The function rvm\_begin\_trans should take in the arguments:

* Rvm\_t rvm
* int numsegs
* Void \*\* segbases

-create a variable called **temptrans** of type trans\_base\_t.

-create a list called **temprangelist** of type trans\_range\_t.

-create a list iterator called **ii.**

-create an int **i**

- iterate through **rvm**

- check whether the segbase of each rvm is the same as the segbase at position i of the argument segbases && the rvm’s segbase is busy.

- if a match is found then return -1.

- check whether the segbase of each rvm is the same as one of the segbases given in the argument. If so, set the rvm’s segbase to 1.

- now you’re gonna add temptrans (the variable created at the beginning of this function) to the transactions stack of the rvm and increment your transaction count.

- rvm is added to the **trtorvm** array.

- for each of the numsegments, do the following:

* select the trans\_base\_t at the back of the rvm->transactions stack.
* select the mods[segbases[i]] and put temprangelist (the list created at the beginning of the function) in that location.
* Insert segbases into the segbases list at back of rvm->transactions stack.
* Return the tid of the back of the rvm->transactions stack.

To begin the transaction: you add it to the stack and check if the segment it plans to modify is currently busy. You return the tid of the back of the transactions list. That is the transaction id of the last transaction.

The function rvm\_about\_to\_modify will do memcpy and malloc onto the map of trans\_base\_t.