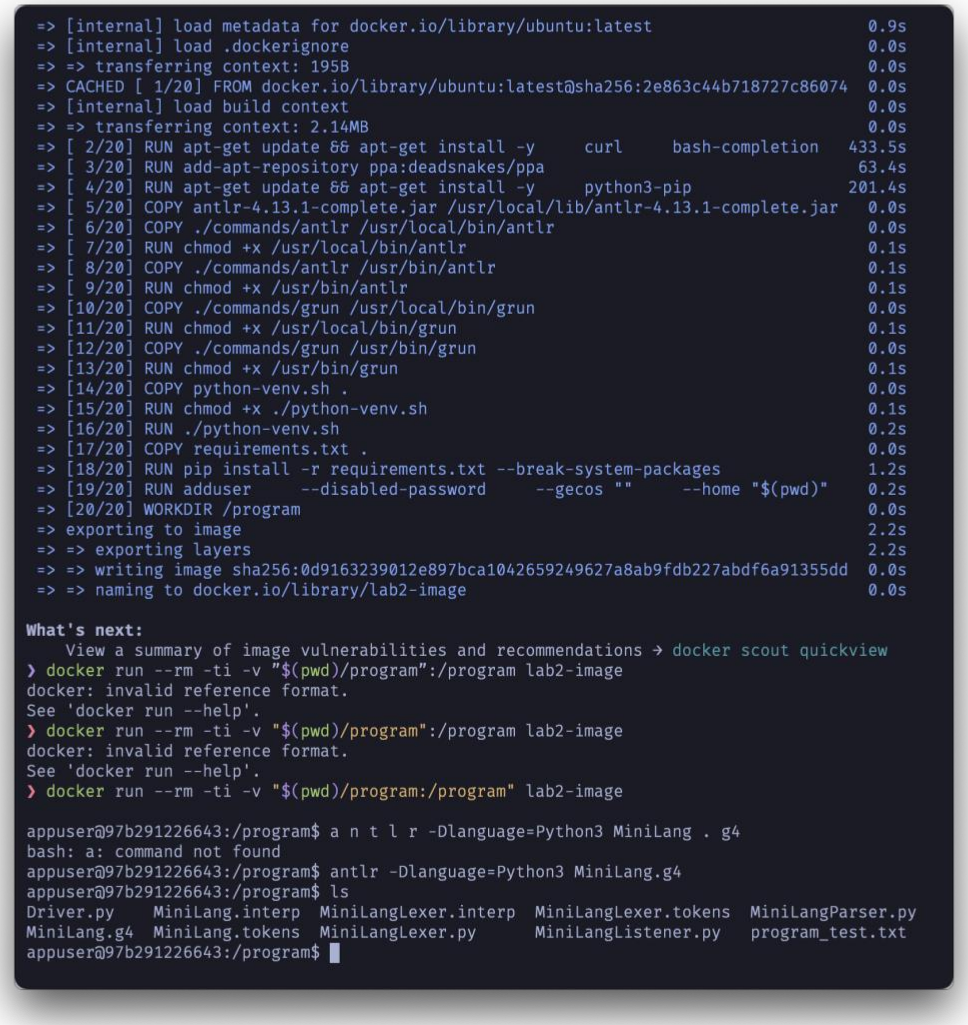
LABORATORIO NO.2

Marco Ramirez 21032 | Josué Morales 21116

Tabla de contenidos

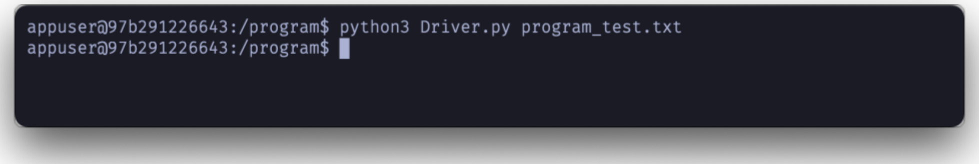
Preparación del ambiente

1. Hacer la construcción y correr el Docker



```
 => [internal] load metadata for docker.io/library/ubuntu:latest                      0.9s
 => [internal] load .dockerignore                                                     0.0s
 => => transferring context: 195B                                                     0.0s
 => CACHED [ 1/20] FROM docker.io/library/ubuntu:latest@sha256:2e863c44b718727c86074  0.0s
 => [internal] load build context                                                     0.0s
 => => transferring context: 2.14MB                                                   0.0s
 => [ 2/20] RUN apt-get update && apt-get install -y     curl     bash-completion   433.5s
 => [ 3/20] RUN add-apt-repository ppa:deadsnakes/ppa                                63.4s
 => [ 4/20] RUN apt-get update && apt-get install -y     python3-pip                201.4s
 => [ 5/20] COPY antlr-4.13.1-complete.jar /usr/local/lib/antlr-4.13.1-complete.jar   0.0s
 => [ 6/20] COPY ./commands/antlr /usr/local/bin/antlr                                0.0s
 => [ 7/20] RUN chmod +x /usr/local/bin/antlr                                         0.1s
 => [ 8/20] COPY ./commands/antlr /usr/bin/antlr                                      0.1s
 => [ 9/20] RUN chmod +x /usr/bin/antlr                                               0.1s
 => [10/20] COPY ./commands/grun /usr/local/bin/grun                                  0.0s
 => [11/20] RUN chmod +x /usr/local/bin/grun                                          0.1s
 => [12/20] COPY ./commands/grun /usr/bin/grun                                        0.0s
 => [13/20] RUN chmod +x /usr/bin/grun                                                0.1s
 => [14/20] COPY python-venv.sh .                                                     0.0s
 => [15/20] RUN chmod +x ./python-venv.sh                                             0.1s
 => [16/20] RUN ./python-venv.sh                                                      0.2s
 => [17/20] COPY requirements.txt .                                                   0.0s
 => [18/20] RUN pip install -r requirements.txt --break-system-packages               1.2s
 => [19/20] RUN adduser     --disabled-password     --gecos ""     --home "$(pwd)"    0.2s
 => [20/20] WORKDIR /program                                                          0.0s
 => exporting to image                                                                2.2s
 => => exporting layers                                                               2.2s
 => => writing image sha256:0d9163239012e897bca1042659249627a8ab9fdb227abdf6a91355dd  0.0s
 => => naming to docker.io/library/lab2-image                                         0.0s

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
❯ docker run --rm -ti -v ”$(pwd)/program”:/program lab2-image
docker: invalid reference format.
See 'docker run --help'.
❯ docker run --rm -ti -v "$(pwd)/program":/program lab2-image
docker: invalid reference format.
See 'docker run --help'.
❯ docker run --rm -ti -v "$(pwd)/program:/program" lab2-image

appuser@97b291226643:/program$ a n t l r -Dlanguage=Python3 MiniLang . g4
bash: a: command not found
appuser@97b291226643:/program$ antlr -Dlanguage=Python3 MiniLang.g4
appuser@97b291226643:/program$ ls
Driver.py    MiniLang.interp  MiniLangLexer.interp  MiniLangLexer.tokens  MiniLangParser.py
MiniLang.g4  MiniLang.tokens  MiniLangLexer.py      MiniLangListener.py   program_test.txt
appuser@97b291226643:/program$ █
```

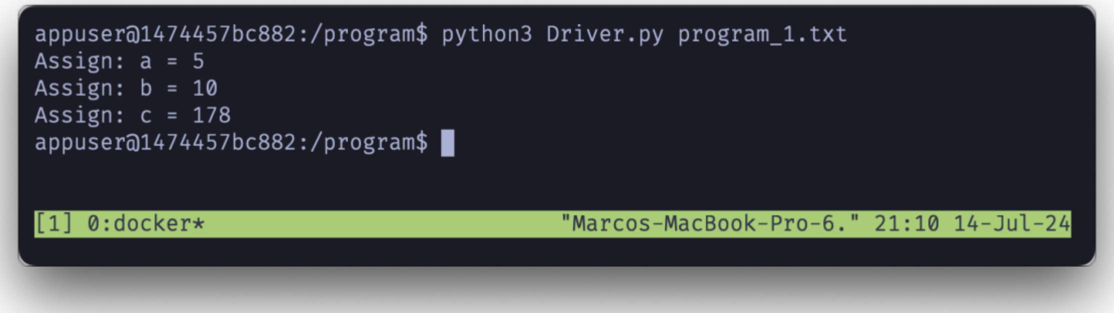
2. Correr el programa



```
appuser@97b291226643:/program$ python3 Driver.py program_test.txt
appuser@97b291226643:/program$ █
```

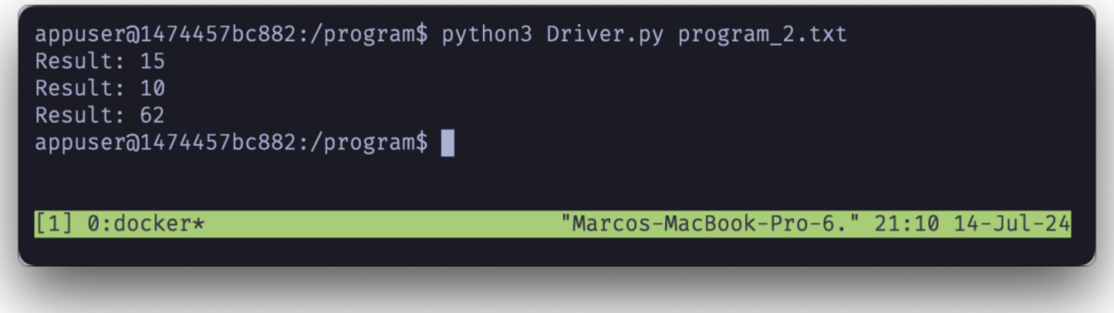
Entregables

1. Programa que asigne un valor a una variable



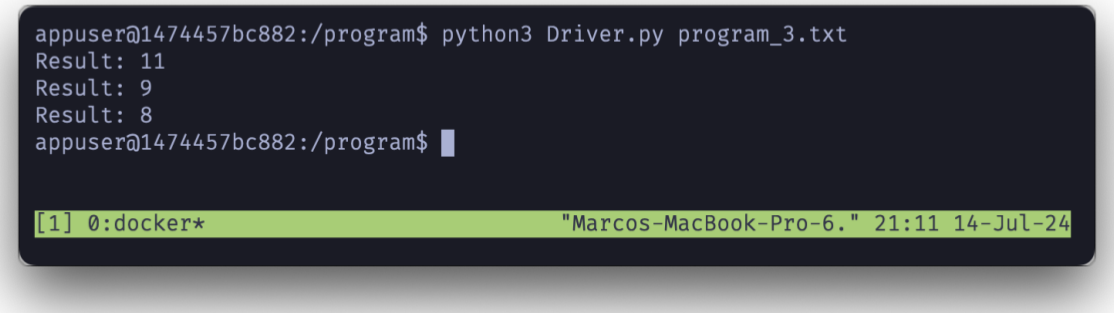
```
appuser@1474457bc882:/program$ python3 Driver.py program_1.txt
Assign: a = 5
Assign: b = 10
Assign: c = 178
appuser@1474457bc882:/program$ █


[1] 0:docker*                            "Marcos-MacBook-Pro-6." 21:10 14-Jul-24
```

2. Programa con expresión básica



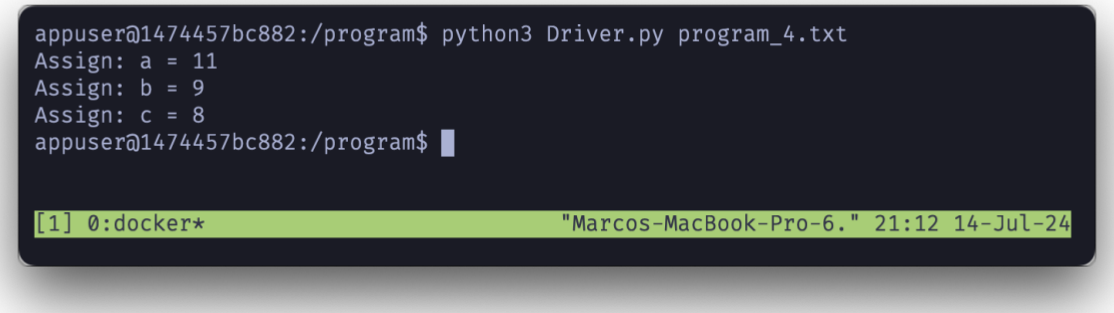
```
appuser@1474457bc882:/program$ python3 Driver.py program_2.txt
Result: 15
Result: 10
Result: 62
appuser@1474457bc882:/program$ █


[1] 0:docker*                            "Marcos-MacBook-Pro-6." 21:10 14-Jul-24
```

3. Expresiones complejas



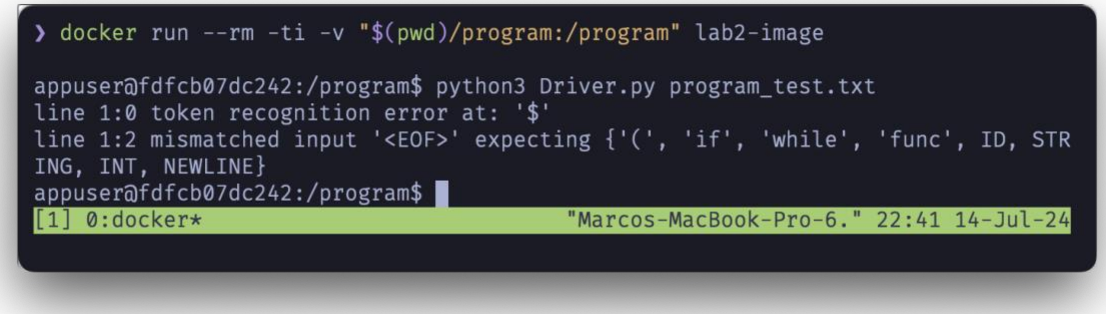
```
appuser@1474457bc882:/program$ python3 Driver.py program_3.txt
Result: 11
Result: 9
Result: 8
appuser@1474457bc882:/program$ 


[1] 0:docker*                          "Marcos-MacBook-Pro-6." 21:11 14-Jul-24
```

4. Incluir la asignación de variables con expresiones aritméticas



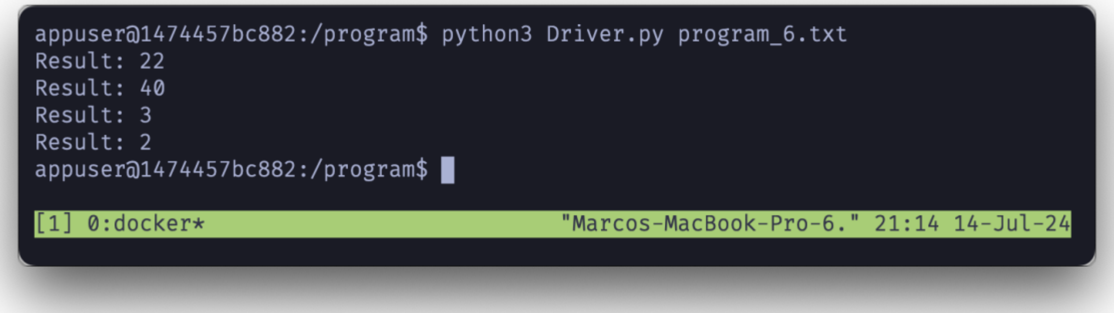
```
appuser@1474457bc882:/program$ python3 Driver.py program_4.txt
Assign: a = 11
Assign: b = 9
Assign: c = 8
appuser@1474457bc882:/program$ 


[1] 0:docker*                          "Marcos-MacBook-Pro-6." 21:12 14-Jul-24
```

5. Manejo de errores léxicos



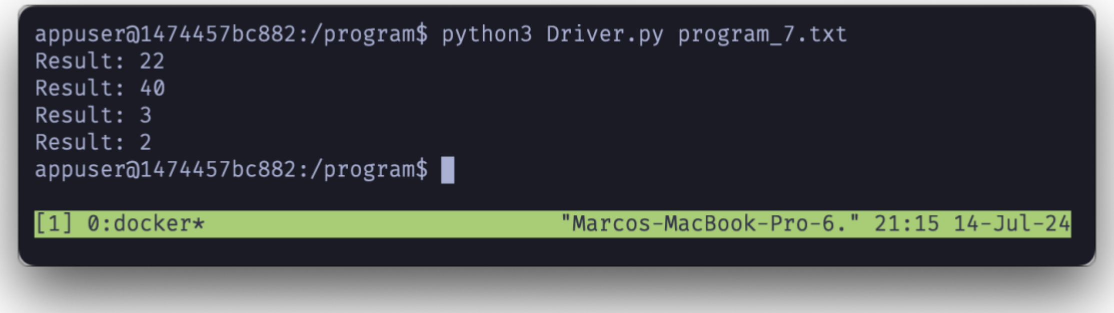
```
❯ docker run --rm -ti -v "$(pwd)/program:/program" lab2-image

appuser@fdfcb07dc242:/program$ python3 Driver.py program_test.txt
line 1:0 token recognition error at: '$'
line 1:2 mismatched input '<EOF>' expecting {'(', 'if', 'while', 'func', ID, STR
ING, INT, NEWLINE}
appuser@fdfcb07dc242:/program$ 
[1] 0:docker*                           "Marcos-MacBook-Pro-6." 22:41 14-Jul-24
```

6. Programa con paréntesis y cambio de precedencia en operadores



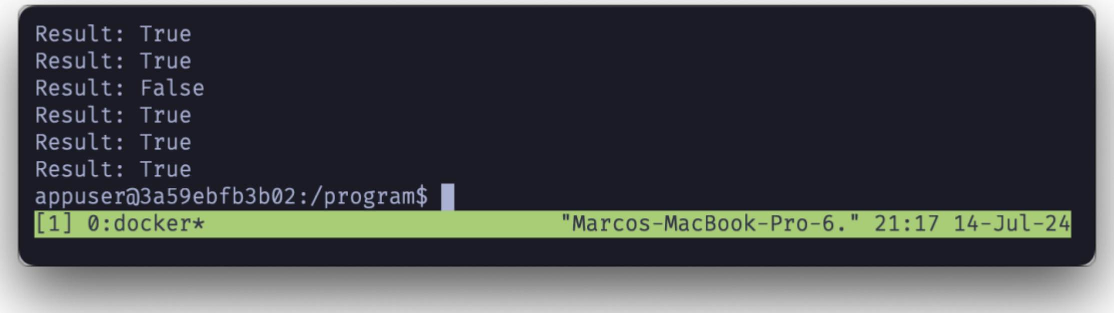
```
appuser@1474457bc882:/program$ python3 Driver.py program_6.txt
Result: 22
Result: 40
Result: 3
Result: 2
appuser@1474457bc882:/program$ █

[1] 0:docker*                           "Marcos-MacBook-Pro-6." 21:14 14-Jul-24
```

7. Programa con comentarios de una sola línea



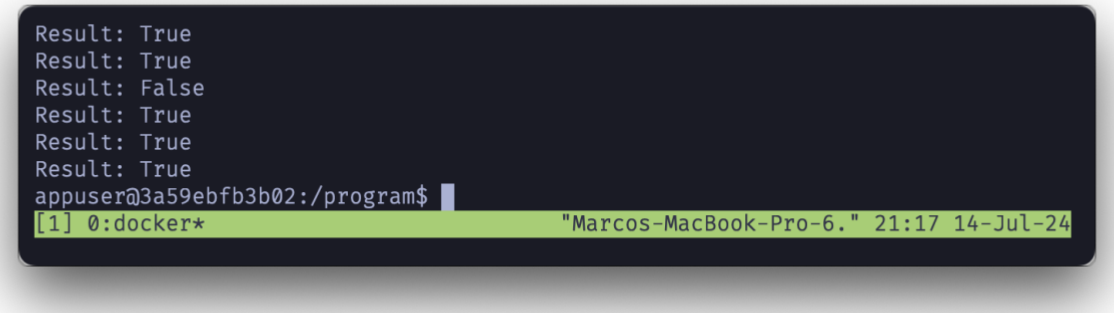
```
appuser@1474457bc882:/program$ python3 Driver.py program_7.txt
Result: 22
Result: 40
Result: 3
Result: 2
appuser@1474457bc882:/program$ █

[1] 0:docker*                           "Marcos-MacBook-Pro-6." 21:15 14-Jul-24
```

8. Programa con operadores de comparación



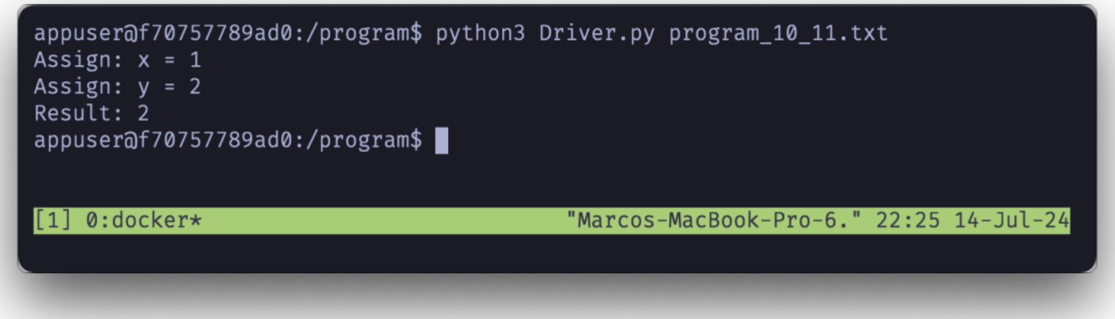
```
Result: True
Result: True
Result: False
Result: True
Result: True
Result: True
appuser@3a59ebfb3b02:/program$ █
[1] 0:docker*                           "Marcos-MacBook-Pro-6." 21:17 14-Jul-24
```

9. Programa experimentando con operadores de comparación



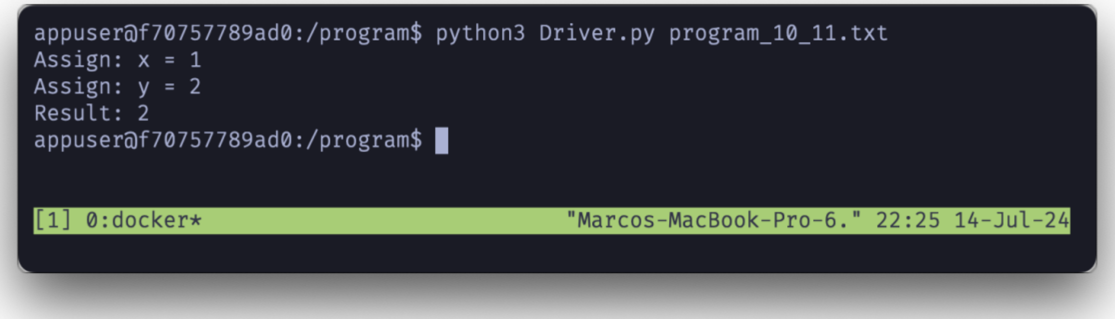
```
Result: True
Result: True
Result: False
Result: True
Result: True
Result: True
appuser@3a59ebfb3b02:/program$ 
[1] 0:docker*                                        "Marcos-MacBook-Pro-6." 21:17 14-Jul-24
```

10. Estructuras de control como “if” y “while”



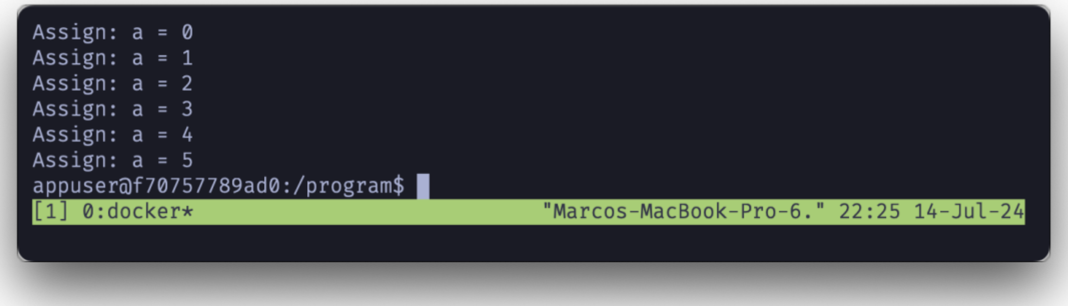
```
appuser@f70757789ad0:/program$ python3 Driver.py program_10_11.txt
Assign: x = 1
Assign: y = 2
Result: 2
appuser@f70757789ad0:/program$ 


[1] 0:docker*                                        "Marcos-MacBook-Pro-6." 22:25 14-Jul-24
```

11. Programa que utilice estructura “if”



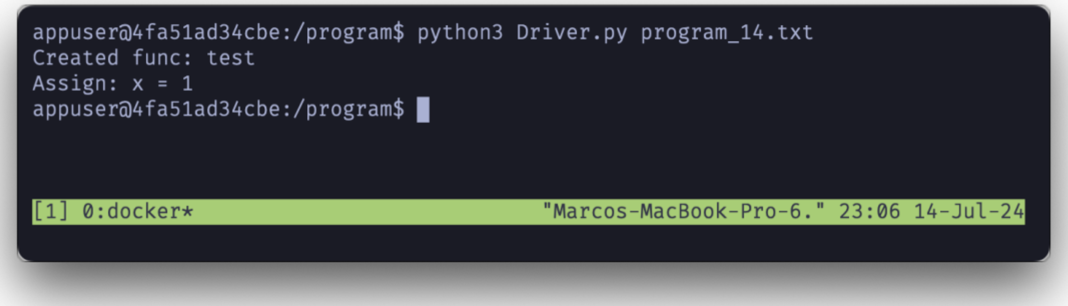
```
appuser@f70757789ad0:/program$ python3 Driver.py program_10_11.txt
Assign: x = 1
Assign: y = 2
Result: 2
appuser@f70757789ad0:/program$ 


[1] 0:docker*                                        "Marcos-MacBook-Pro-6." 22:25 14-Jul-24
```

12. Programa que utilice estrcutra “while”



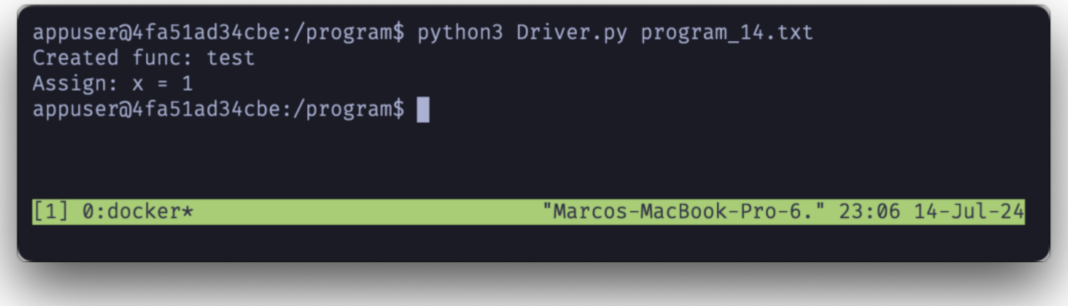
```
Assign: a = 0
Assign: a = 1
Assign: a = 2
Assign: a = 3
Assign: a = 4
Assign: a = 5
appuser@f70757789ad0:/program$ 
[1] 0:docker*                          "Marcos-MacBook-Pro-6." 22:25 14-Jul-24
```

13. Soporte de funciones definidas por el usuario



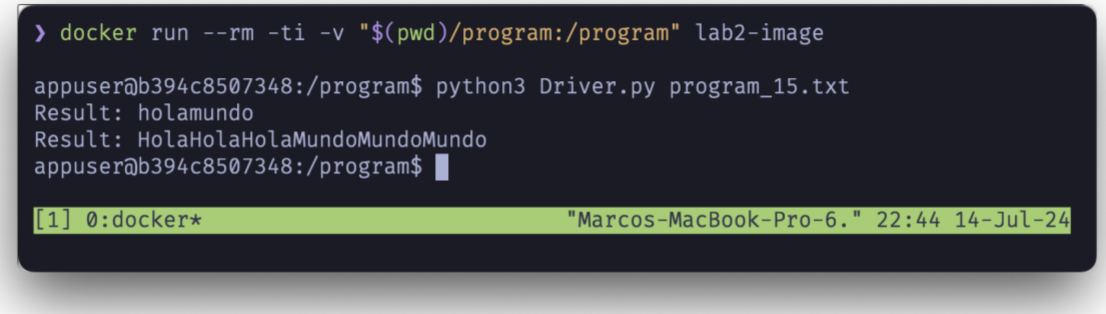
```
appuser@4fa51ad34cbe:/program$ python3 Driver.py program_14.txt
Created func: test
Assign: x = 1
appuser@4fa51ad34cbe:/program$ 



[1] 0:docker*                          "Marcos-MacBook-Pro-6." 23:06 14-Jul-24
```

14. Programa que defina y llame una función



```
appuser@4fa51ad34cbe:/program$ python3 Driver.py program_14.txt
Created func: test
Assign: x = 1
appuser@4fa51ad34cbe:/program$ 



[1] 0:docker*                          "Marcos-MacBook-Pro-6." 23:06 14-Jul-24
```

15. Implementación de sistema de tipos



```
❯ docker run --rm -ti -v "$(pwd)/program:/program" lab2-image

appuser@b394c8507348:/program$ python3 Driver.py program_15.txt
Result: holamundo
Result: HolaHolaHolaMundoMundoMundo
appuser@b394c8507348:/program$ 

[1] 0:docker*                               "Marcos-MacBook-Pro-6." 22:44 14-Jul-24
```

Bibliografía

IBM. (2021). Rules to help remove ambiguities. Obtenido dehttps://www.ibm.com/docs/en/zos/2.4.0?topic=ambiguities-rules-help-remove

IBM. (2023). yacc grammar file declarations. Obtenido de https://www.ibm.com/docs/en/aix/7.2?topic=information-yacc-grammar-file-declarations