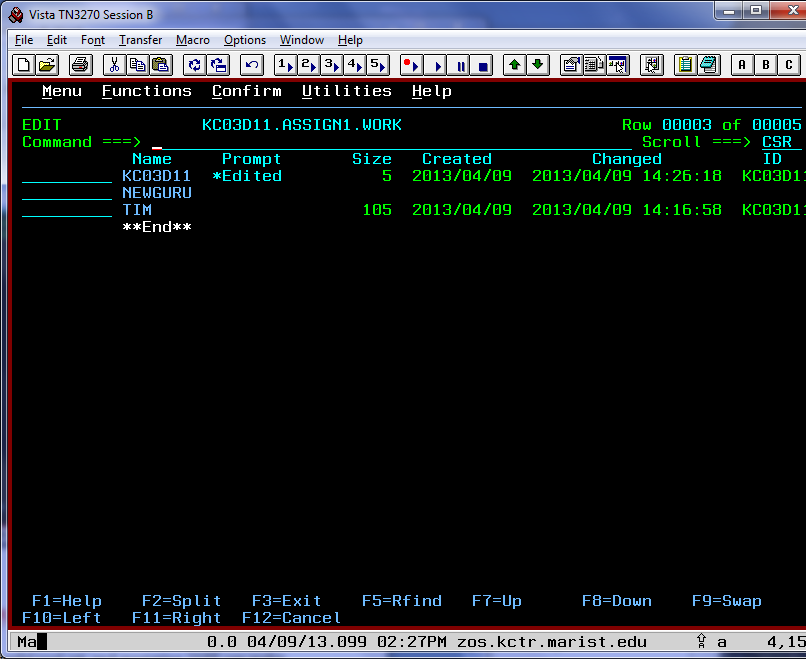
**IMPORTANT:** Type SAVE on the command line and press Enter. Doing this will time-stamp the member so I can tell when you completed this step. **If you fail to do this, you may not receive marks for this part of the assignment**:



After you save the member, you can verify that it was saved correctly by the white message in the upper right-hand corner, **Member KC03*###* saved**:



F3 back one screen. The prompt informs you that the member was edited:



**8. Become an ISPF Edit Guru (15 Marks)**

**Background:** As you've already seen, ISPF is a powerful tool set, and mastering ISPF can make your mainframe life much easier. ISPF can be used in many, many ways, including (but not limited to):

* Using ISPF Edit and the Hardcopy utility to prepare and print memos
* Communicating with z/OS through TSO commands, CLISTs, and REXX execs
* Developing batch or interactive programs
* Monitoring and controlling program libraries

If you'd like to do a little light reading on ISPF, you can find just about everything you ever wanted to know in the [ISPF User's Guide, Volume 1](https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R3SC193627/$file/f54ug00_v2r3.pdf).   
  
Your next challenge is to become comfortable with some advanced ISPF editor commands.   
  
A member named **ISPFGURU** has already been created in your ASSIGN1.WORK data set. Open this member for editing.   
  
Ready to go? Okay. Let's get rid of the "warning" messages at the top of the member. To do this, enter **RESET** on the command line. Voila! The next time you edit the member, it will be back, but it can be dismissed with the same command.   
  
Since RESET is entered on the command line, it is called a *primary command*. The other flavor is a *line command*, which is entered over the line numbers to the left of the text (such as the I and CC commands you used previously). Primary commands affect the entire member and line commands affect only the line (or group of lines) they are entered on.  
  
Now try one more primary command: **RECOVERY**  
  
This command makes it easier to fix any finger flubs that might occur during this challenge. If you enter any incorrect commands, just use the primary command UNDO to undo them (just like Ctrl+z in the PC world). UNDO reverses your edits, one-at-a-time up to the last SAVE point. You can keep undoing until you get the message "No more to UNDO".   
  
Now would be a good time to check out one of your handy reference books, the [ISPF Reference Summary](http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r13.f54rs00/toc.htm). Check out the sections under "PDF Edit and View Commands," which list primary commands and line commands that can be used while in edit mode. To read a more detailed description of any of these commands, the [ISPF Edit and Edit Macros](http://publib.boulder.ibm.com/infocenter/zos/v1r13/topic/com.ibm.zos.r13.f54em00/toc.htm) reference is a good place to look.   
  
Now, back to the ISPFGURU member.   
  
To make some sense of this gibberish, let's use a few ISPF Edit commands.  
  
First, a bit about about labels. Labels can be used to place ranges on the data in a member. They can be specified with a primary command in order to perform operations on a subset of the lines. ISPF has a number of predefined labels, namely **.ZCSR** (which points to the line where the cursor is currently located), **.ZFIRST** (which labels the first line in the data set) and **.ZLAST** (which labels the last line in the data set). These are called special labels and begin with the letter Z. You can learn all you'd ever want to know about labels by reading [this chapter](http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r13.f54em00/linergn.htm) in the ISPF Edit and Edit Macros book.   
  
Okay, let's get in there and label some data! In the prefix (the prefix is the area to the left of the line where line commands go) of Line 18, type .A (that's a period followed by the letter A). Notice that all labels must begin with a period. On Line 19, type .B, on Line 36, type .C and on Line 37, type .D and press enter. We have now defined four labels!   
  
Some extra information: To locate these labels, type LOCATE .A (or LOCATE .B or LOCATE .C or LOCATE .D) on the command line and press Enter. This will position you directly at each of the labels you defined earlier.   
  
  
We can now perform other ISPF actions and limit the range so it is between these labels. If a range is not specified, it defaults to .ZFIRST and .ZLAST, so be extra careful when attempting the steps below.   
  
  
Use the [CHANGE primary command](http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=%2Fcom.ibm.zos.r13.f54em00%2Fchng.htm) to complete the following steps (in order):   
  
Between .ZFIRST and .A labels:

1. Change **all** vertical bars ( | ) to underscores ( \_ )
2. Change **all** question marks ( ? ) to double quotes ( " )
3. Change **all** occurrences of the letter m to a period ( .)

Between .B and .C labels:

* Change **all** exclamation points ( ! ) to grave accents ( ` ) (**Hint:** On a standard keyboard, you'll find grave accent next to the number 1.)
* Change **all** occurrences of the digit 7 to colons ( :)
* Change **all** occurrences of the letter s to forward slashes ( / )

Between .D and .ZLAST labels:

* Change **all** ampersands ( & ) to the capital letter S
* Change **all** hashes ( # ) to the letter capital letter E
* Change **all** dollar signs ($ ) to the capital letter A
* Change **all** asterisks ( \* ) to the capital letter I

Did that last one give you some trouble? In ISPF Edit, the asterisk is a special character that means "the last thing I specified." Put the asterisk in quotes if you're trying to specify the asterisk symbol itself.  
  
Next, use the primary command SORT to sort all the lines within the ranges specified below. The SORT command has many options which allow you to sort on any field of data. First, sort lines between .ZFIRST and .A and then sort the lines between .B and .C.

***Note: If you enter the SORT command without any parameters, try the UNDO command and then re-enter two SORT commands described above.***   
  
Now, enter the following command:   
  
C "SPIDERMAN" "MAINFRAMES"   
  
There you go! Now we're talkin'. Believe it or not, you have now mastered a great deal of very useful ISPF commands: RESET, LOCATE, UNDO, RECOVERY, CHANGE, SORT and even labeling. Impressive!  
  
That's almost it! Except for one more ISPF trick that you'll learn right now. Along with a multi-functional super awesome editor, ISPF includes several useful utilities. One of these is the SuperC compare utility that can be used to compare two data sets (or members within data sets) and flag any changes. To use SuperC, enter the primary command =3.12. Remember, this is the same as going to the ISPF Primary Option Menu and choosing option 3, then option 12.  
  
First you will be prompted for a "new" data set name.   
  
In the "Data Set Name" field, enter '***yourid***.ASSIGN1.WORK(ISPFGURU)' then press Enter.   
  
At the next screen you will be prompted for an "old" data set name.   
  
Update this panel with two data set name changes as follows:  
  
Data Set Name . . . 'KC02315.ASSIGN1.SOLUTION(ISPFGURU)'  
Listing DS Name . . . '***yourid***.ASSIGN1.WORK(SUPERC)'  
  
The Listing DS Name field is the location to which SuperC will save its results.   
  
If you completed every part of this task correctly, you should see a report that tells you there were **39** line matches and **0** total changes. If not, all of the non-matching lines will be listed. If this happens, you can go back and fix those lines or copy 'KC02315.ASSIGN1.WORK(ISPFGURU)' back to your ASSIGN1.WORK data set and start over.

## 9. ISPF Edit Macro (15 marks)

**Background:** In this task, you'll be creating a macro. A macro allows you to execute a series of commands in only one step. The macro you'll be working with will execute all the ISPF commands you entered in the previous section (ISPFGURU) in one step.

**Your challenge:** Correctly execute a macro in ISPF. To execute this macro, we'll be using something called a [CLIST](http://en.wikipedia.org/wiki/CLIST), which is an acronym for Command List. A sample MYMACRO CLIST has been copied to you ASSIGN1.REXX data set which you will modify and run against the NEWGURU member in your ASSIGN1.WORK data set. You'll also need to execute an ALTLIB statement, which will tell the mainframe where to find your macro.   
  
Here are some brief descriptions of the members you'll be using:  
  
ALTLIB - An executable member written in REXX  
MYMACRO - A CLIST macro   
NEWGURU - The member against which you will run your macro  
  
Follow these steps to get your macro functioning correctly:

1. KC03###.ASSIGN1.REXX(ALTLIB). This is a skeleton ALTLIB statement.  
   The first line tells the mainframe that the member is a REXX file. This comment is necessary; without it, the system may assume the file is a CLIST.   
     
   The second line tells the REXX interpreter what to do with non-REXX commands. If the interpreter finds a statement that it doesn't understand, it will attempt to pass it on. In this case, we are telling the interpreter to pass all non-REXX statements to TSO. These statements need to be wrapped in quotations.   
     
   The third line is the TSO command ALTLIB. You will need to alter this statement's parameters to execute it correctly. Go ahead and edit this statement so it points to your own ASSIGN1.REXX data set (change ### to the last three hexadecimal digits of your userid) then hit F3 to save and exit. Next, on the line beside the ALTLIB member, type EXEC and press enter or on the Command line enter:

Command ===> tso ex 'KC03###.ASSIGN1.REXX(altlib)'

This will tell the system where to find the macro when you attempt to execute it.   
  
 **Note:** If you have chosen to use F2 to split your screen, you will need to reissue the ALTLIB macro on the new split screen as well. This can be done the same way as before**.**

1. Update ASSIGN1.REXX(MYMACRO) to include all the ISPF commands you entered to change the ISPFGURU member. Here are a couple of hints:
   * In the CLIST language the ampersand (&) is use to identify a variable (e.g. &var1). If you want to include an ampersand character in a command list, you simply specify two ampersands together (e.g. ISREDIT C '&&' 'S').
   * To set a label in a macro you must use the ISREDIT LABEL command. The syntax is ISREDIT LABEL linenum = labelname. For example to set line 18 to label A, enter:

ISREDIT LABEL 18 = .A

Sound good? Great, give it a try.

The [ISPF Edit and Edit Macros](http://publib.boulder.ibm.com/infocenter/zos/v1r13/topic/com.ibm.zos.r13.f54em00/toc.htm) reference is a good place to look for help.

1. Ready to try out your new macro? Edit ASSIGN1.WORK(NEWGURU) and on the command line enter the command **MYMACRO**.   
     
   If everything worked, you should see the same result as you did in the end of Challenge.
2. If something went wrong and you get funky output, don't fret! You can enter **CANCEL** (or **CAN**) on the command line to cancel the changes you just made to NEWGURU, fix MYMACRO, then execute MYMACRO again.   
     
   Use the same method (=3.12 SuperC compare) to verify that everything matches up, this time saving the SuperC output in ASSIGN1.WORK(SUPER2).

**Note:** If you're using split screens, be sure to navigate to KC03###.ASSIGN1.REXX and execute ALTLIB from both screens. If you neglect this, you may find that your macro commands are not recognized.

Once you have everything worked out, copy MYMACRO to your ASSIGN1.WORK data set (SUPER2 and NEWGURU should already be there).

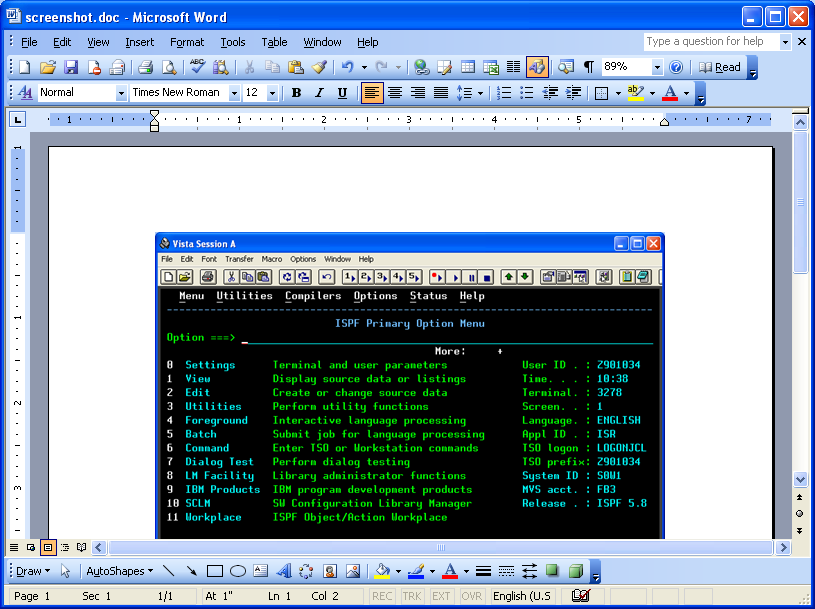
Congratulations you have completed Assignment 1!

## 10. Submitting Assignment 1

Follow the instructions below to submit your assignment.

Create a new folder on your c: drive named; **assignment1**. In the folder add a new empty text file named; ***yourname\_yourid*.TXT** (e.g. TimPatrick\_KC03D11.TXT).

1. Take a screenshot of yourself logged in to the Marist mainframe. Alt-PrtSc will put a copy of the screen on the clipboard.
2. Paste the screenshot into an empty MS-Word document and save it as **c:\assignment1\screenshot.doc**.

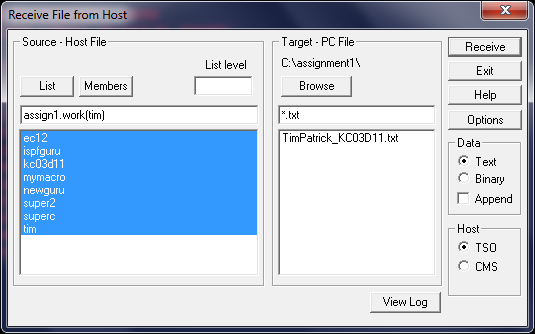


1. As you work through each challenge of assignment 1, you will add members to a PDS named ***yourid.ASSIGN1.WORK***. It should have 8 members when you finish everything.
2. When you are finished the steps of assignment 1, I want you to use the “Tranfer” menu of the Vista TN3270 Emulator to download the contents of this PDS to your **c:\assignment1** folder. See the example below for instructions on downloading a PDS from a mainframe to your PC using the Vista TN3270 Emulator.
3. Zip all the files in your **c:\assignment1** folder into a file called ***your\_name\_assignment#1.zip***, and submit the zip file to me via Blackboard.

**Downloading a mainframe PDS to your PC using the Vista TN3270 Emulator.**

Here’s an example of downloading a mainframe PDS to your PC.

1. Logon to TSO on the Marist mainframe. Note: The transfer is best done from native TSO, so exit ISPF if you are currently in it.
2. Select the “Receive from Host” menu item from the “Transfer” menu on your Vista TN3270 Emulator.
3. In the “Source – From Host” entry field, enter: 'yourid.ASSIGN1.WORK' and click the “members” button.
4. Select all the members in the list that appears by clicking on the first member in the list and then clicking on the last member of the list while holding down the shift button. All members should be highlighted.
5. Under “Target – PC File”, use the “Browse” button to find and select C:\assignment1\yourname\_yourid.txt. In the entry field change yourname\_yourid.txt to an asterisk (\*), make sure “Text” data and “TSO” host are selected and finally click on the “Receive” button. The transfer will take a minute or two.

**