JCL2

JCL: On The Job

Fix some JCL and get that program running right!

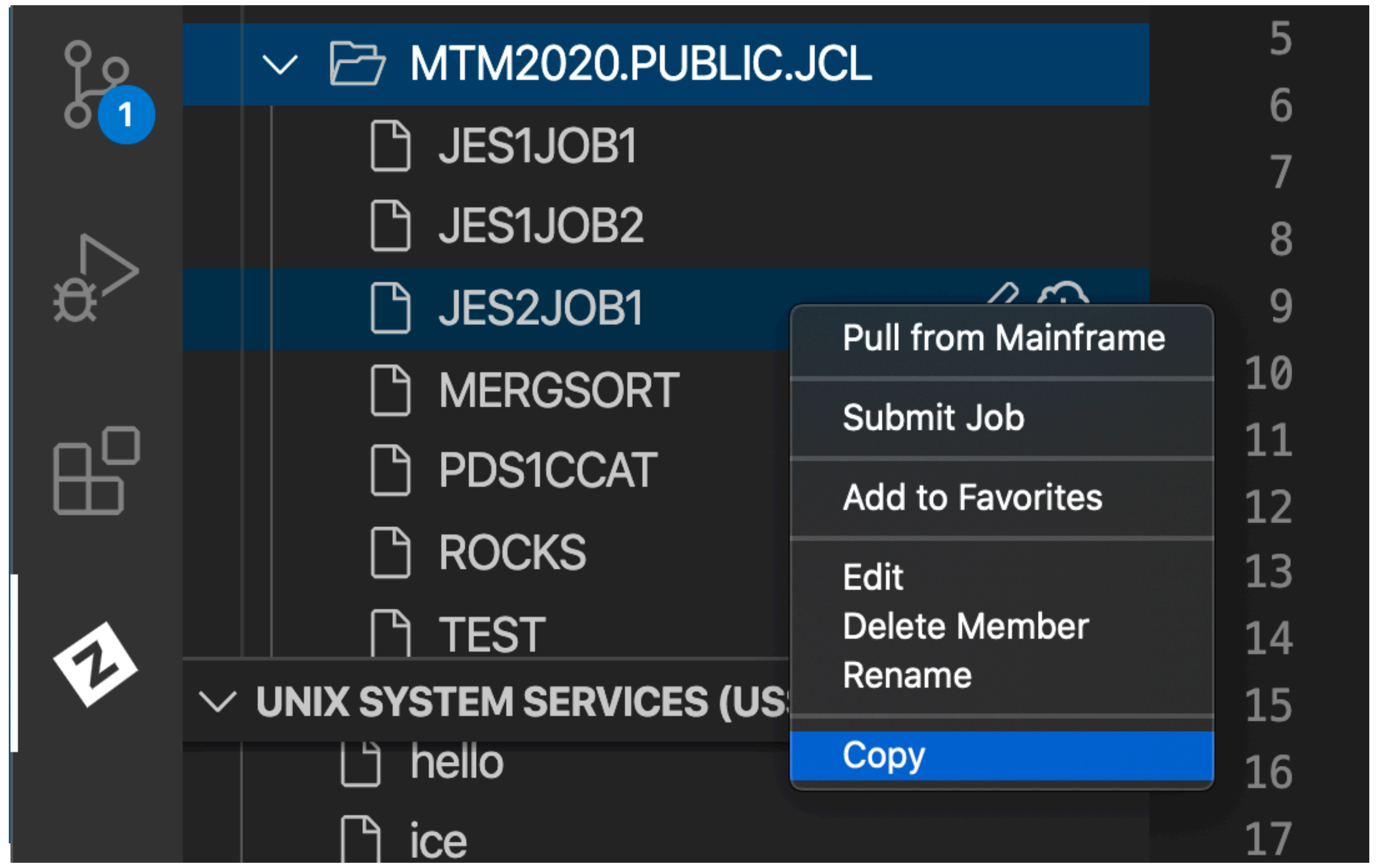
9 steps 60 minutes

THE CHALLENGE

Now that you've seen some JCL and how to submit it, let's try to fix a simple problem in some JCL. The code you'll be looking at is supposed to combine the contents of a few files, but as you'll find out, there is a piece missing. There are clues in the output that will help, and when you're done, the job will run successfully without any errors.

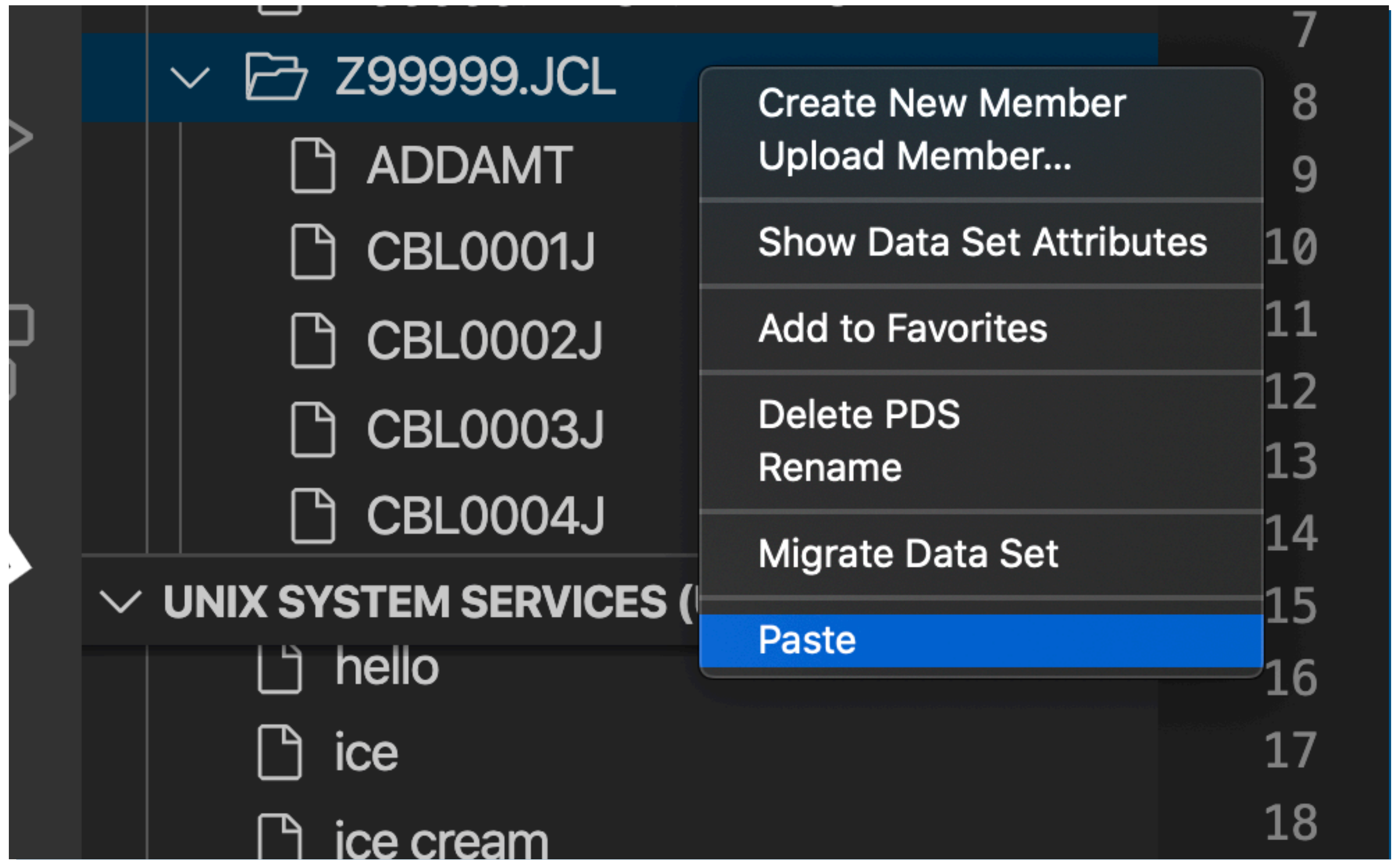
BEFORE YOU BEGIN

You should have completed the JCL1 challenge before attempting this one. Nothing else is really required except the VS Code setup which you've already done.



1. FIND AND COPY JES2JOB1

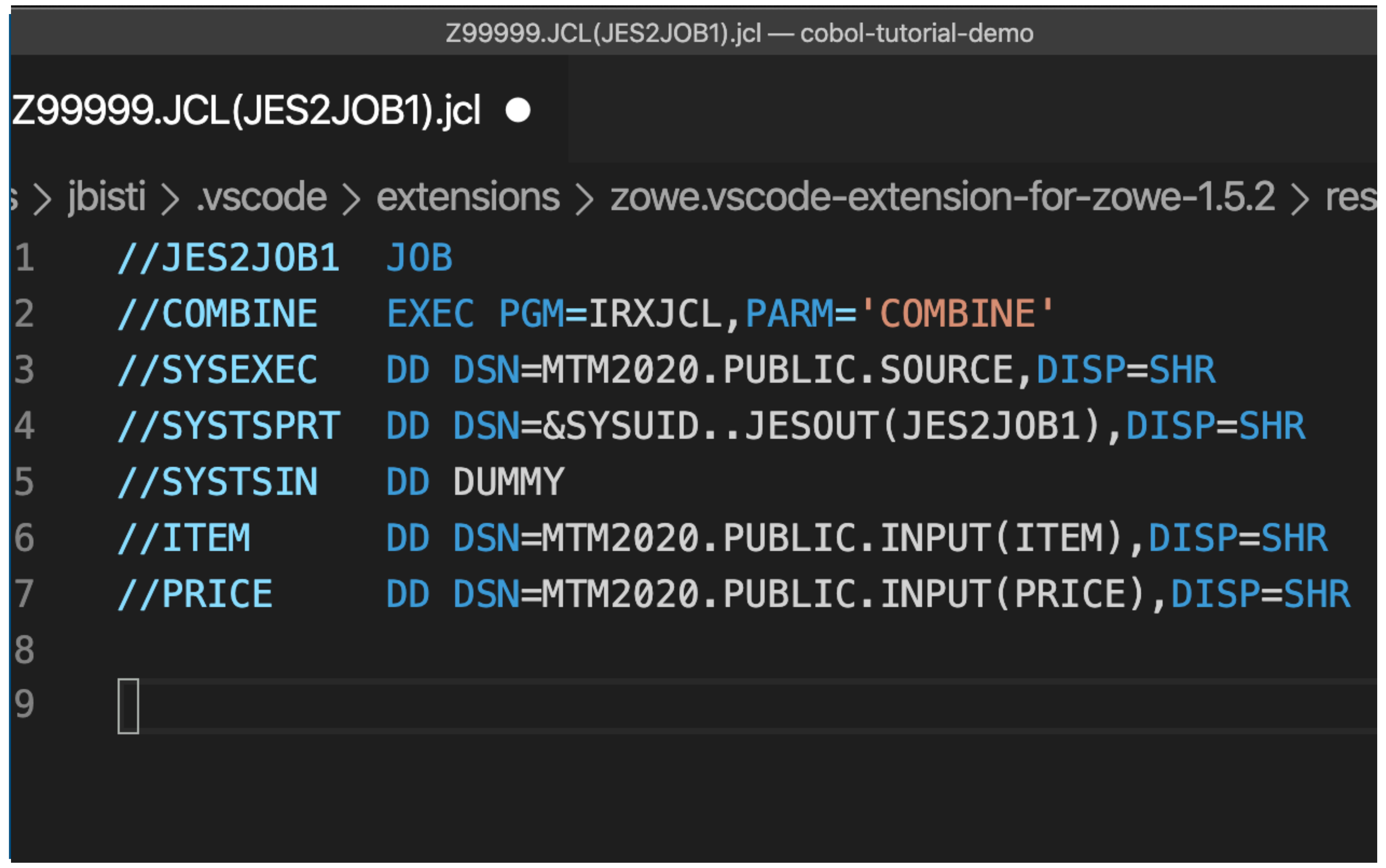
Look in MTM2020.PUBLIC.JCL for a file called **JES2JOB1**. This is the JCL you're going to be working with for this challenge. Right-click on it and select **Copy**.



2. MAKE IT YOUR OWN

Find your own **ZXXXXX.JCL** data set, and right-click on it, selecting **Paste**. This will, as you might have guessed, put a new copy of the **JES2JOB1** file into your JCL data set, so you can edit it.

When pasting, you will be prompted to enter a name. Usually, you'll just want to enter the same name as the source, which is exactly what you should do here.



Z99999.JCL(JES2JOB1).jcl — cobol-tutorial-demo
Z99999.JCL(JES2JOB1).jcl ●
› jbisti › .vscode › extensions › zowe.vscode-extension-for-zowe-1.5.2 › res

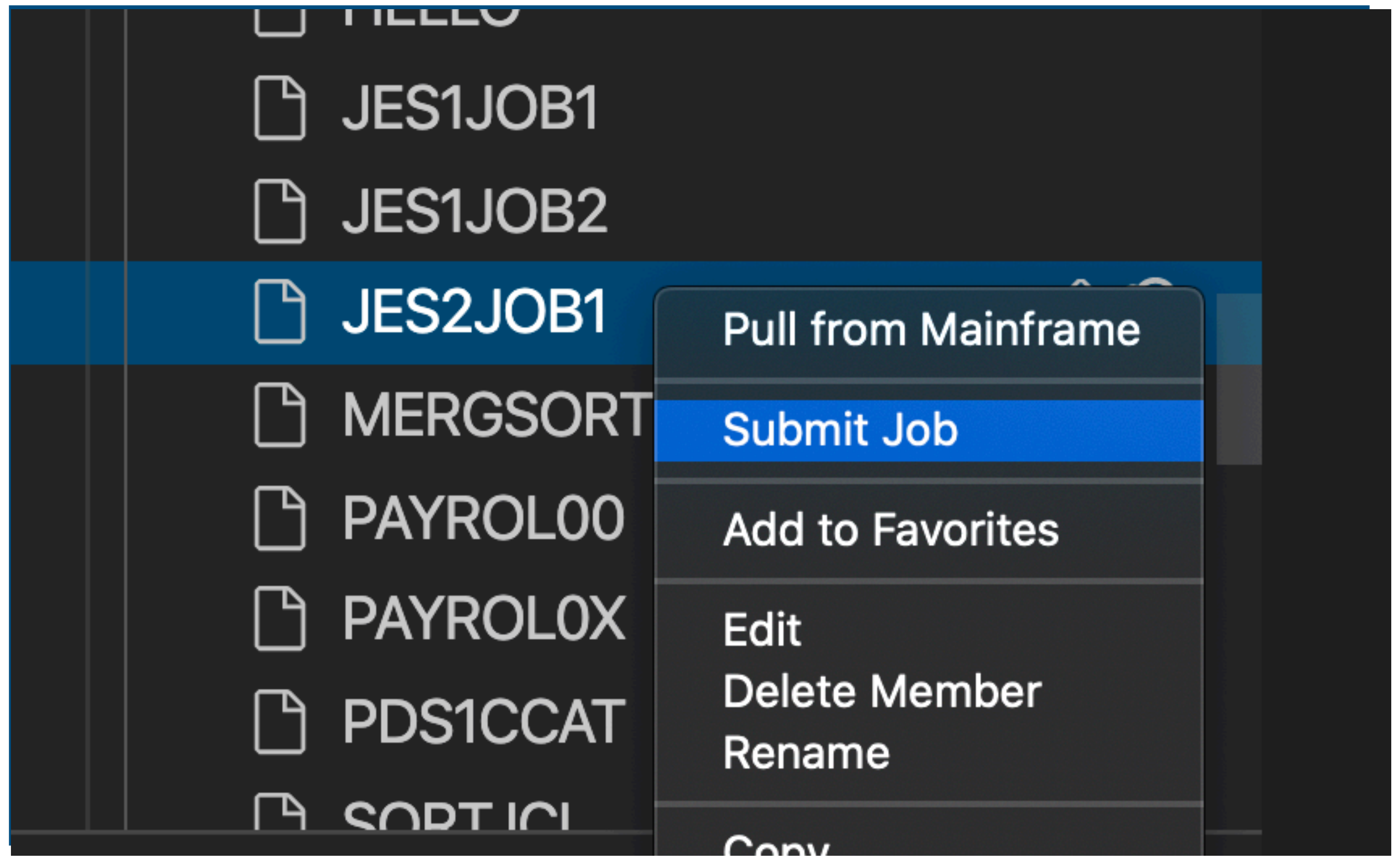
```
//JES2JOB1  JOB
//COMBINE   EXEC PGM=IRXJCL,PARM='COMBINE'
//SYSEXEC   DD DSN=MTM2020.PUBLIC.SOURCE,DISP=SHR
//SYSTSPRT  DD DSN=&SYSUID..JESOUT(JES2JOB1),DISP=SHR
//SYSTSIN   DD DUMMY
//ITEM      DD DSN=MTM2020.PUBLIC.INPUT(ITEM),DISP=SHR
//PRICE     DD DSN=MTM2020.PUBLIC.INPUT(PRICE),DISP=SHR
```

3. OPEN IT UP

Click on the file (from your own **JCL** data set, not the MTM2020 one) and check out the code. It looks like it's running the **IRXJCL** program with the '**COMBINE**' parameter. It's doing something in the **OUTPUT** data set, and also using two files from the **MTM2020.PUBLIC.INPUT** data set.

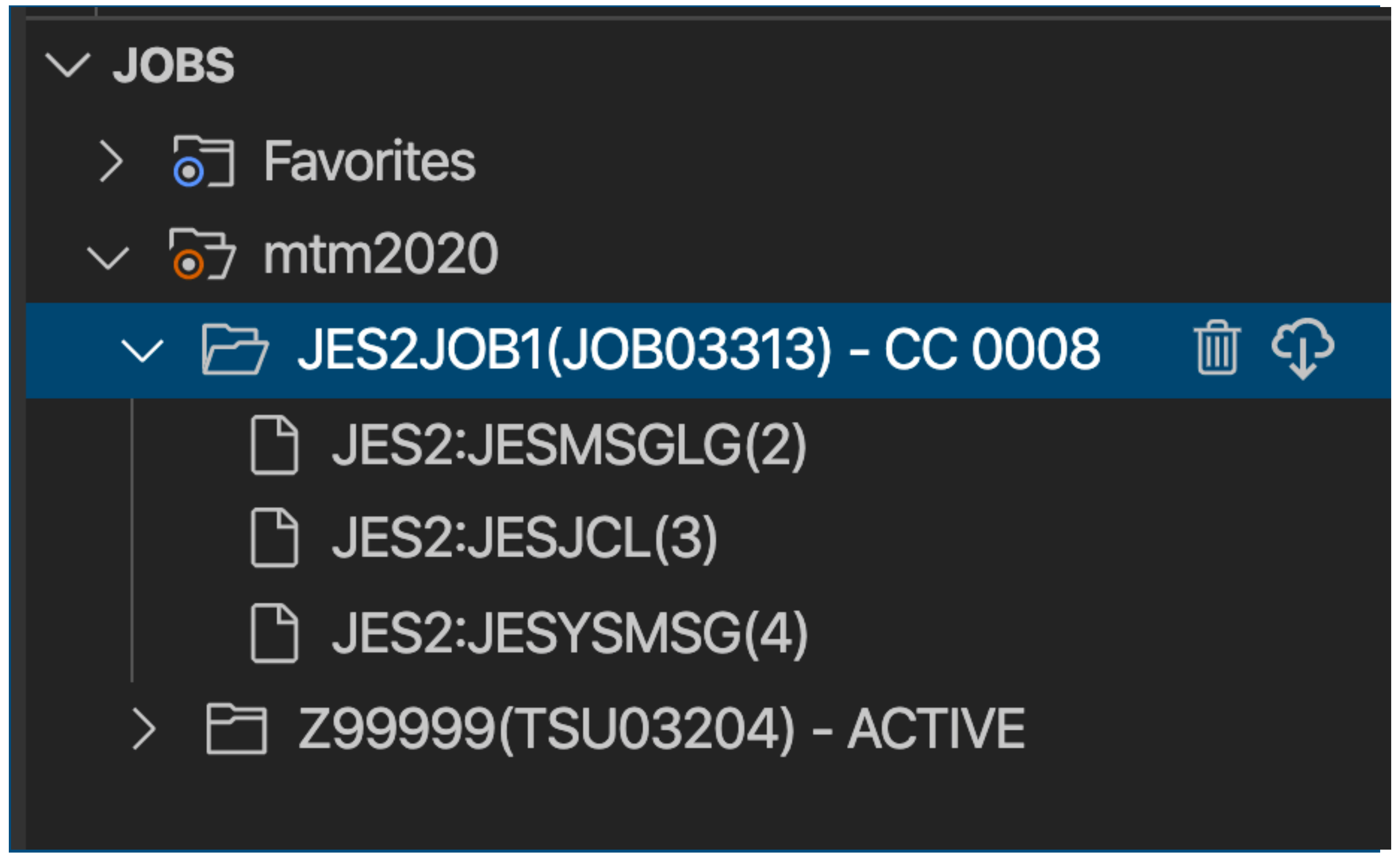
Want to talk? Join our Slack
ibm.biz/mtm_slack

Tweet about it!
#MasterTheMainframe



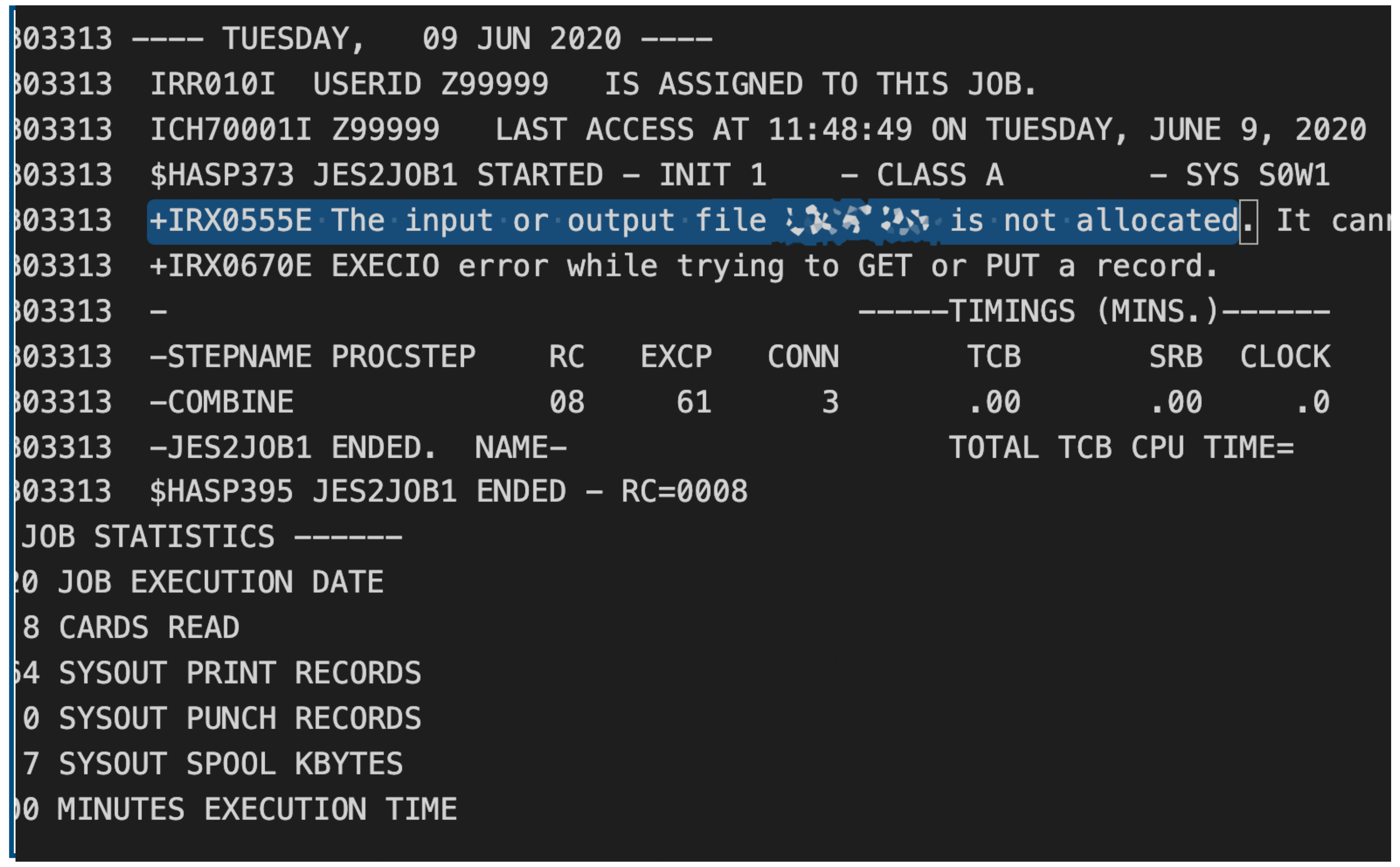
4. RUN THAT CODE

Right-click on **JES2JOB1** (again, making sure to select the one in your own JCL data set, not the MTM2020 one) and select **'Submit Job'.** Notice the Job Name that pops up in the lower right corner of your screen. It should start with JOB, followed by some numbers.



5. CHECK THE OUTPUT

Go down to JOBS and find the job you just submitted, which matches the numbers that popped up in Step 4. You may need to reload the view to find it. You will notice that it got a CC 0008. That's a condition code, and 8 means that there was definitely an error.



```
303313 ---- TUESDAY,   09 JUN 2020 ----
303313  IRR010I  USERID Z99999   IS ASSIGNED TO THIS JOB.
303313  ICH70001I Z99999   LAST ACCESS AT 11:48:49 ON TUESDAY, JUNE 9, 2020
303313  $HASP373 JES2JOB1 STARTED - INIT 1    - CLASS A         - SYS S0W1
303313  +IRX0555E The input or output file [redacted] is not allocated. It cann
303313  +IRX0670E EXECIO error while trying to GET or PUT a record.
303313  -                                        -----TIMINGS (MINS.)------
303313  -STEPNAME PROCSTEP    RC   EXCP   CONN       TCB       SRB  CLOCK
303313  -COMBINE              08     61      3       .00       .00     .0
303313  -JES2JOB1 ENDED.  NAME-                     TOTAL TCB CPU TIME=
303313  $HASP395 JES2JOB1 ENDED - RC=0008
 JOB STATISTICS ------
20 JOB EXECUTION DATE
 8 CARDS READ
54 SYSOUT PRINT RECORDS
 0 SYSOUT PUNCH RECORDS
 7 SYSOUT SPOOL KBYTES
00 MINUTES EXECUTION TIME
```

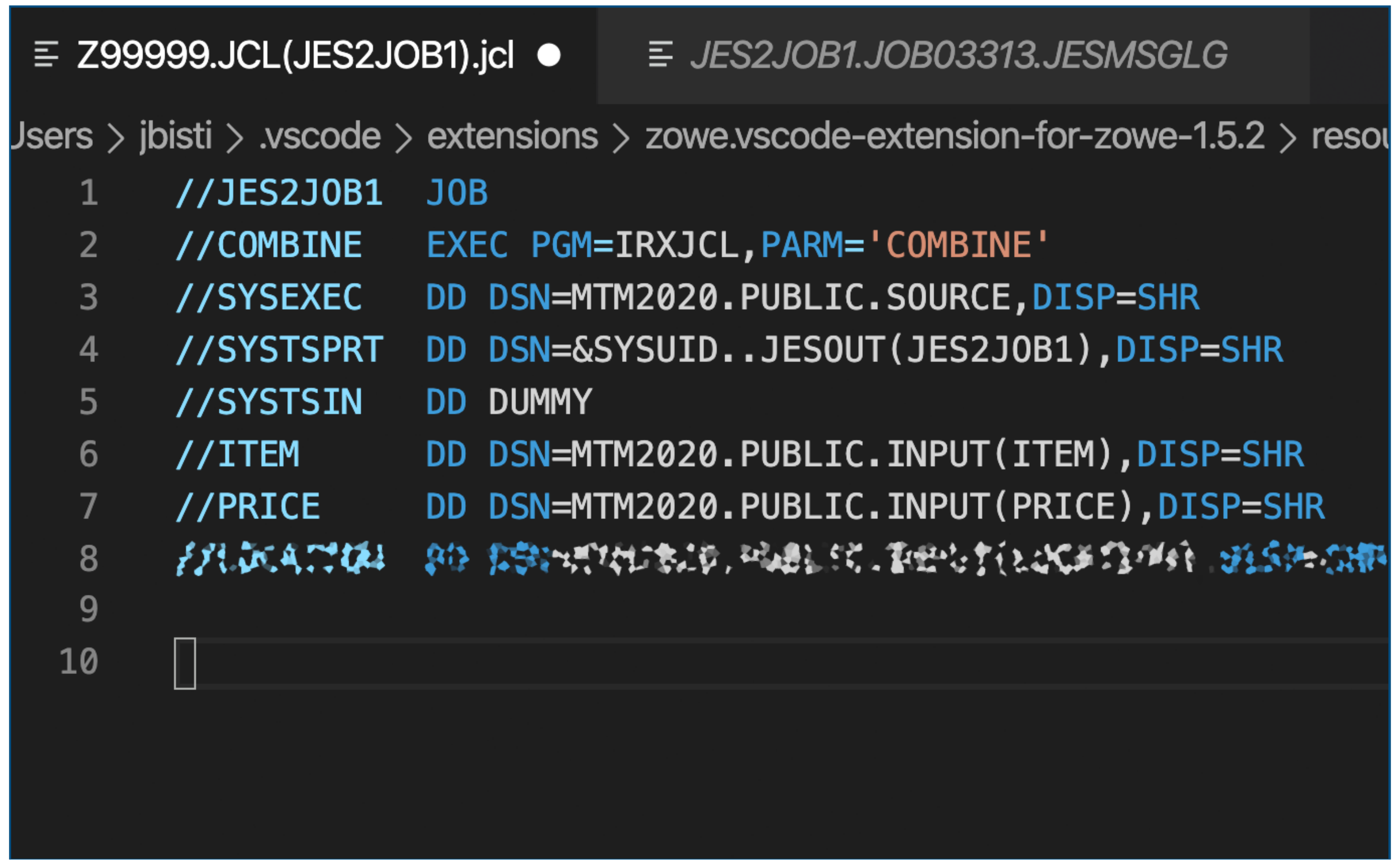
6. INVESTIGATE THE PROBLEM

Poke around the output that the failing job created. You may not be able to make sense of much of this, but one line in JESMSGLG gives a very clear clue about why the program didn't run.

"HOW DO I KNOW WHAT INPUT A PROGRAM NEEDS?"

JCL is often used to run programs on z/OS, and part of running a program is giving that program the files that it needs to run successfully The last few lines of the JCL we're working on here are called DD Statements. DD stands for Data Definition, and they spell out the name of the data being defined, as well as its location, and how it should be handled.

In this case, the program is looking for a third piece of input that was not there in the JCL you were given. You can copy/paste the second DD statement, and then alter it to match what the program is looking for, based on what you discovered in the failing program's output from Step 5.


Z99999.JCL(JES2JOB1).jcl ●
JES2JOB1.JOB03313.JESMSGLG
Users > jbisti > .vscode > extensions > zowe.vscode-extension-for-zowe-1.5.2 > resou

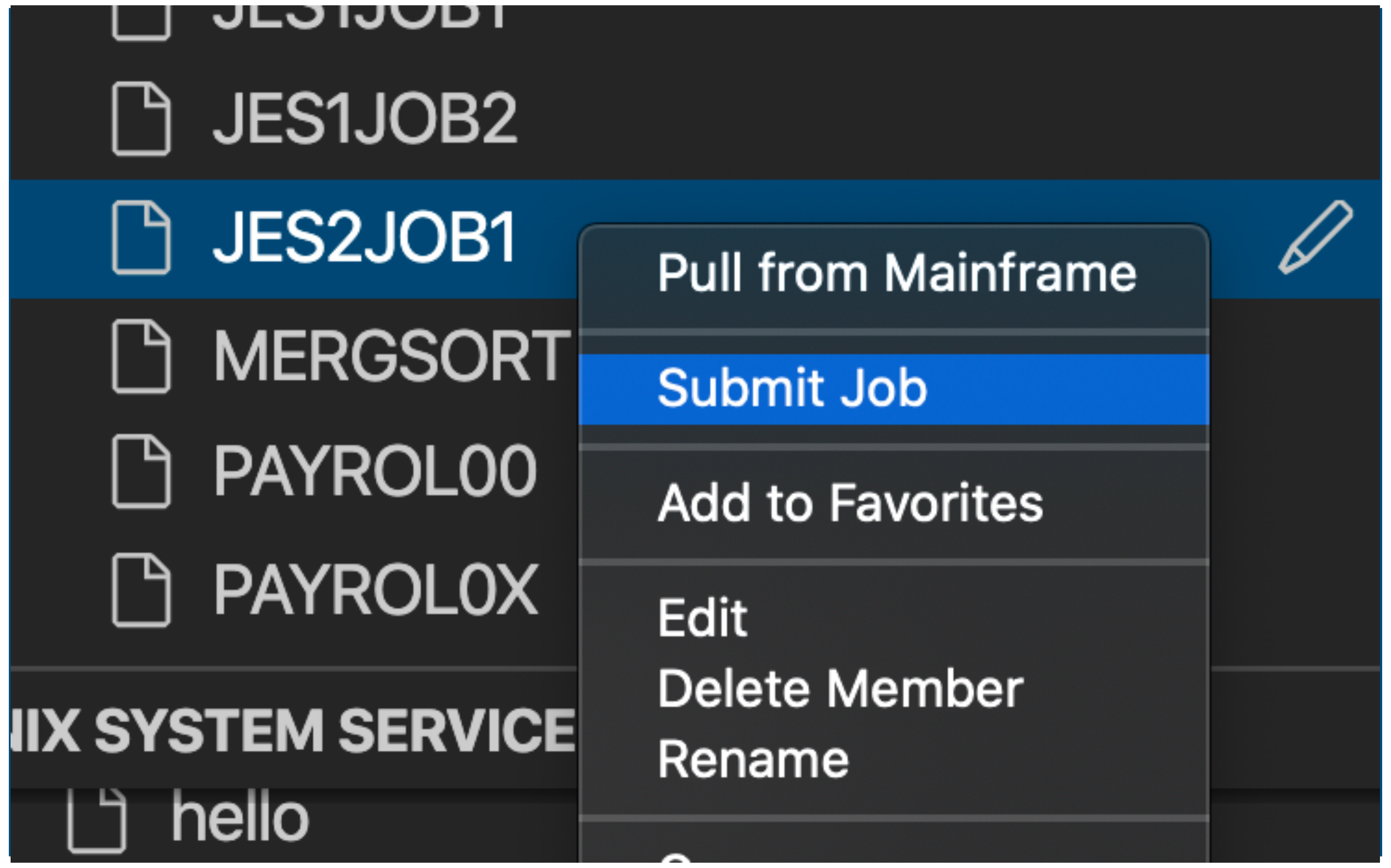
```
//JES2JOB1  JOB
//COMBINE   EXEC PGM=IRXJCL,PARM='COMBINE'
//SYSEXEC   DD DSN=MTM2020.PUBLIC.SOURCE,DISP=SHR
//SYSTSPRT  DD DSN=&SYSUID..JESOUT(JES2JOB1),DISP=SHR
//SYSTSIN   DD DUMMY
//ITEM      DD DSN=MTM2020.PUBLIC.INPUT(ITEM),DISP=SHR
//PRICE     DD DSN=MTM2020.PUBLIC.INPUT(PRICE),DISP=SHR
[illegible]
```

7. MAKE THE UPDATE

Head back to your copy of JES2JOB1 and try to fix the problem. There is one missing line of code which you have to add. You can use the PRICE DD Statement as a reference for the line you need to create, but there are two parts of that line that you'll have to change to make the job work correctly.

If you want an additional clue, look in the MTM2020.PUBLIC.INPUT data set to see what other files in there might be helpful for this program.

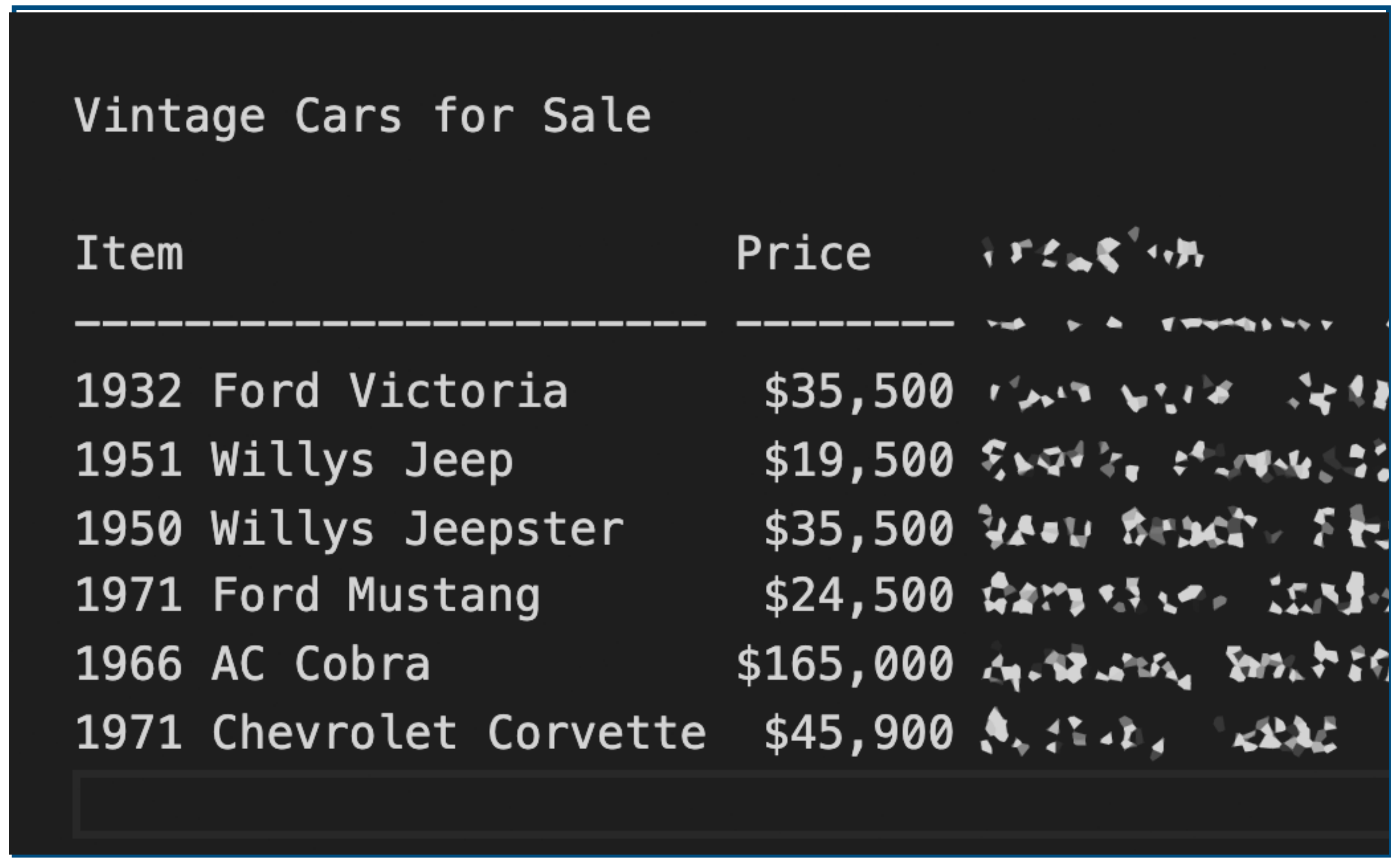
And remember, spacing matters! Make sure those columns line up.



8. SAVE AND SUBMIT

Make sure to Save a file whenever you've made changes. If you don't, and try to re-submit it, it will just submit the old version without your updates.

Once it's saved, right-click and select Submit Job, again noting the number that pops up so you can easily find it in the Jobs output.



```
Vintage Cars for Sale

Item                     Price     [illegible]
------------------------ --------  [illegible]
1932 Ford Victoria        $35,500  [illegible]
1951 Willys Jeep          $19,500  [illegible]
1950 Willys Jeepster      $35,500  [illegible]
1971 Ford Mustang         $24,500  [illegible]
1966 AC Cobra            $165,000  [illegible]
1971 Chevrolet Corvette   $45,900  [illegible]
```

9. CHECK THE OUTPUT

Look through the output and hopefully you got a Condition Code of 0000, meaning no problems. If not, review Step 7, and make sure you're saving and submitting the right copy of the JCL. When corrected, the job should produce output in your OUTPUT data set called JES2JOB1 similar to what you see above. If you got that, open up MTM2020.PUBLIC.JCL and look for a member named CHK. Right-click on that bad boy, and select SUBMIT JOB to mark it as complete.

NICE JOB! LET'S RECAP

Congratulations on fixing your first JCL program. A z/OS Sysprog will spend a lot of time hunting down bugs in code. Sometimes the error is just a misunderstanding between what the programmer *meant* to do, and how it actually got executed by the system. Other times, it's just a simple slip of the finger that turned an M into an N, or put an extra space between two fields. We've all been there.

NEXT UP...

You've got data sets, you've got jobs. Why not go wild and try some Unix System Services?

Try out USS1 and USS2 (preferably in that order) and you'll be ready to handle even more mainframe tasks.