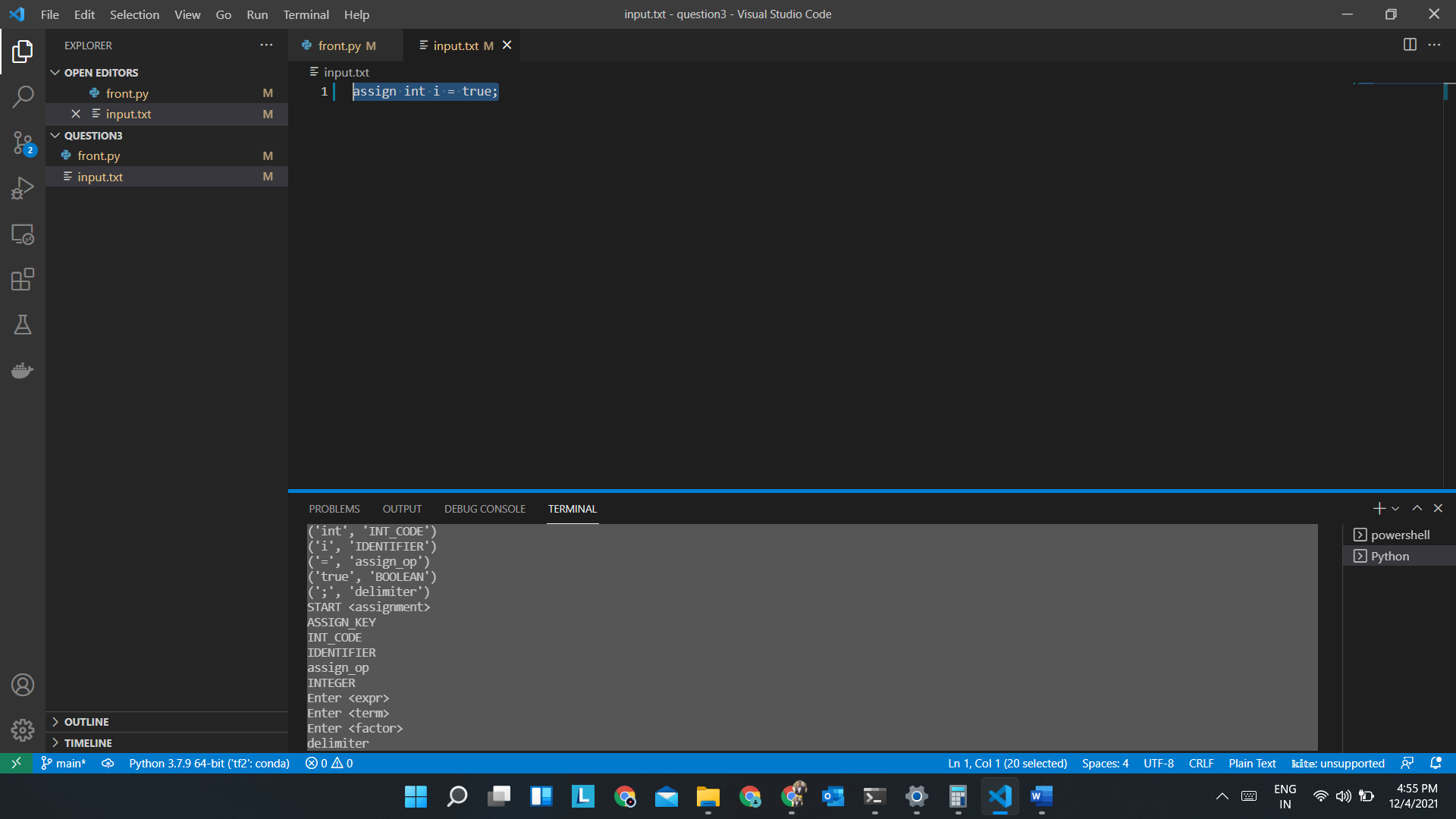
**Question 3 outputs for different scenarios**

3.

**Coercion of Boolean to integers**

Input:

Input.txt



Output:

(tf2) PS D:\OneDrive\Documents\PLC\final assessment\question3> python .\front.py

('assign', 'ASSIGN\_KEY')

('int', 'INT\_CODE')

('i', 'IDENTIFIER')

('=', 'assign\_op')

('true', 'BOOLEAN')

(';', 'delimiter')

START <assignment>

ASSIGN\_KEY

INT\_CODE

IDENTIFIER

assign\_op

INTEGER

Enter <expr>

Enter <term>

Enter <factor>

delimiter

Exit <factor>

Exit <term>

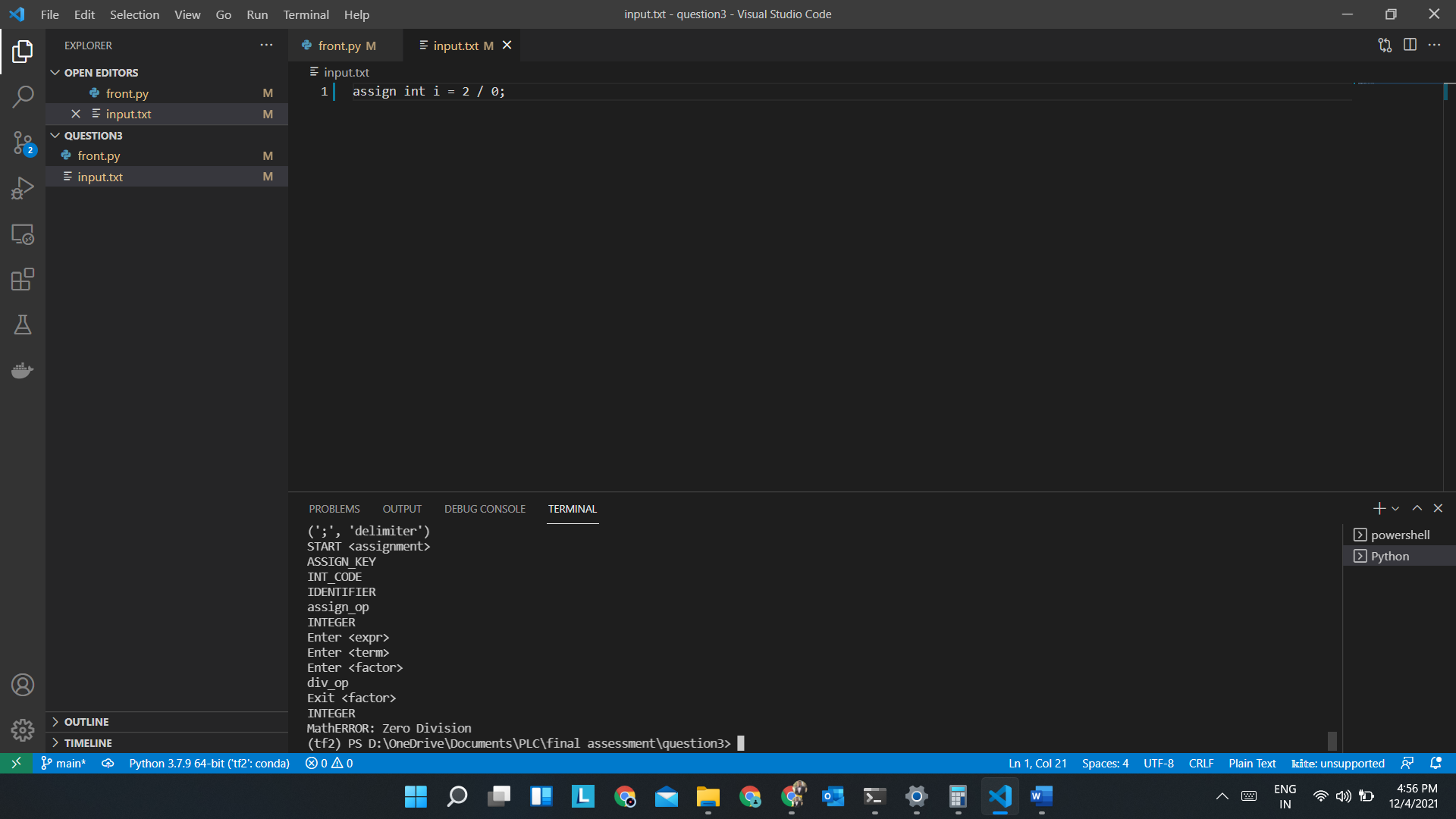
Exit <expr>

the expected type and actual type are matching

END <assignment>

**Division of zero**

Input:



Output:

(tf2) PS D:\OneDrive\Documents\PLC\final assessment\question3> python .\front.py

('assign', 'ASSIGN\_KEY')

('int', 'INT\_CODE')

('i', 'IDENTIFIER')

('=', 'assign\_op')

('2', 'INTEGER')

('/', 'div\_op')

('0', 'INTEGER')

(';', 'delimiter')

START <assignment>

ASSIGN\_KEY

INT\_CODE

IDENTIFIER

assign\_op

INTEGER

Enter <expr>

Enter <term>

Enter <factor>

div\_op

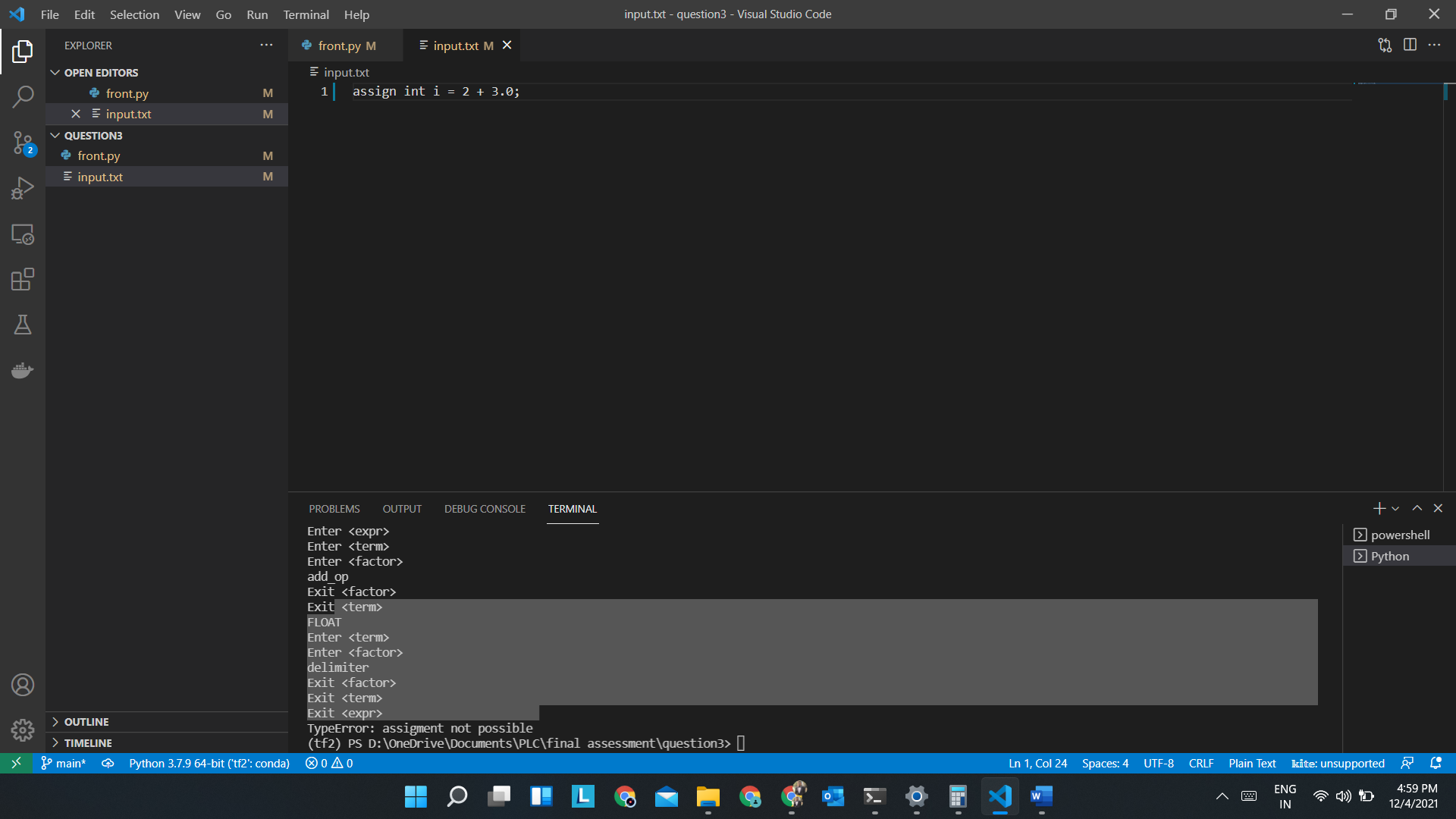
Exit <factor>

INTEGER

MathERROR: Zero Division

**Type mismatch checking**

Input:



Output:

(tf2) PS D:\OneDrive\Documents\PLC\final assessment\question3> python .\front.py

('assign', 'ASSIGN\_KEY')

('int', 'INT\_CODE')

('i', 'IDENTIFIER')

('=', 'assign\_op')

('2', 'INTEGER')

('+', 'add\_op')

('3.0', 'FLOAT')

(';', 'delimiter')

START <assignment>

ASSIGN\_KEY

INT\_CODE

IDENTIFIER

assign\_op

INTEGER

Enter <expr>

Enter <term>

Enter <factor>

add\_op

Exit <factor>

Exit <term>

FLOAT

Enter <term>

Enter <factor>

delimiter

Exit <factor>

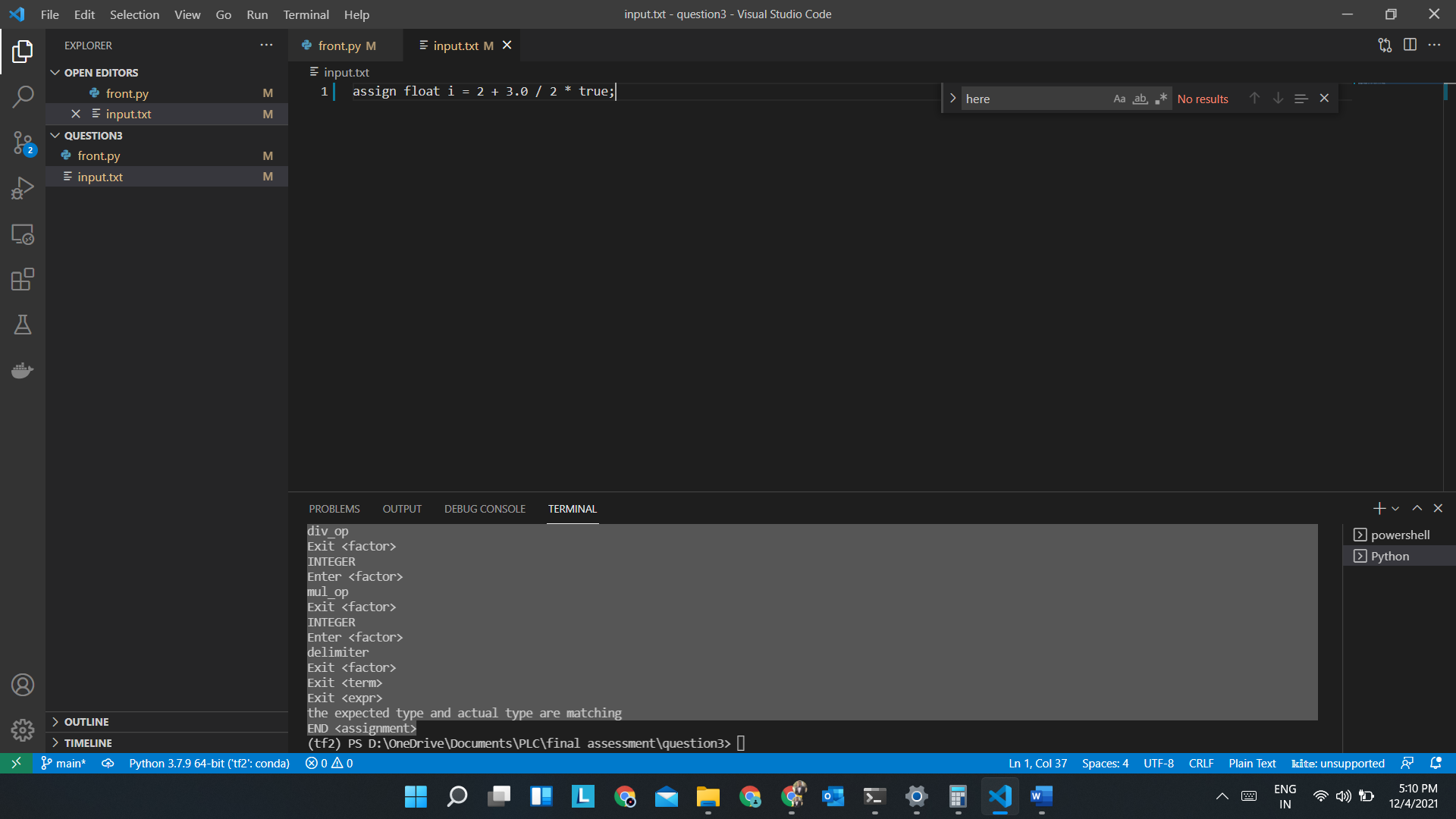
Exit <term>

Exit <expr>

TypeError: assignment not possible

**Positive case and type casting**

Input:



Output:

(tf2) PS D:\OneDrive\Documents\PLC\final assessment\question3> python .\front.py

('assign', 'ASSIGN\_KEY')

('float', 'FLOAT\_CODE')

('i', 'IDENTIFIER')

('=', 'assign\_op')

('2', 'INTEGER')

('+', 'add\_op')

('3.0', 'FLOAT')

('/', 'div\_op')

('2', 'INTEGER')

('\*', 'mul\_op')

('true', 'BOOLEAN')

(';', 'delimiter')

START <assignment>

ASSIGN\_KEY

FLOAT\_CODE

IDENTIFIER

assign\_op

INTEGER

Enter <expr>

Enter <term>

Enter <factor>

add\_op

Exit <factor>

Exit <term>

FLOAT

Enter <term>

Enter <factor>

div\_op

Exit <factor>

INTEGER

Enter <factor>

mul\_op

Exit <factor>

INTEGER

Enter <factor>

delimiter

Exit <factor>

Exit <term>

Exit <expr>

the expected type and actual type are matching

END <assignment>