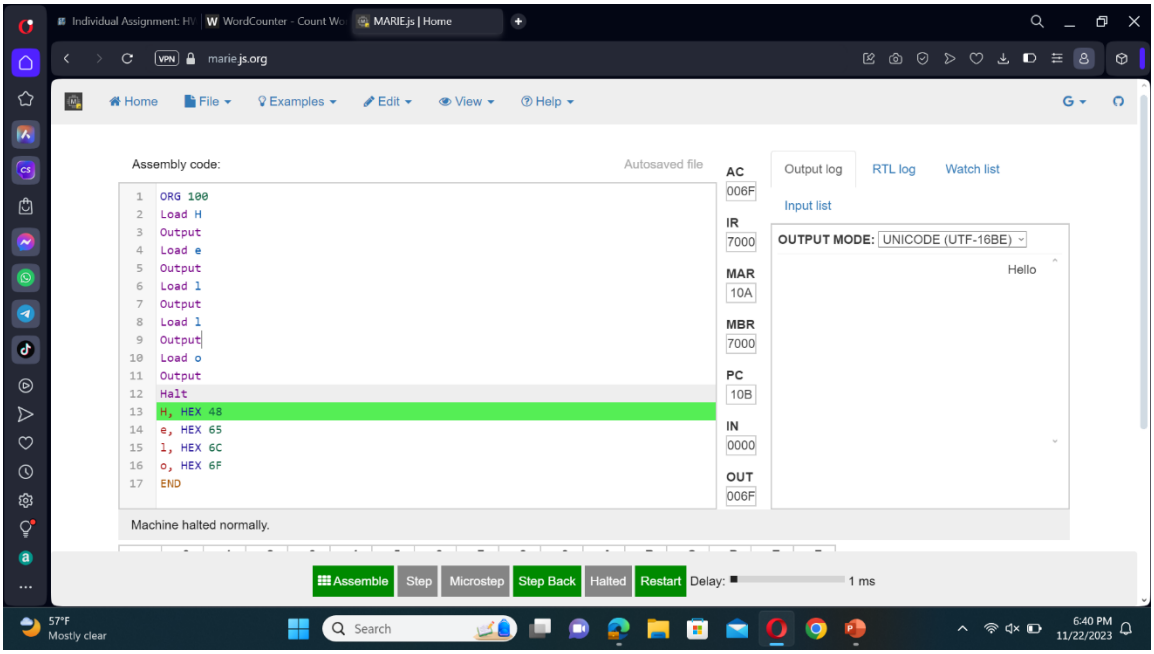
1. Write the program to print the string "Hello" in MARIE assembly language

```
ORG 100
Load H
Output
Load e
Output
Load l
Output
Load l
Output
Load o
Output
Halt
H, HEX 48
e, HEX 65
l, HEX 6C
o, HEX 6F
END
```



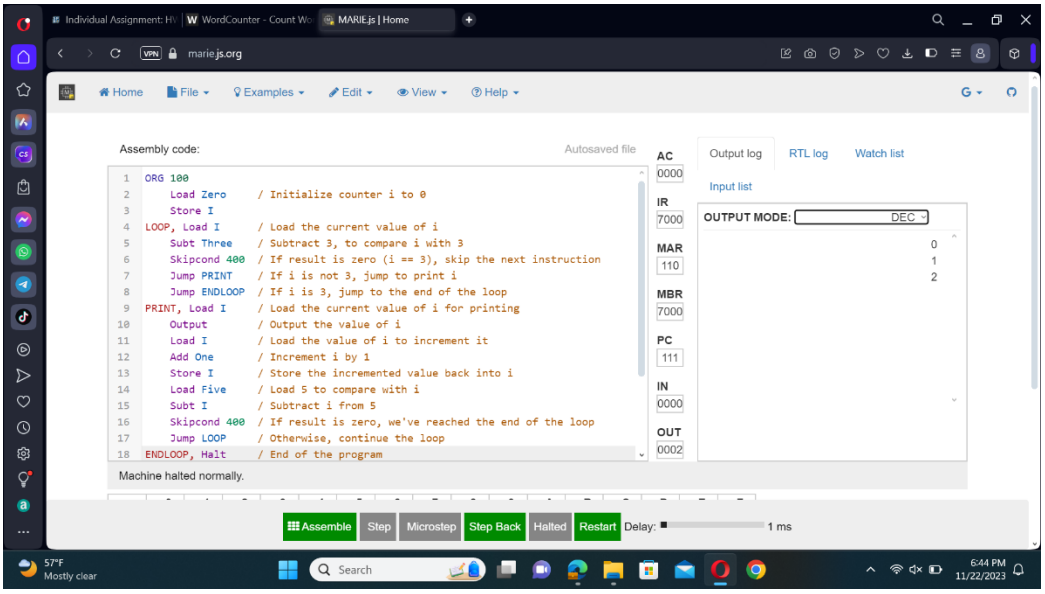
2. Write the MARIE assembly program to implement "break" statement in for-loop shown as follows in Python program.

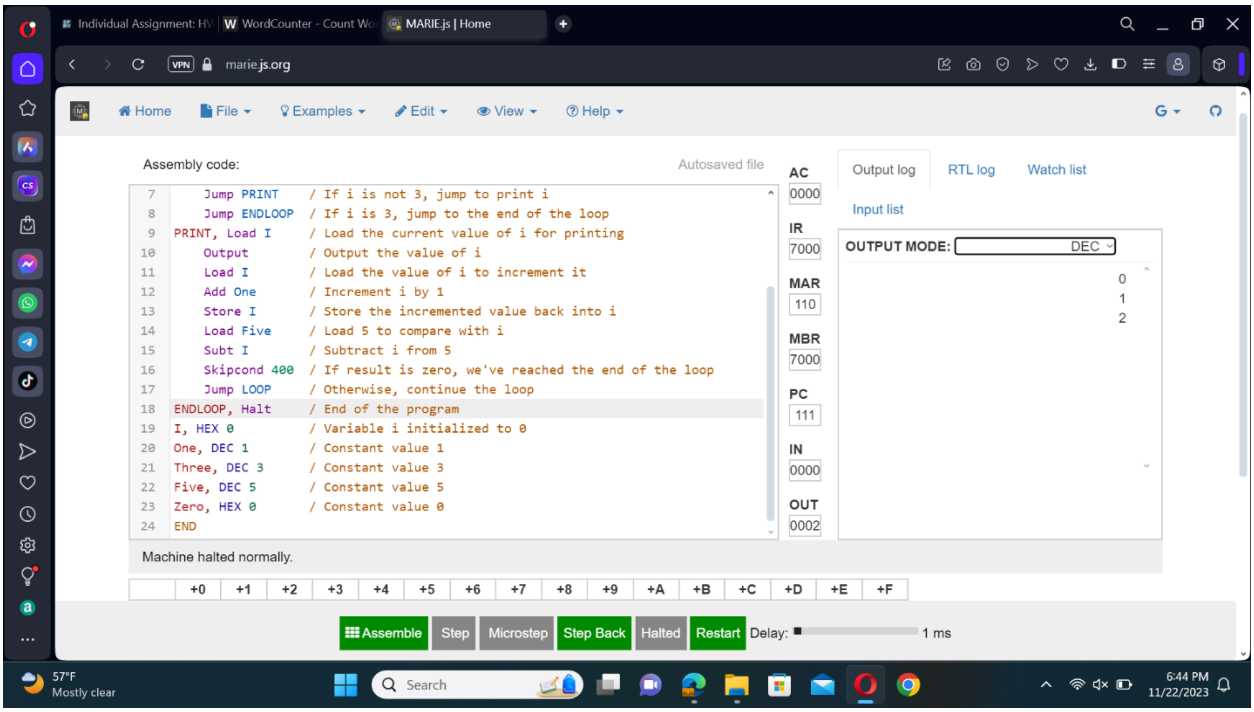
```
for i in range(5):
if i == 3:
break
print(i)
0
1
2
```

```
ORG 100
    Load Zero     / Initialize counter i to 0
    Store I
LOOP, Load I      / Load the current value of i
    Subt Three    / Subtract 3, to compare i with 3
    Skipcond 400  / If result is zero (i == 3), skip the next instruction
    Jump PRINT    / If i is not 3, jump to print i
    Jump ENDLOOP  / If i is 3, jump to the end of the loop
PRINT, Load I     / Load the current value of i for printing
    Output        / Output the value of i
    Load I        / Load the value of i to increment it
    Add One       / Increment i by 1
    Store I       / Store the incremented value back into i
    Load Five     / Load 5 to compare with i
    Subt I        / Subtract i from 5
    Skipcond 400  / If result is zero, we've reached the end of the loop
    Jump LOOP     / Otherwise, continue the loop
ENDLOOP, Halt     / End of the program
I, HEX 0          / Variable i initialized to 0
One, DEC 1        / Constant value 1
Three, DEC 3      / Constant value 3
Five, DEC 5       / Constant value 5
```

Zero, HEX 0 / Constant value 0

END





3. As the question above, it is very similar but needs to implement "continue" statement in MARIE assembly language within the for-loop as follows in Python program.

```
for i in range(5):
if i == 3:

continue
print(i)
```

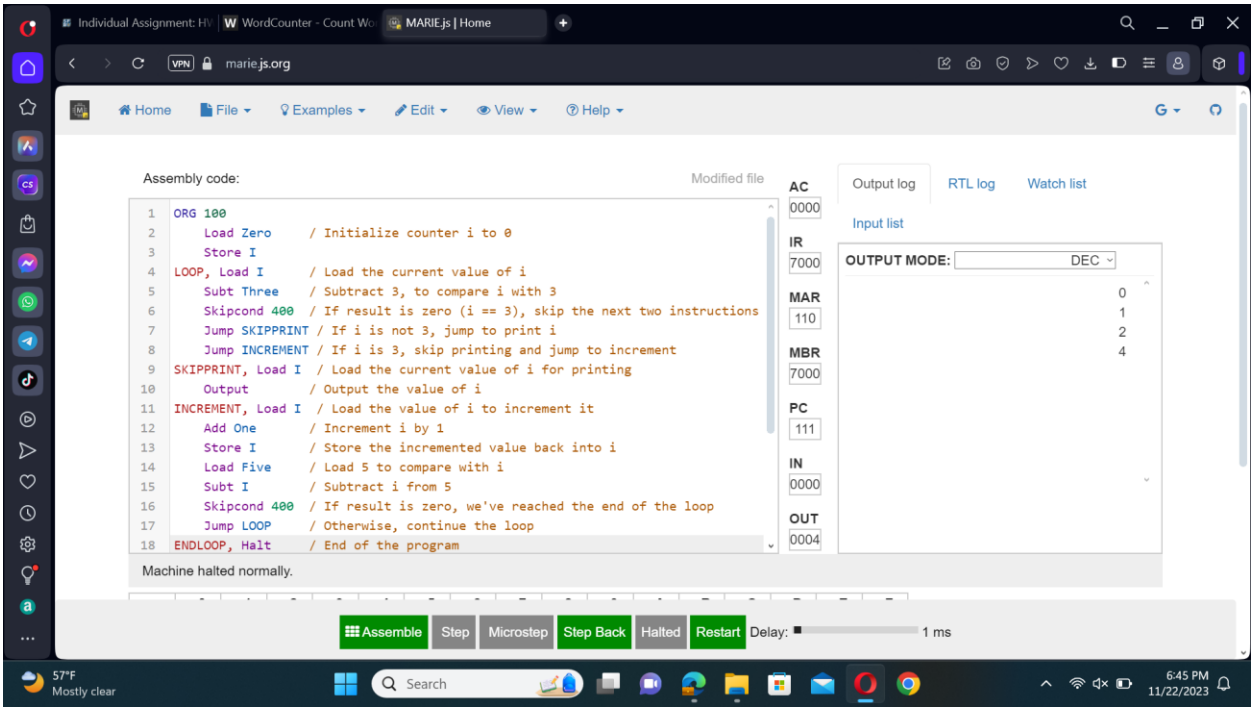
0
1
2
4

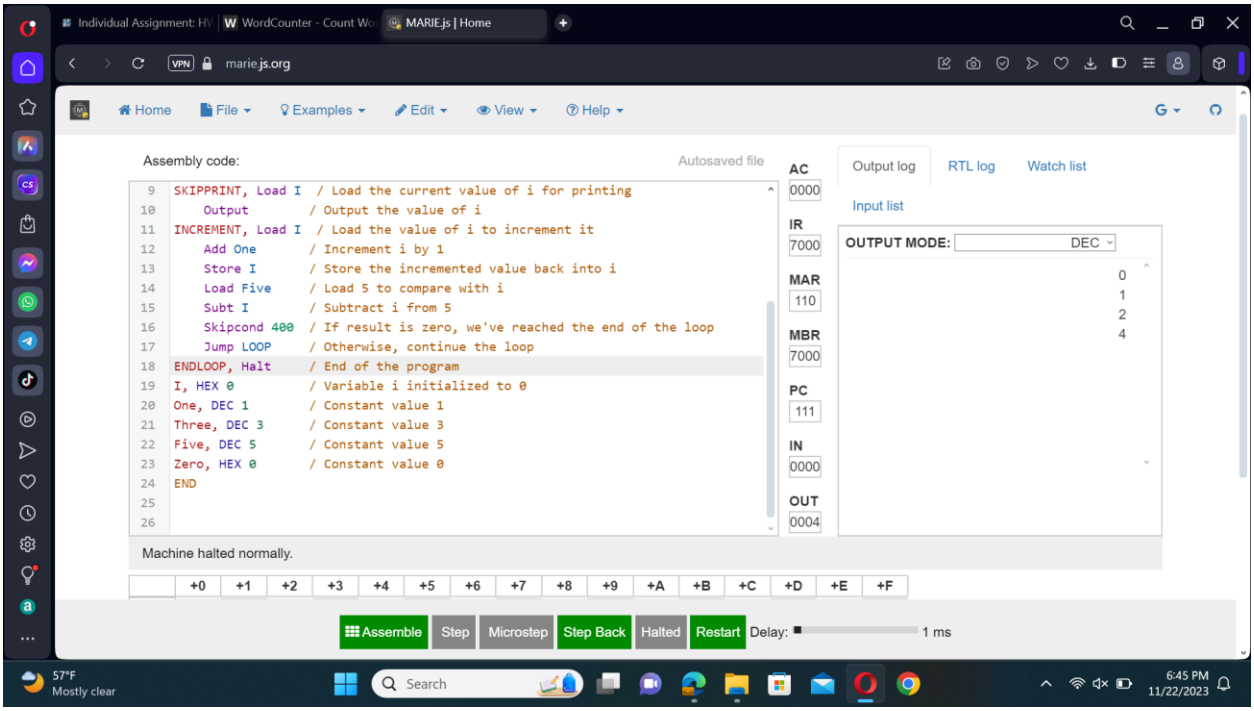
```
ORG 100
    Load Zero     / Initialize counter i to 0
    Store I
LOOP, Load I      / Load the current value of i
    Subt Three    / Subtract 3, to compare i with 3
    Skipcond 400  / If result is zero (i == 3), skip the next two instructions
    Jump SKIPPRINT / If i is not 3, jump to print i
    Jump INCREMENT / If i is 3, skip printing and jump to increment
SKIPPRINT, Load I  / Load the current value of i for printing
    Output        / Output the value of i
INCREMENT, Load I  / Load the value of i to increment it
    Add One       / Increment i by 1
    Store I       / Store the incremented value back into i
    Load Five     / Load 5 to compare with i
    Subt I        / Subtract i from 5
    Skipcond 400  / If result is zero, we've reached the end of the loop
    Jump LOOP     / Otherwise, continue the loop
ENDLOOP, Halt     / End of the program
I, HEX 0          / Variable i initialized to 0
One, DEC 1        / Constant value 1
Three, DEC 3      / Constant value 3
```

Five, DEC 5 / Constant value 5

Zero, HEX 0 / Constant value 0

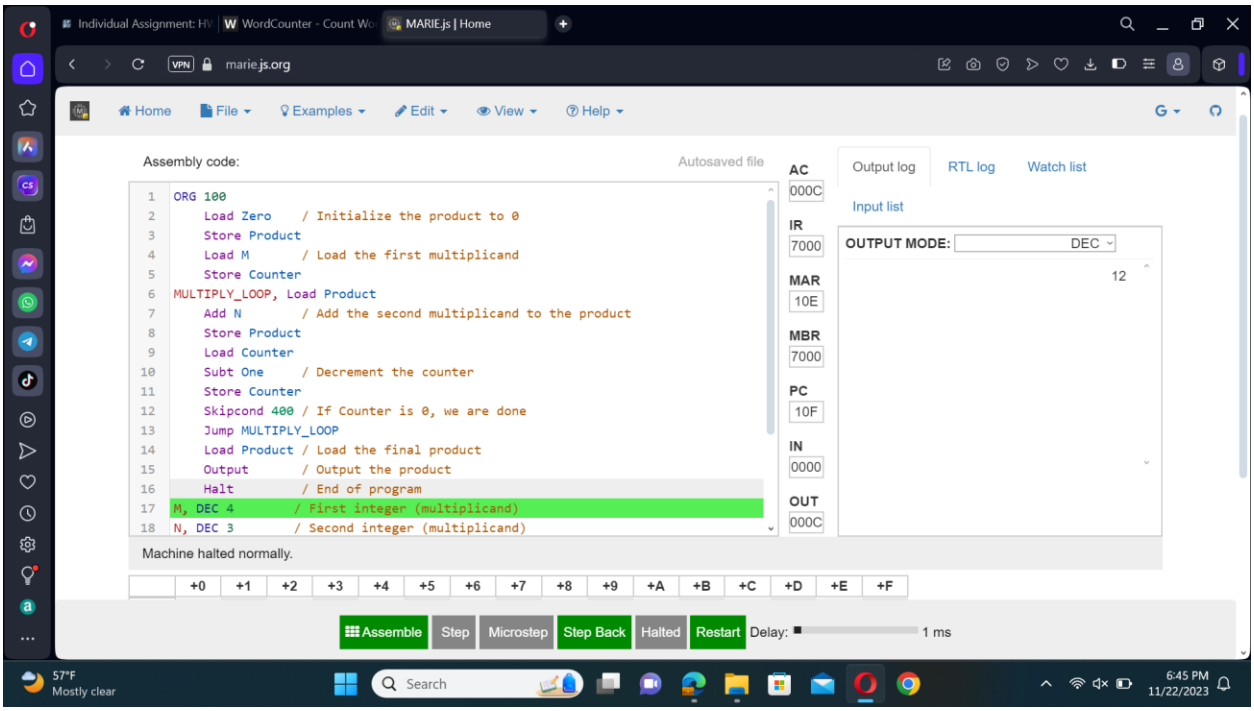
END

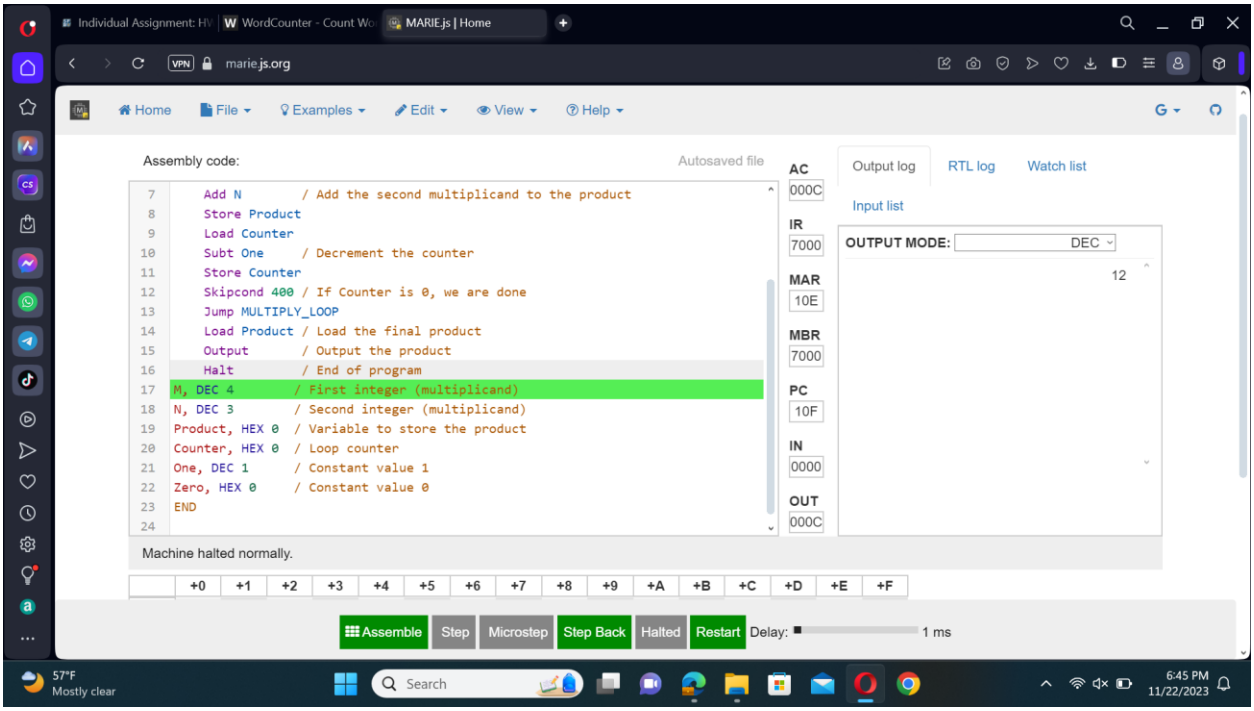




4. Since there is not a multiplication instruction in ISA of MARIE, two integers multiplication operation, for instance, 4×3, must be done by the addition operation, like 4×3 = 4 + 4 + 4. Write the MARIE assembly program to find the product of two integers $m \times n$.

```
ORG 100
    Load Zero    / Initialize the product to 0
    Store Product
    Load M       / Load the first multiplicand
    Store Counter
MULTIPLY_LOOP, Load Product
    Add N        / Add the second multiplicand to the product
    Store Product
    Load Counter
    Subt One     / Decrement the counter
    Store Counter
    Skipcond 400 / If Counter is 0, we are done
    Jump MULTIPLY_LOOP
    Load Product / Load the final product
    Output       / Output the product
    Halt         / End of program
M, DEC 4         / First integer (multiplicand)
N, DEC 3         / Second integer (multiplicand)
Product, HEX 0   / Variable to store the product
Counter, HEX 0   / Loop counter
One, DEC 1       / Constant value 1
Zero, HEX 0      / Constant value 0
END
```


Individual Assignment: HV
WordCounter - Count Wo
MARIE.js | Home
marie.js.org
Home
File
Examples
Edit
View
Help
Assembly code:
Autosaved file
ORG 100
Load Zero / Initialize the product to 0
Store Product
Load M / Load the first multiplicand
Store Counter
MULTIPLY_LOOP, Load Product
Add N / Add the second multiplicand to the product
Store Product
Load Counter
Subt One / Decrement the counter
Store Counter
Skipcond 400 / If Counter is 0, we are done
Jump MULTIPLY_LOOP
Load Product / Load the final product
Output / Output the product
Halt / End of program
M, DEC 4 / First integer (multiplicand)
N, DEC 3 / Second integer (multiplicand)
AC 000C
IR 7000
MAR 10E
MBR 7000
PC 10F
IN 0000
OUT 000C
Output log
RTL log
Watch list
Input list
OUTPUT MODE: DEC
12
Machine halted normally.
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
Assemble
Step
Microstep
Step Back
Halted
Restart
Delay:
1 ms
57°F
Mostly clear
Search
6:45 PM
11/22/2023


Individual Assignment: HV
WordCounter - Count Wo
MARIE.js | Home
marie.js.org
Home
File
Examples
Edit
View
Help
Assembly code:
Autosaved file
Add N / Add the second multiplicand to the product
Store Product
Load Counter
Subt One / Decrement the counter
Store Counter
Skipcond 400 / If Counter is 0, we are done
Jump MULTIPLY_LOOP
Load Product / Load the final product
Output / Output the product
Halt / End of program
M, DEC 4 / First integer (multiplicand)
N, DEC 3 / Second integer (multiplicand)
Product, HEX 0 / Variable to store the product
Counter, HEX 0 / Loop counter
One, DEC 1 / Constant value 1
Zero, HEX 0 / Constant value 0
END
AC 000C
IR 7000
MAR 10E
MBR 7000
PC 10F
IN 0000
OUT 000C
Output log
RTL log
Watch list
Input list
OUTPUT MODE: DEC
12
Machine halted normally.
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
Assemble
Step
Microstep
Step Back
Halted
Restart
Delay:
1 ms
57°F
Mostly clear
Search
6:45 PM
11/22/2023