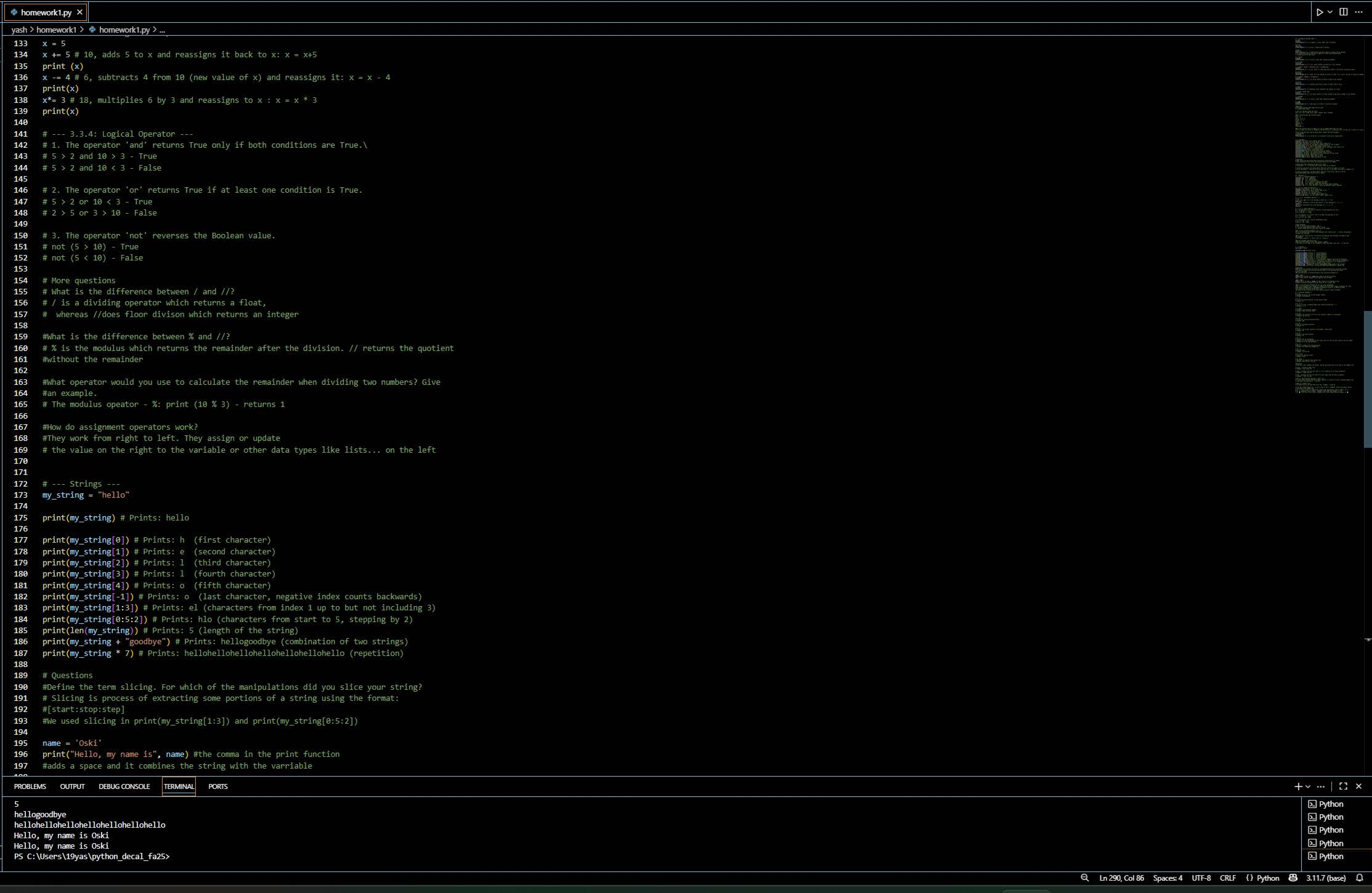
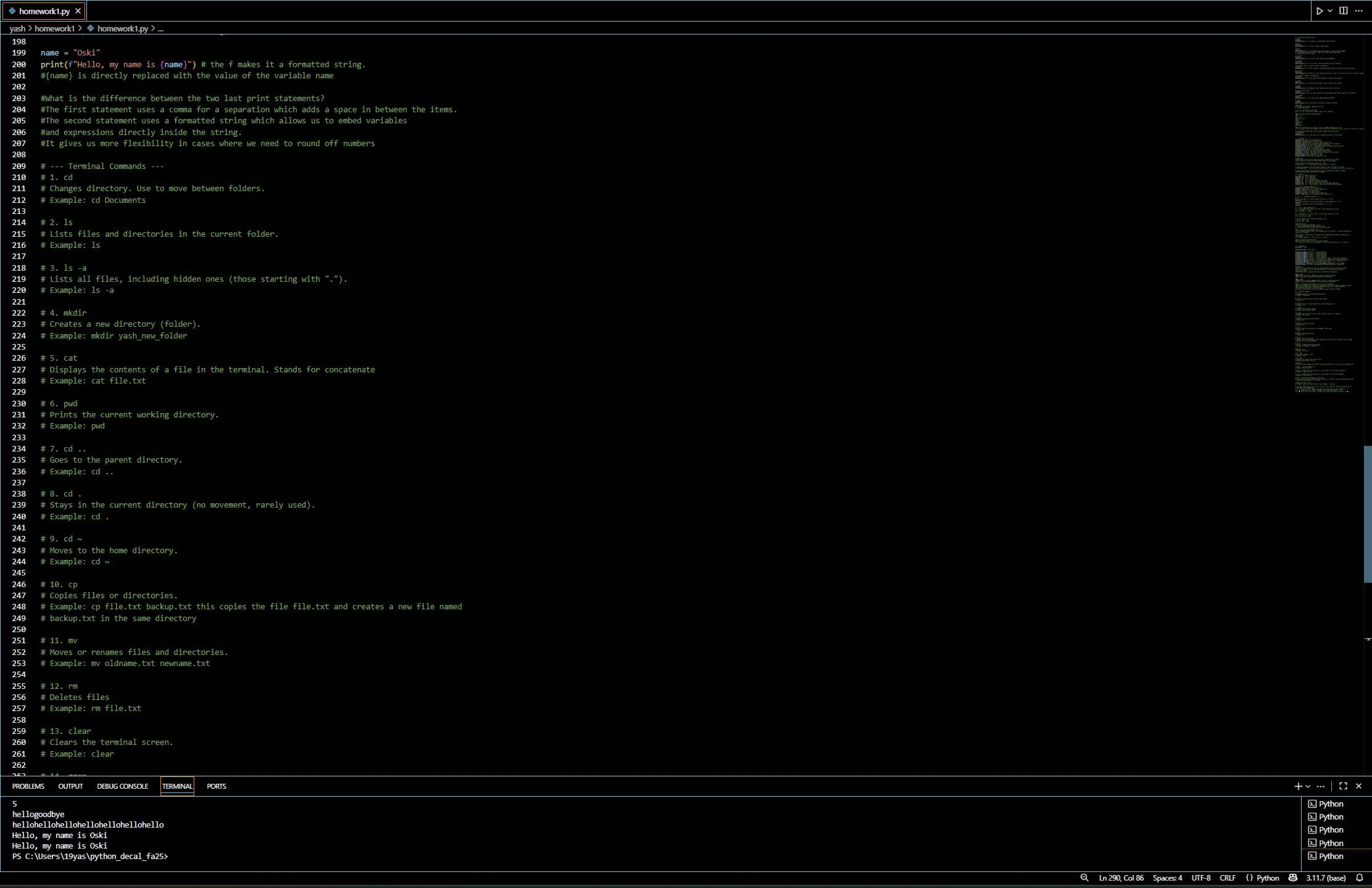


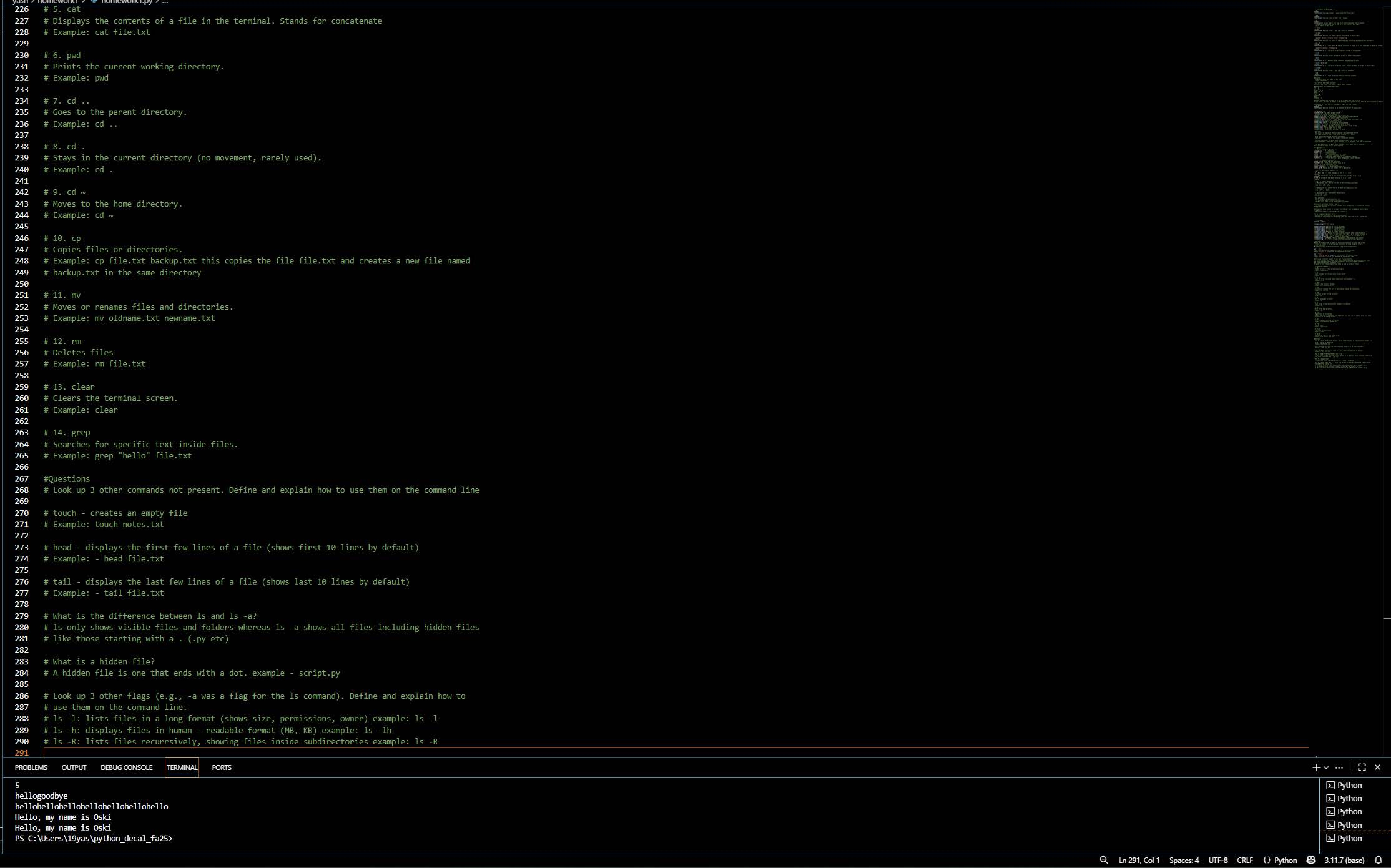
```python
x = 5
x += 5 # 10, adds 5 to x and reassigns it back to x: x = x+5
print (x)
x -= 4 # 6, subtracts 4 from 10 (new value of x) and reassigns it: x = x - 4
print(x)
x*= 3 # 18, multiplies 6 by 3 and reassigns to x : x = x * 3
print(x)

# --- 3.3.4: Logical Operator ---
# 1. The operator 'and' returns True only if both conditions are True.\
# 5 > 2 and 10 > 3 - True
# 5 > 2 and 10 < 3 - False

# 2. The operator 'or' returns True if at least one condition is True.
# 5 > 2 or 10 < 3 - True
# 2 > 5 or 3 > 10 - False

# 3. The operator 'not' reverses the Boolean value.
# not (5 > 10) - True
# not (5 < 10) - False

# More questions
# What is the difference between / and //?
# / is a dividing operator which returns a float,
#  whereas //does floor divison which returns an integer

#What is the difference between % and //?
# % is the modulus which returns the remainder after the division. // returns the quotient
#without the remainder

#What operator would you use to calculate the remainder when dividing two numbers? Give
#an example.
# The modulus opeator - %: print (10 % 3) - returns 1

#How do assignment operators work?
#They work from right to left. They assign or update
# the value on the right to the variable or other data types like lists... on the left


# --- Strings ---
my_string = "hello"

print(my_string) # Prints: hello

print(my_string[0]) # Prints: h  (first character)
print(my_string[1]) # Prints: e  (second character)
print(my_string[2]) # Prints: l  (third character)
print(my_string[3]) # Prints: l  (fourth character)
print(my_string[4]) # Prints: o  (fifth character)
print(my_string[-1]) # Prints: o  (last character, negative index counts backwards)
print(my_string[1:3]) # Prints: el (characters from index 1 up to but not including 3)
print(my_string[0:5:2]) # Prints: hlo (characters from start to 5, stepping by 2)
print(len(my_string)) # Prints: 5 (length of the string)
print(my_string + "goodbye") # Prints: hellogoodbye (combination of two strings)
print(my_string * 7) # Prints: hellohellohellohellohellohellohello (repetition)

# Questions
#Define the term slicing. For which of the manipulations did you slice your string?
# Slicing is process of extracting some portions of a string using the format:
#[start:stop:step]
#We used slicing in print(my_string[1:3]) and print(my_string[0:5:2])

name = 'Oski'
print("Hello, my name is", name) #the comma in the print function
#adds a space and it combines the string with the varriable
```

```
5
hellogoodbye
hellohellohellohellohellohellohello
Hello, my name is Oski
Hello, my name is Oski
PS C:\Users\19yas\python_decal_fa25>
```



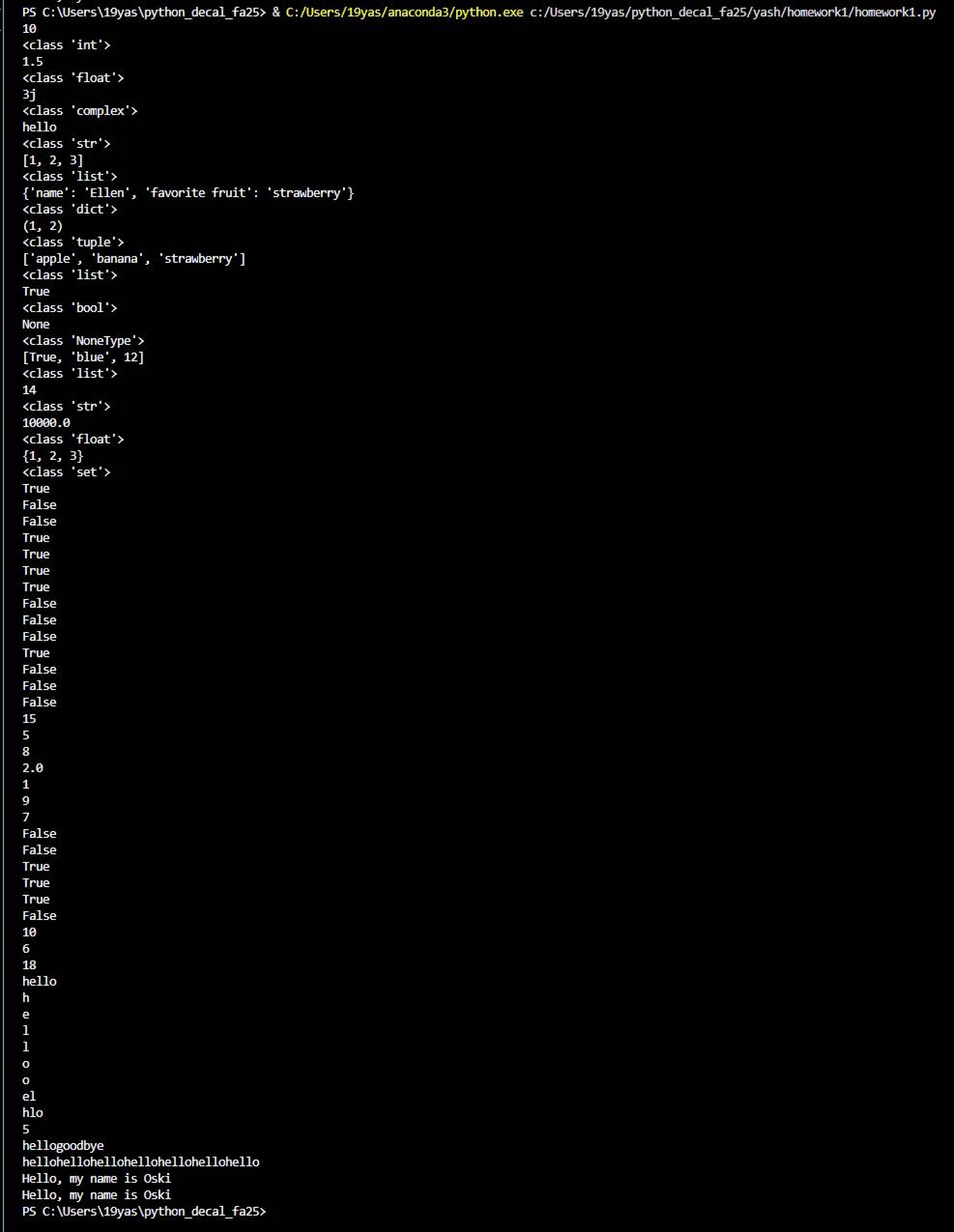
```python

name = "Oski"
print(f"Hello, my name is {name}") # the f makes it a formatted string.
#{name} is directly replaced with the value of the variable name

#What is the difference between the two last print statements?
#The first statement uses a comma for a separation which adds a space in between the items.
#The second statement uses a formatted string which allows us to embed variables
#and expressions directly inside the string.
#It gives us more flexibility in cases where we need to round off numbers

# --- Terminal Commands ---
# 1. cd
# Changes directory. Use to move between folders.
# Example: cd Documents

# 2. ls
# Lists files and directories in the current folder.
# Example: ls

# 3. ls -a
# Lists all files, including hidden ones (those starting with ".").
# Example: ls -a

# 4. mkdir
# Creates a new directory (folder).
# Example: mkdir yash_new_folder

# 5. cat
# Displays the contents of a file in the terminal. Stands for concatenate
# Example: cat file.txt

# 6. pwd
# Prints the current working directory.
# Example: pwd

# 7. cd ..
# Goes to the parent directory.
# Example: cd ..

# 8. cd .
# Stays in the current directory (no movement, rarely used).
# Example: cd .

# 9. cd ~
# Moves to the home directory.
# Example: cd ~

# 10. cp
# Copies files or directories.
# Example: cp file.txt backup.txt this copies the file file.txt and creates a new file named
# backup.txt in the same directory

# 11. mv
# Moves or renames files and directories.
# Example: mv oldname.txt newname.txt

# 12. rm
# Deletes files
# Example: rm file.txt

# 13. clear
# Clears the terminal screen.
# Example: clear

```

```
5
hellogoodbye
hellohellohellohellohellohellohello
Hello, my name is Oski
Hello, my name is Oski
PS C:\Users\19yas\python_decal_fa25>
```



```
# 5. cat
# Displays the contents of a file in the terminal. Stands for concatenate
# Example: cat file.txt

# 6. pwd
# Prints the current working directory.
# Example: pwd

# 7. cd ..
# Goes to the parent directory.
# Example: cd ..

# 8. cd .
# Stays in the current directory (no movement, rarely used).
# Example: cd .

# 9. cd ~
# Moves to the home directory.
# Example: cd ~

# 10. cp
# Copies files or directories.
# Example: cp file.txt backup.txt this copies the file file.txt and creates a new file named
# backup.txt in the same directory

# 11. mv
# Moves or renames files and directories.
# Example: mv oldname.txt newname.txt

# 12. rm
# Deletes files
# Example: rm file.txt

# 13. clear
# Clears the terminal screen.
# Example: clear

# 14. grep
# Searches for specific text inside files.
# Example: grep "hello" file.txt

#Questions
# Look up 3 other commands not present. Define and explain how to use them on the command line

# touch - creates an empty file
# Example: touch notes.txt

# head - displays the first few lines of a file (shows first 10 lines by default)
# Example: - head file.txt

# tail - displays the last few lines of a file (shows last 10 lines by default)
# Example: - tail file.txt

# What is the difference between ls and ls -a?
# ls only shows visible files and folders whereas ls -a shows all files including hidden files
# like those starting with a . (.py etc)

# What is a hidden file?
# A hidden file is one that ends with a dot. example - script.py

# Look up 3 other flags (e.g., -a was a flag for the ls command). Define and explain how to
# use them on the command line.
# ls -l: lists files in a long format (shows size, permissions, owner) example: ls -l
# ls -h: displays files in human - readable format (MB, KB) example: ls -lh
# ls -R: lists files recurrsively, showing files inside subdirectories example: ls -R

```

```
5
hellogoodbye
hellohellohellohellohellohellohello
Hello, my name is Oski
Hello, my name is Oski
PS C:\Users\19yas\python_decal_fa25>
```



```
PS C:\Users\19yas\python_decal_fa25> & C:/Users/19yas/anaconda3/python.exe c:/Users/19yas/python_decal_fa25/yash/homework1/homework1.py
10
<class 'int'>
1.5
<class 'float'>
3j
<class 'complex'>
hello
<class 'str'>
[1, 2, 3]
<class 'list'>
{'name': 'Ellen', 'favorite fruit': 'strawberry'}
<class 'dict'>
(1, 2)
<class 'tuple'>
['apple', 'banana', 'strawberry']
<class 'list'>
True
<class 'bool'>
None
<class 'NoneType'>
[True, 'blue', 12]
<class 'list'>
14
<class 'str'>
10000.0
<class 'float'>
{1, 2, 3}
<class 'set'>
True
False
False
True
True
True
True
False
False
False
True
False
False
False
15
5
8
2.0
1
9
7
False
False
True
True
True
False
10
6
18
hello
h
e
l
l
o
o
el
hlo
5
hellogoodbye
hellohellohellohellohellohellohello
Hello, my name is Oski
Hello, my name is Oski
PS C:\Users\19yas\python_decal_fa25>
```