# Fundamentals of Computer Programming (FOCP)



Submitted by Submitted to

Aarson Subba Krishna Devkota

Section E

Bsc (Computing)

# Introduction to Programming

## Lab Worksheet

Week 2

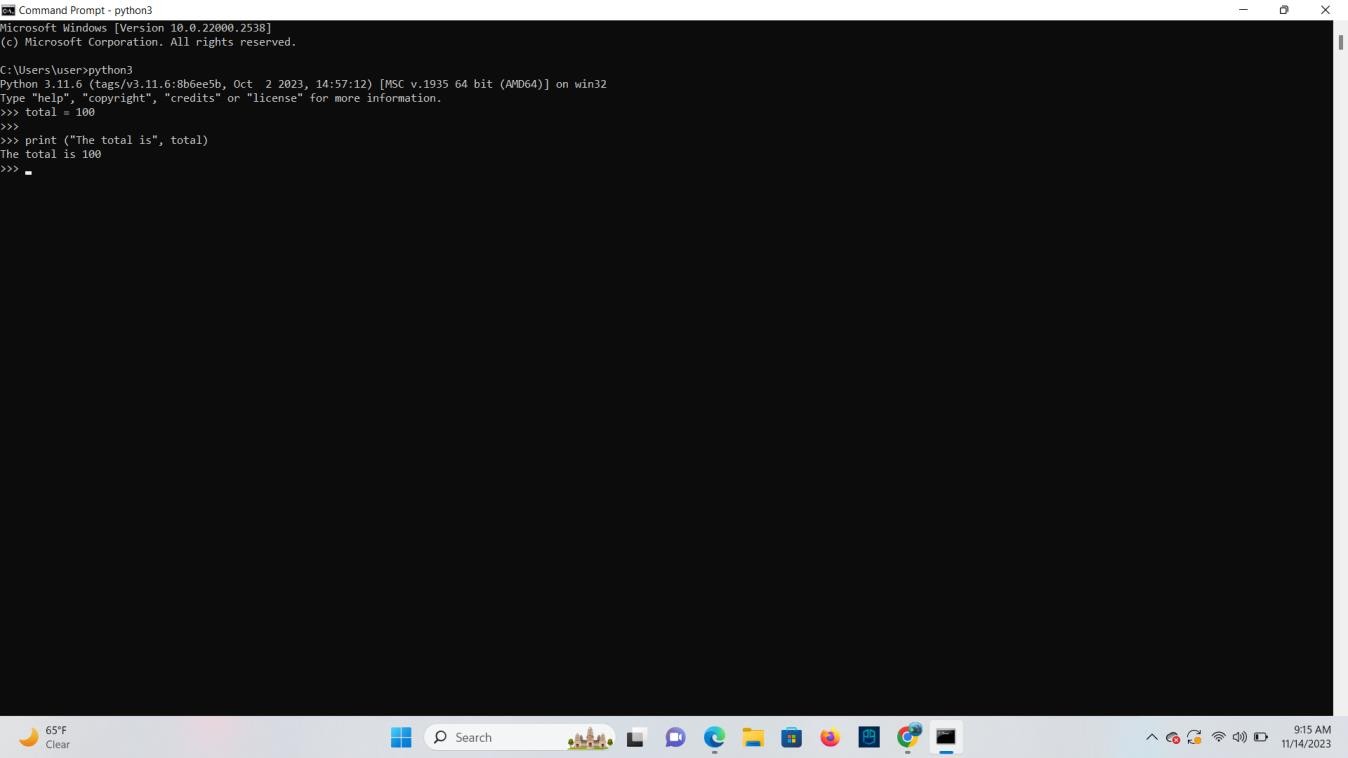
# Working with Variables

## Variable assignment

**TASK**: Try inputting the following code and examine the results. total = 100

Print (“The total is”, total)

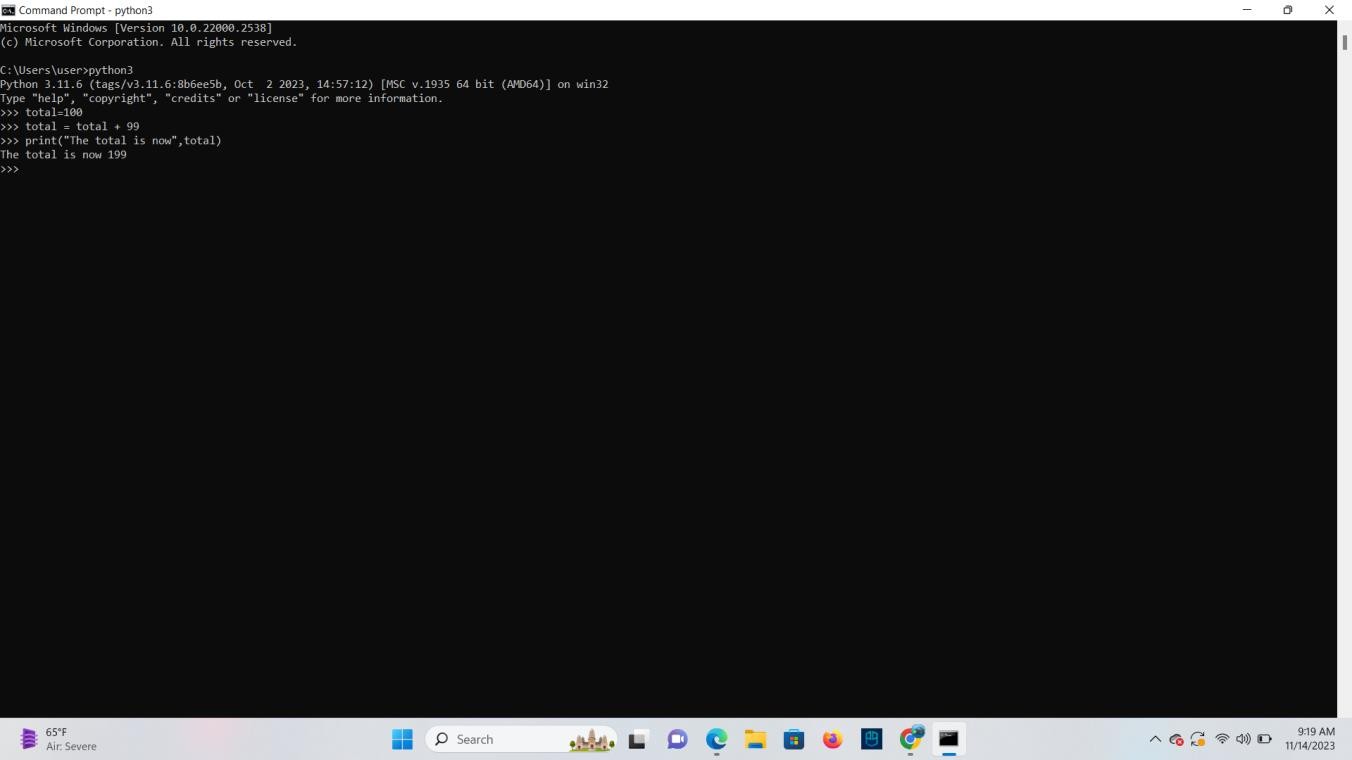
### Ans



**TASK**: Try inputting the following code and examine the results. total = total + 99

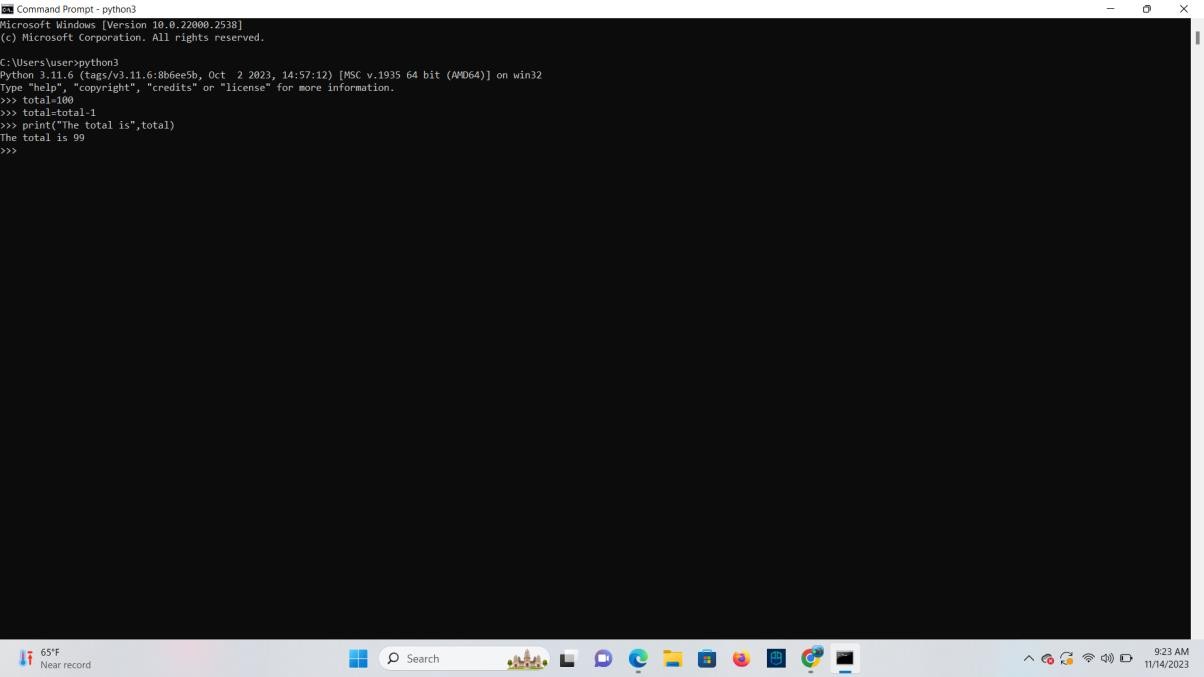
print(“The total is now”, total)

### Ans



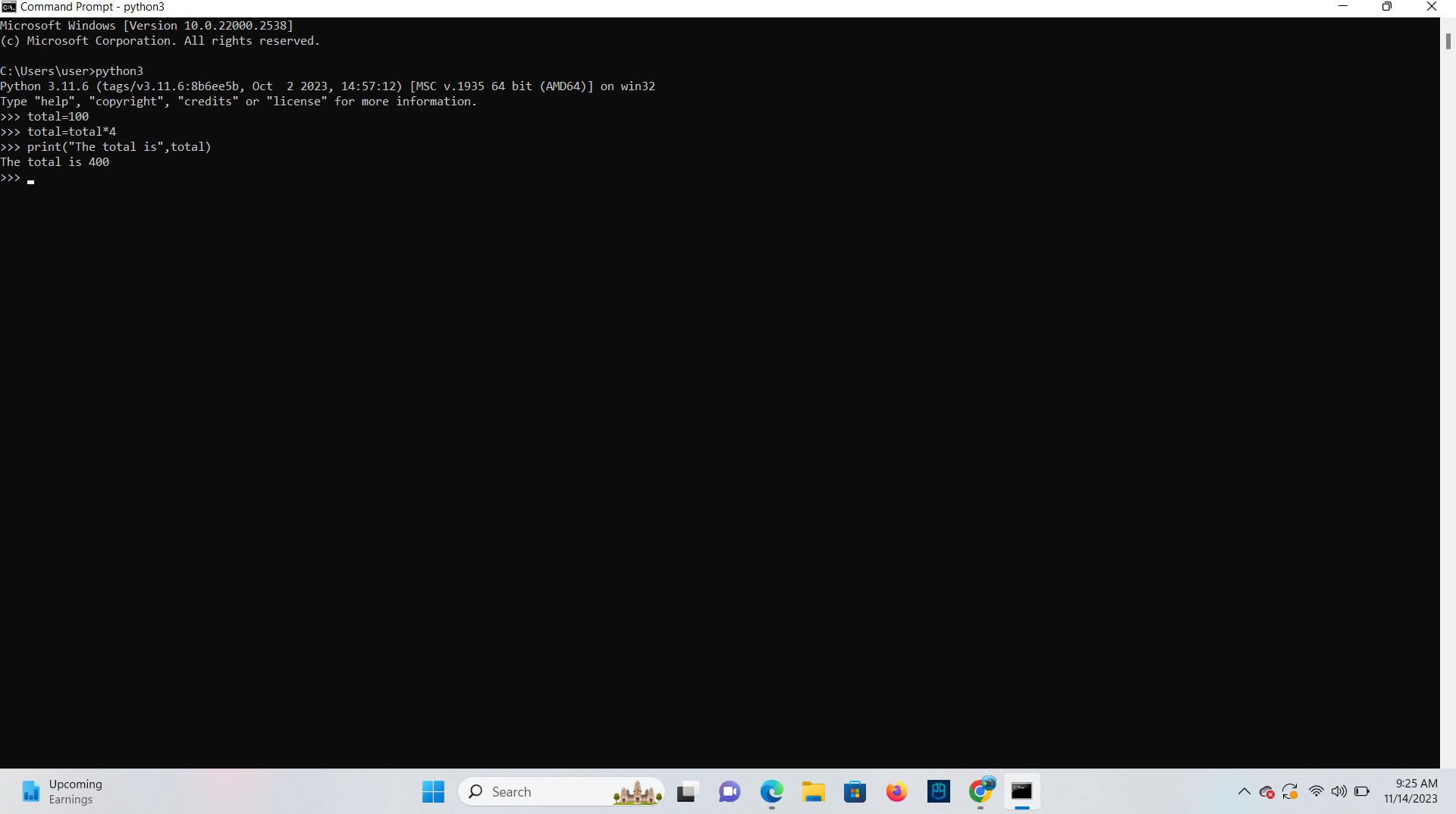
**TASK**: Try inputting the following code, but replace each of the assignment expressions with the equivalent augmented assignment.

total = total - 1 print(“The total is”, total) **Ans**

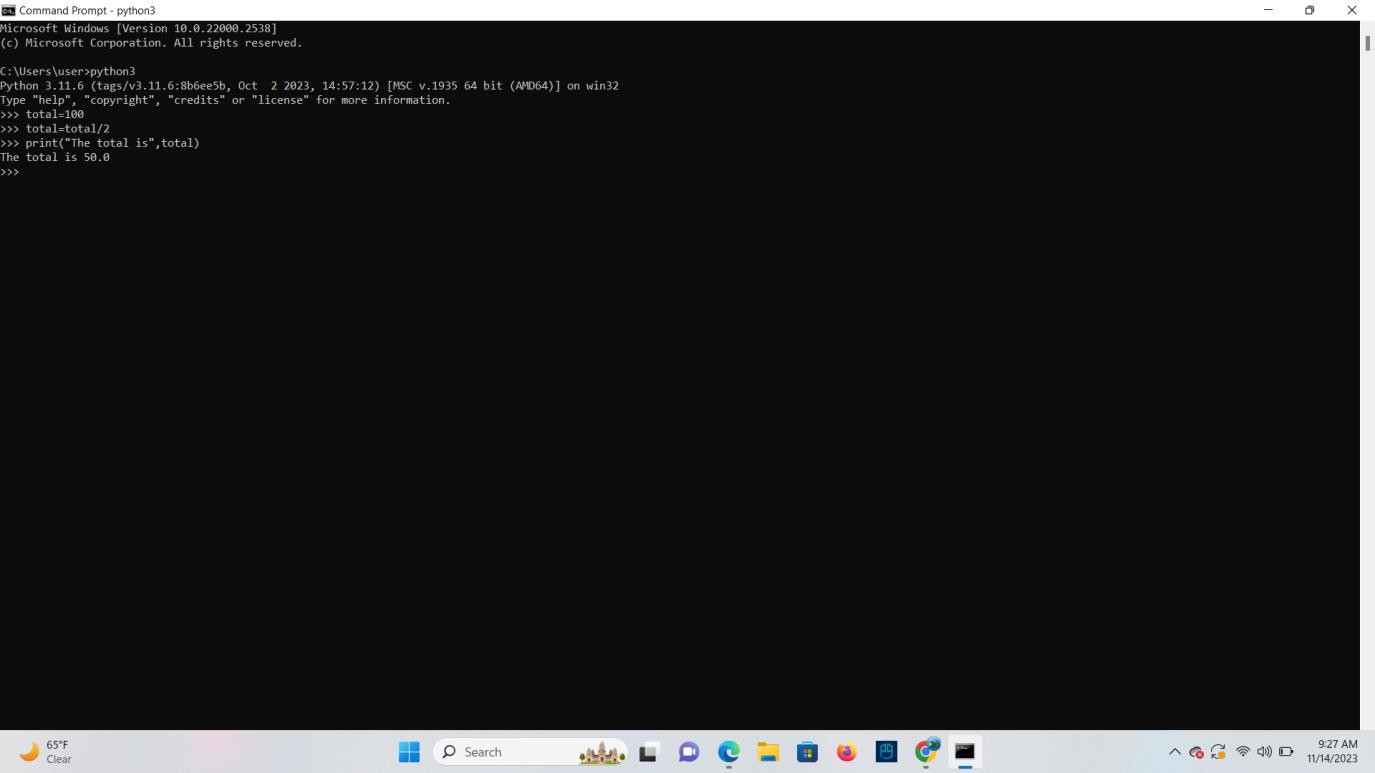


total = total \* 4 print(“The total is”, total)

### Ans



total = total / 2 print(“The total is”, total) **Ans**



**TASK**: Try extending the code below so that it creates a new variable called ‘average’, that is set to equal the average calculated from the two other variables.

total = 98.2

count = 5

# add your extra code here

### Ans

total = 98.2

count = 5

Average = total/count

Print(“The average is:”, Average)

# Data-Types

## A Variable’s data-type

**TASK**: Use the type() function to determine the type of each of the following values. False

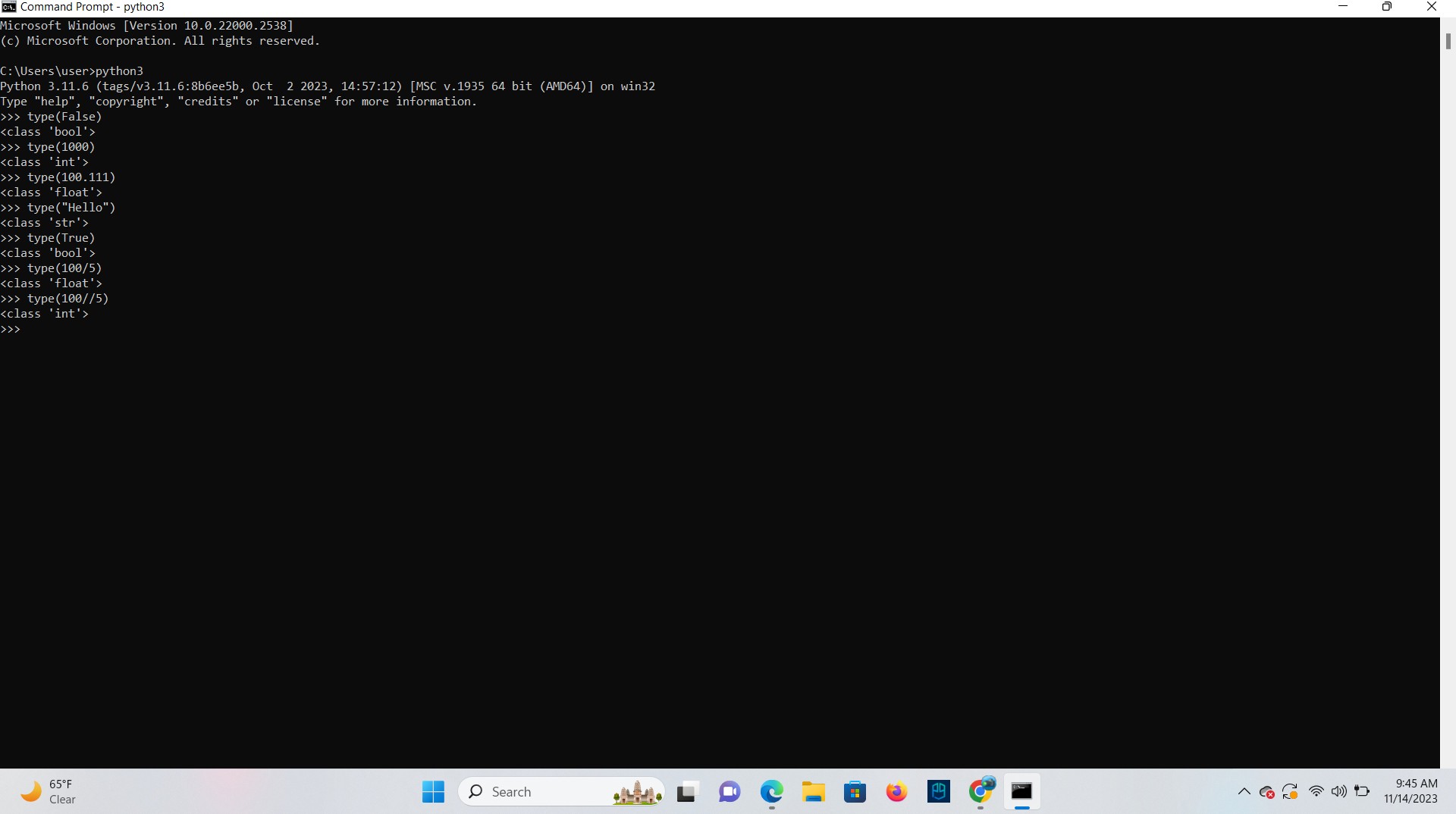
1000

100.111

“Hello” True 100 / 5

100//5

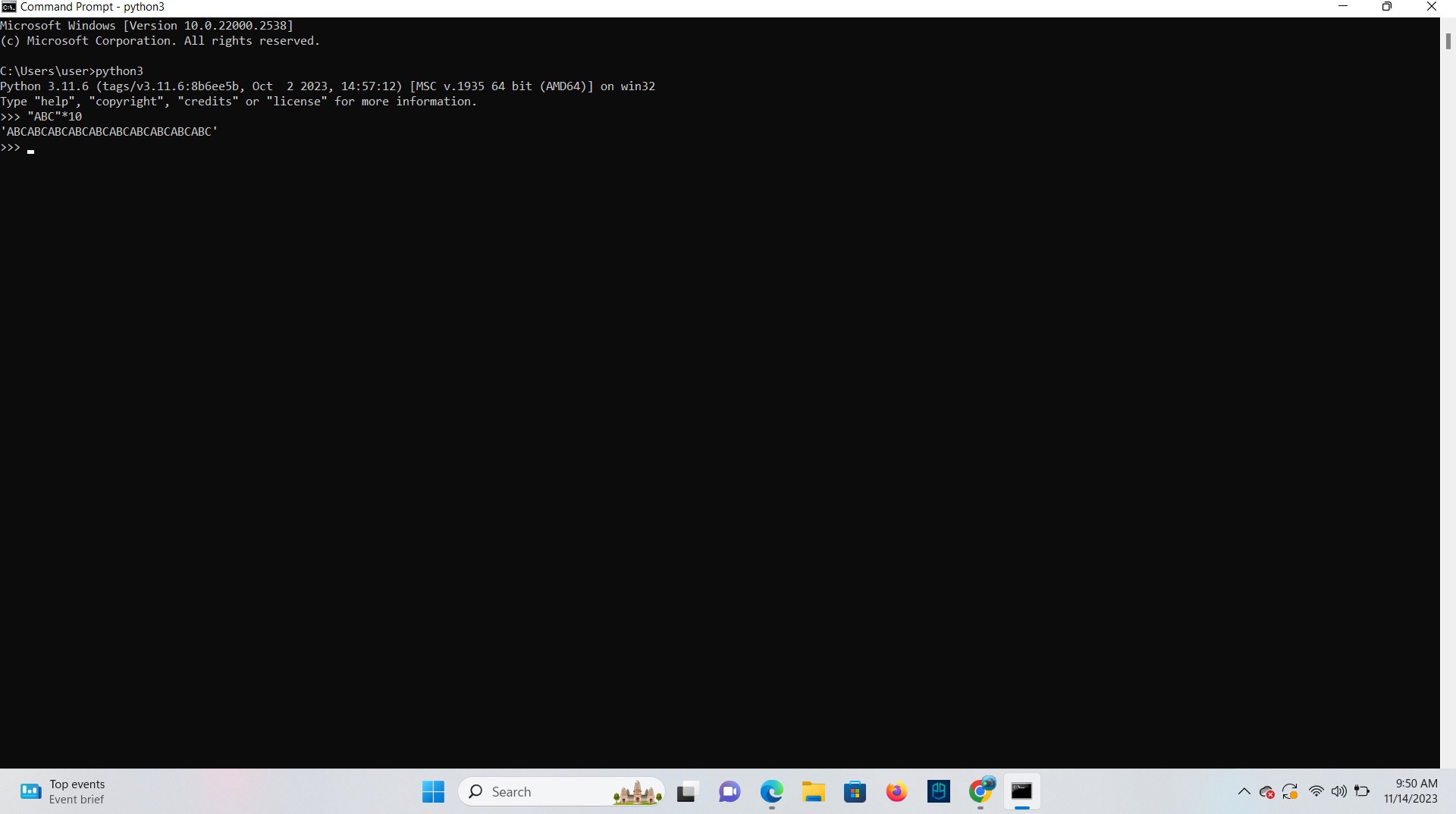
### Ans



**TASK**: Input the following code and examine the result. What is the \* operator doing to the given string operand?

“ABC” \* 10

**Ans** The \* operator is repeating the string operand 10 times.



## Calling Built-in functions

**TASK**: Write some code that calls the print() function several times, displaying your name, address and contact details. Add additional calls to the print() function which includes an argument that calculates and prints the length of your name, by calling the len() function.

### Ans

Name= “Aarson” Address= 123, capitalcity Contact= 9765589969

print (“My name is:”, Name) print (“My address is:”, Address) print (My Contact is:”, Contact) Name\_length=len(Name)

print (“My name length is:”, Name\_length)

## Getting input from the user

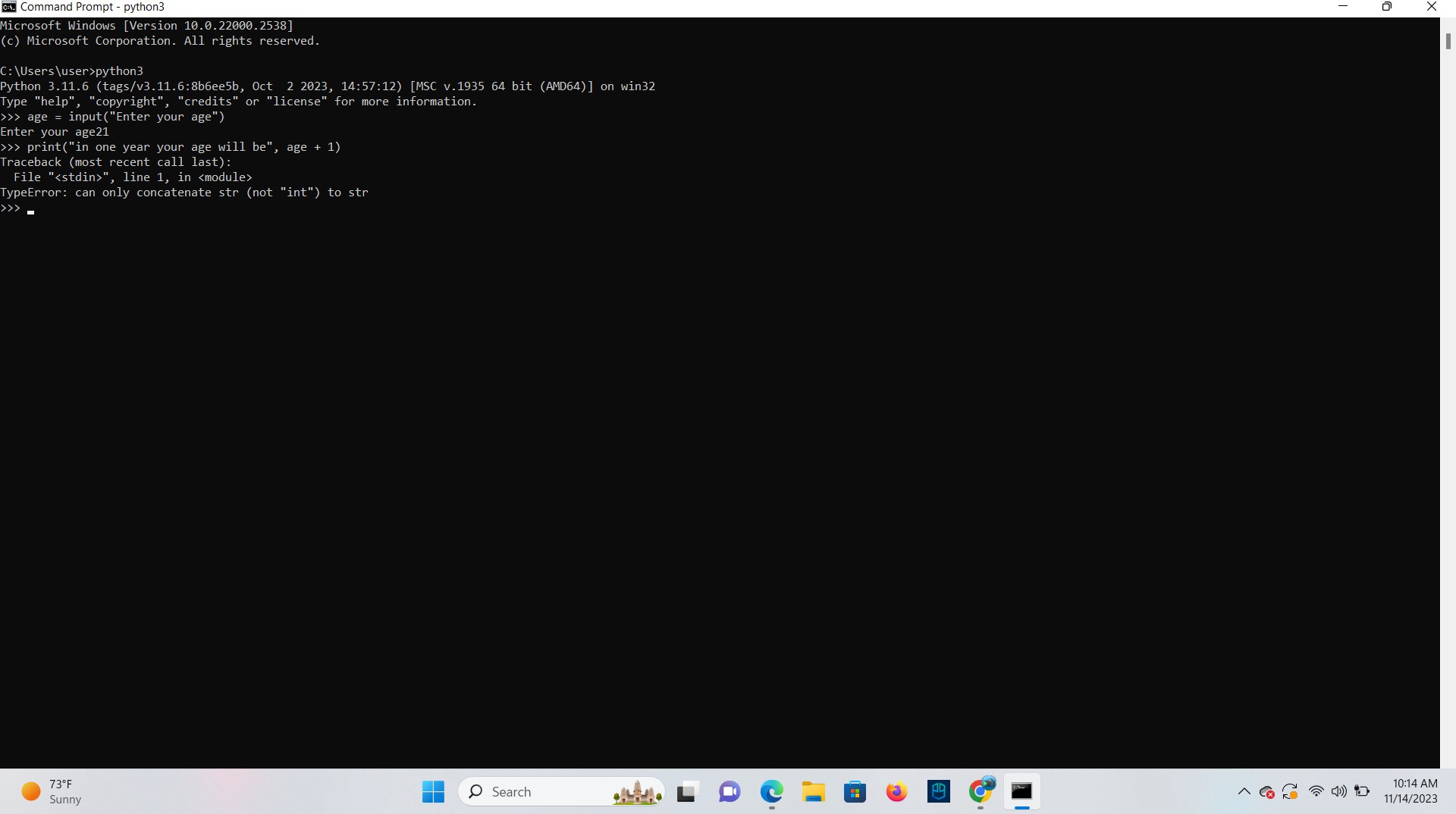
**TASK**: Input the following code, when asked to type your age input a numeric value such as

20. Does this program work? If not, why?

age = input(“Enter your age”)

print(“in one year your will be”, age + 1)

Ans: This program does not work as input() function return a string and when it concatenates with integer 1 it gives us a type error.



**TASK**: Write a program that prompts the user to input two numeric values. Once the values have been input display the product of these values, using the multiply (\*) operator.

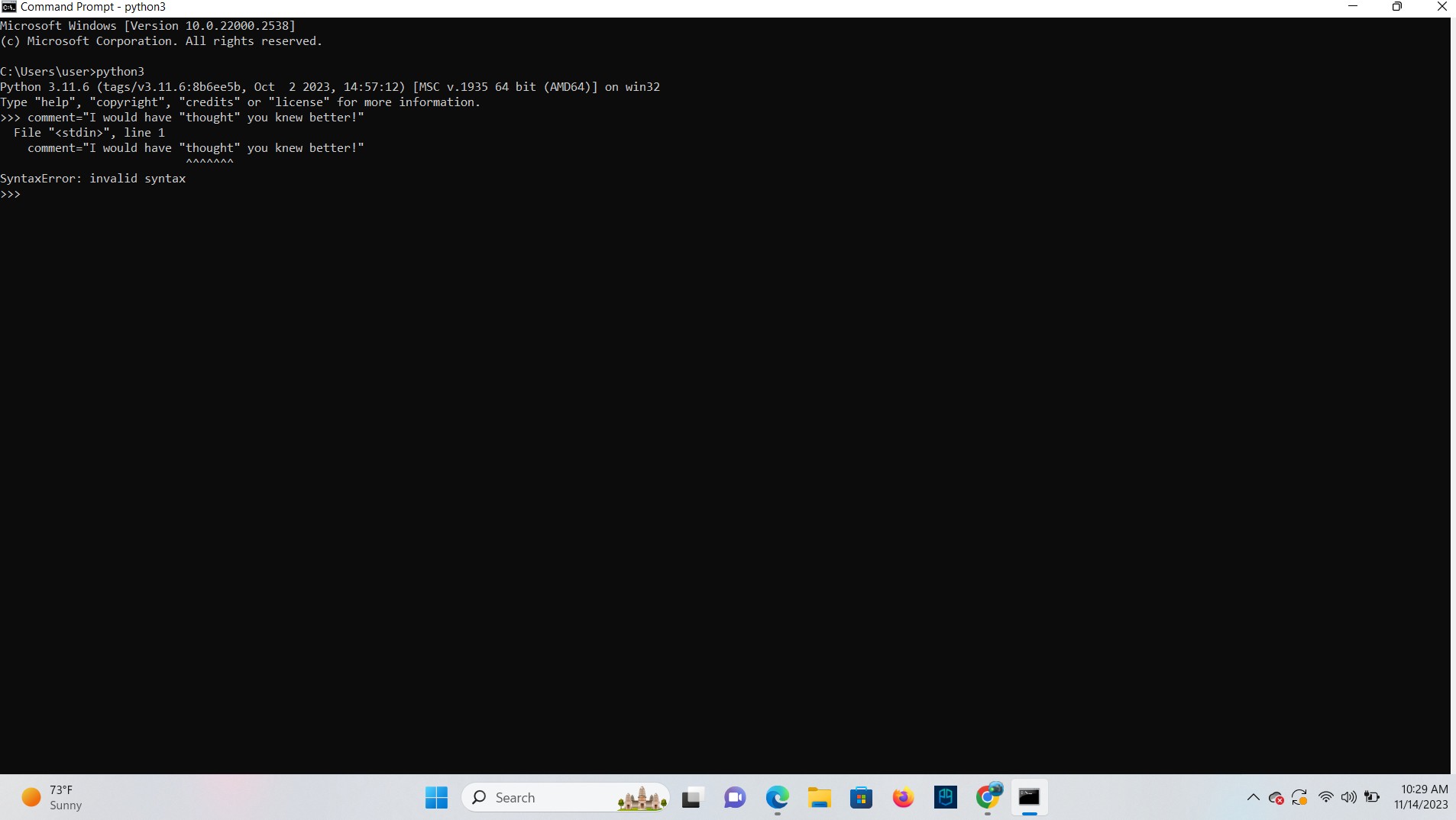
**Ans** a=2 b=4 c=a\*b

print (“The product of c is:”,c)

## Single, Double and Triple Quotes

comment = ‘I would have “thought” you knew better!’

**TASK**: Try writing the above assignment statement but only use double quotes instead of single quotes as the string delimiter. What is the result?

**Ans**

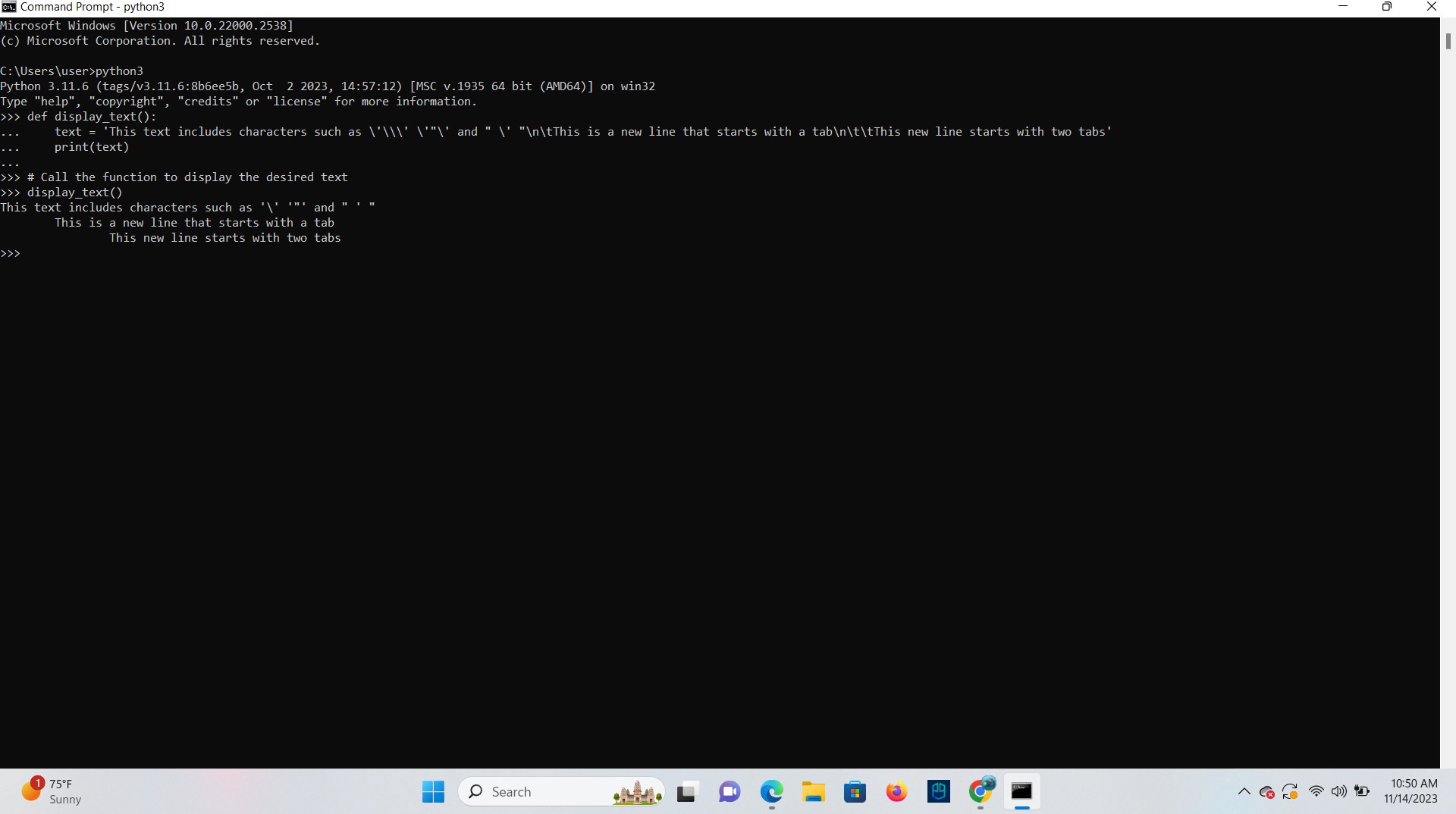
## Escape Sequences

**TASK**: Write some code that calls a print() function, which takes a single string argument that results in the following text being displayed (exactly as shown).

This text includes characters such as ‘\’ ‘ “ ‘ and “ ‘ “, This is a new line that starts with a tab

This new line starts with two tabs

### Ans

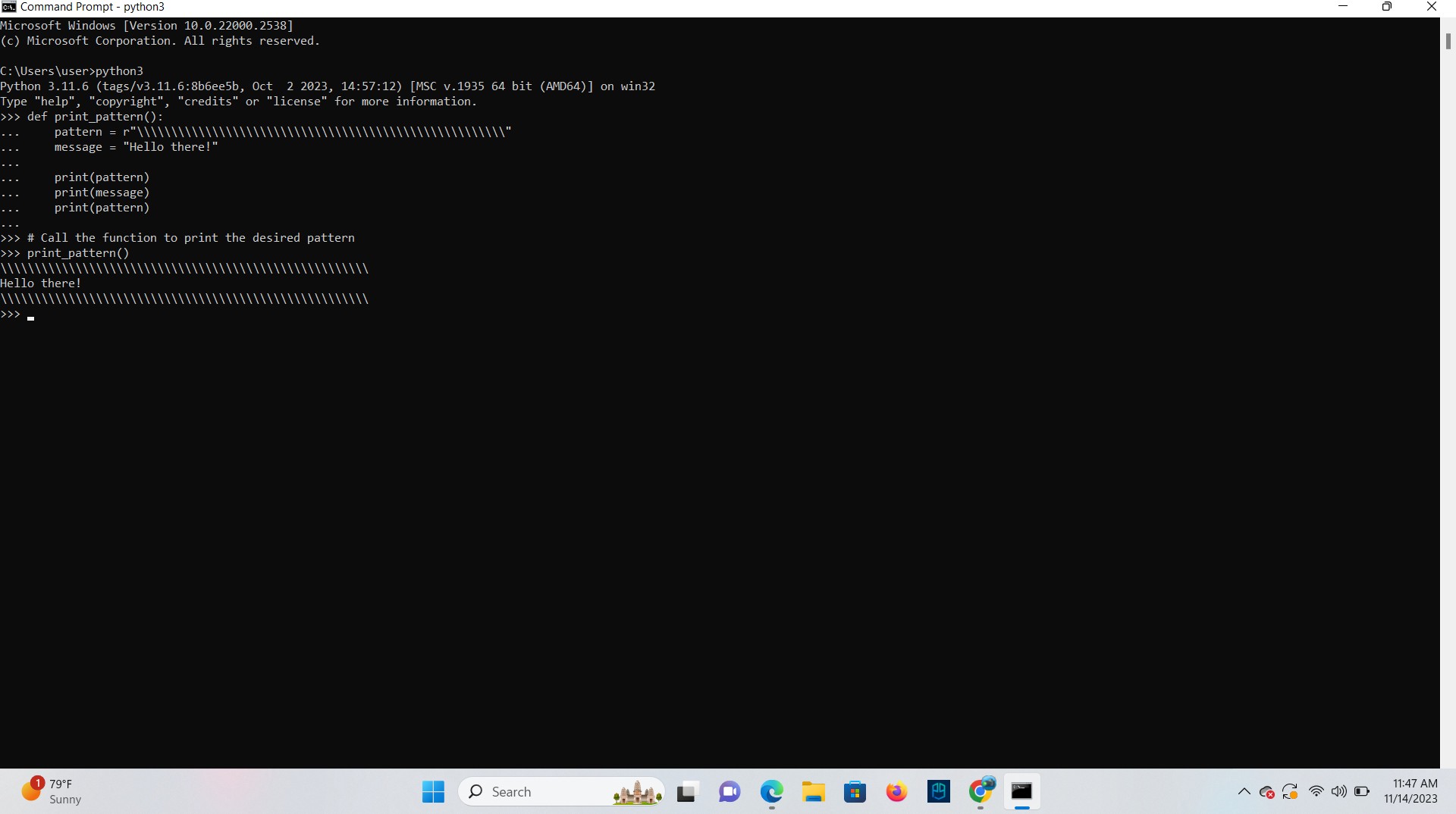


**TASK**: Write some code that calls a print() function, which takes a single string argument that results in the following text being displayed (exactly as shown).

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ Hello there!

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

**Ans**



## Using Triple Quotes

**TASK**: Write some code that calls a print() function, which takes a single string argument that results in the following text being displayed (exactly as shown). Do this without the use of any escape sequences.

This text spans three lines, and includes both single (‘), and double quotes (“).

### Ans

Def print ():

print (“This text spans three lines,\n\n and includes both single (‘) ,\n\n and double quotes (“).”)

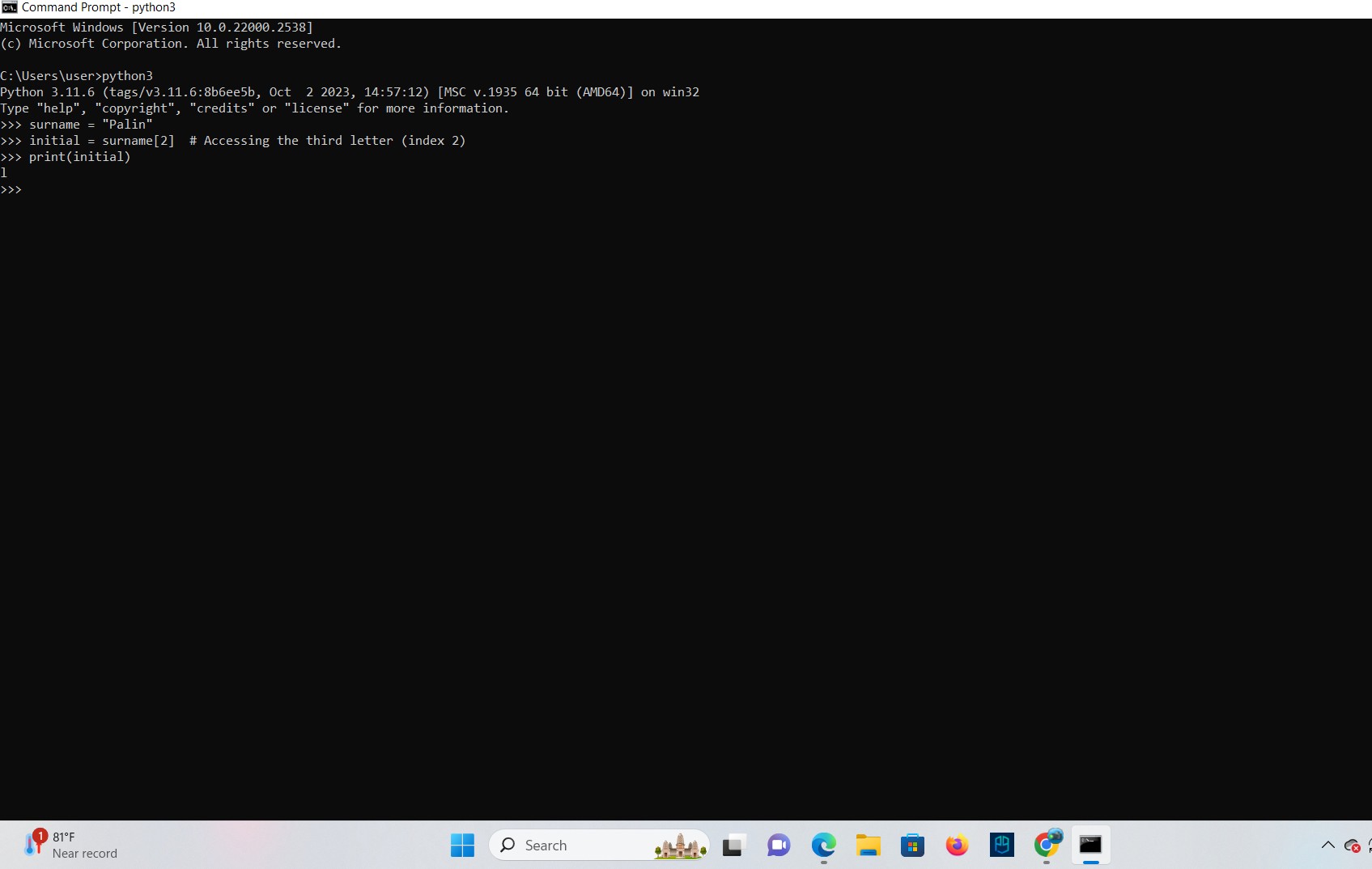
print ()

## Indexing and Slicing

surname = “Palin” initial = surname**[**0]

**TASK**: Rewrite the above example, so that the third letter of the ‘surname’ is accessed rather than the first, then print this letter to the screen.

### Ans



**TASK**: Rewrite the above example, so that the tenth letter of the ‘surname’ is accessed, and note the result.

### Ans

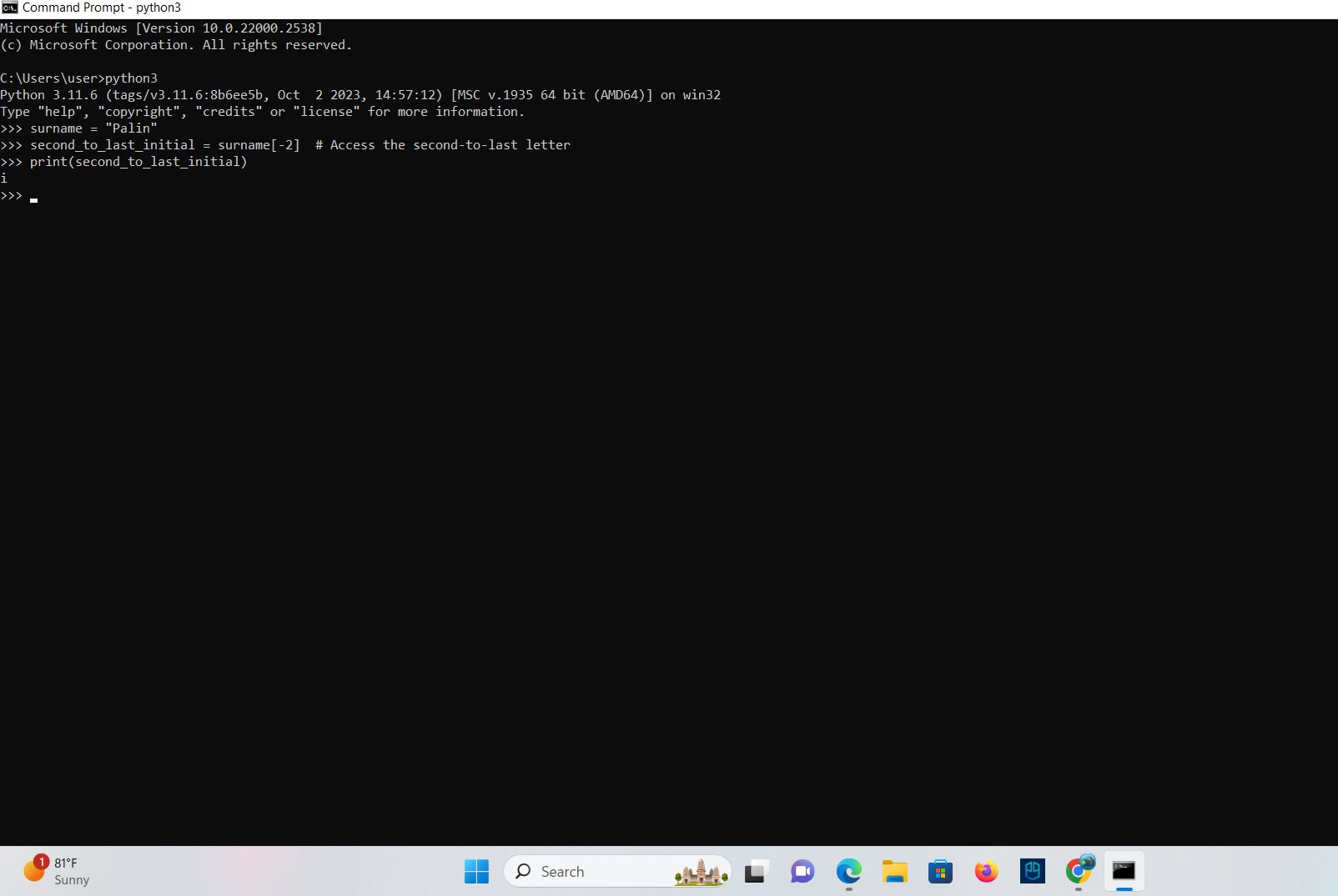
surname = "Palin" tenth\_letter = surname[9]

print("Tenth letter of the surname:", tenth\_letter**)**

Output:

Tenth letter of the surname: n

**TASK**: Rewrite the above example, so that the second from last letter of the ‘surname’ is accessed rather than the last, then print this letter to the screen.

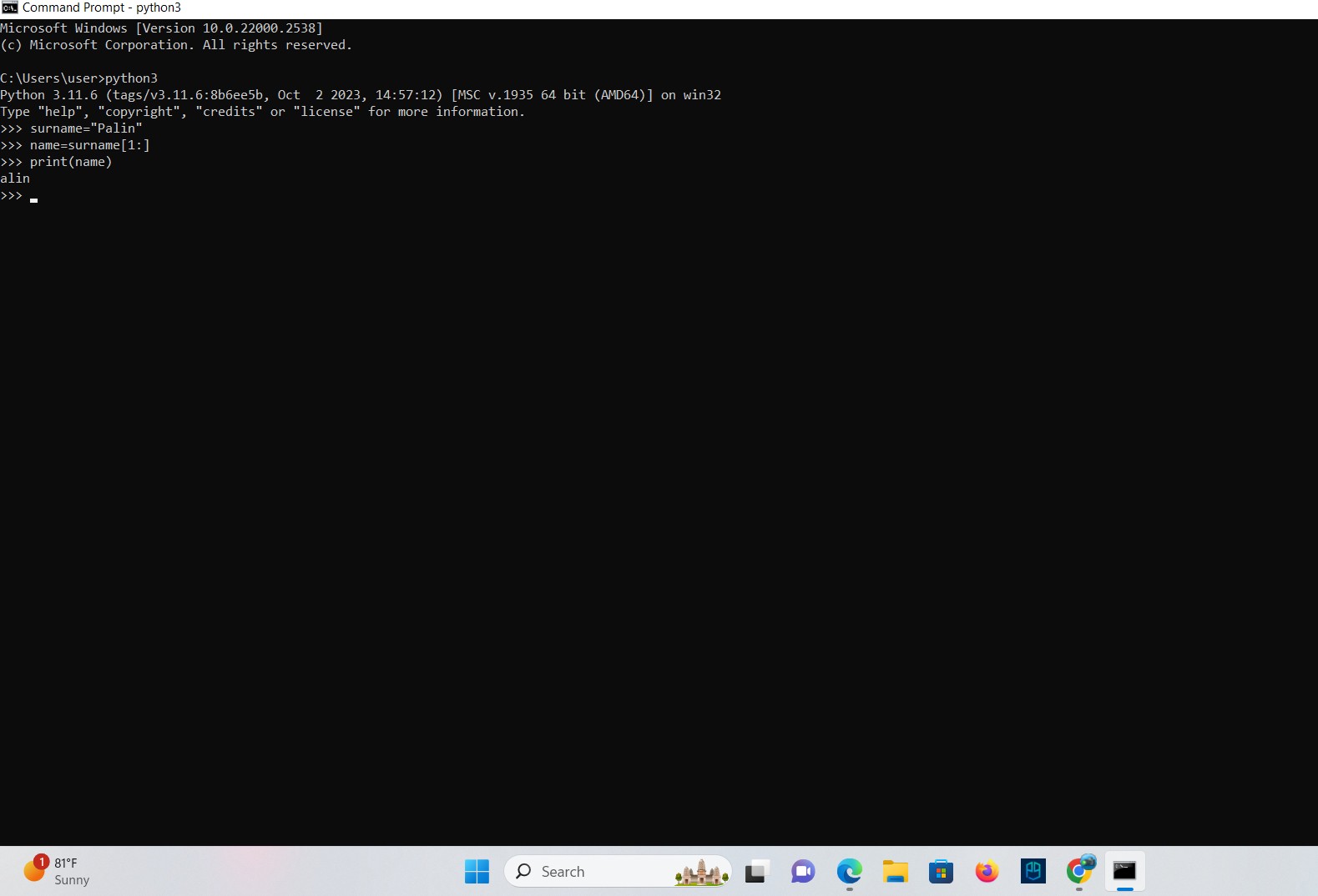
**Ans**

## Slicing

surname = “Palin” middle = surname[1:4]

**TASK**: Rewrite the above example, so that all of the characters of the ‘surname’ except the first character are sliced and then displayed on the screen.

### Ans



**TASK**: Write code that accesses and prints all characters of the ‘surname’ except the last

character.

### Ans

Surname=”Aarson”

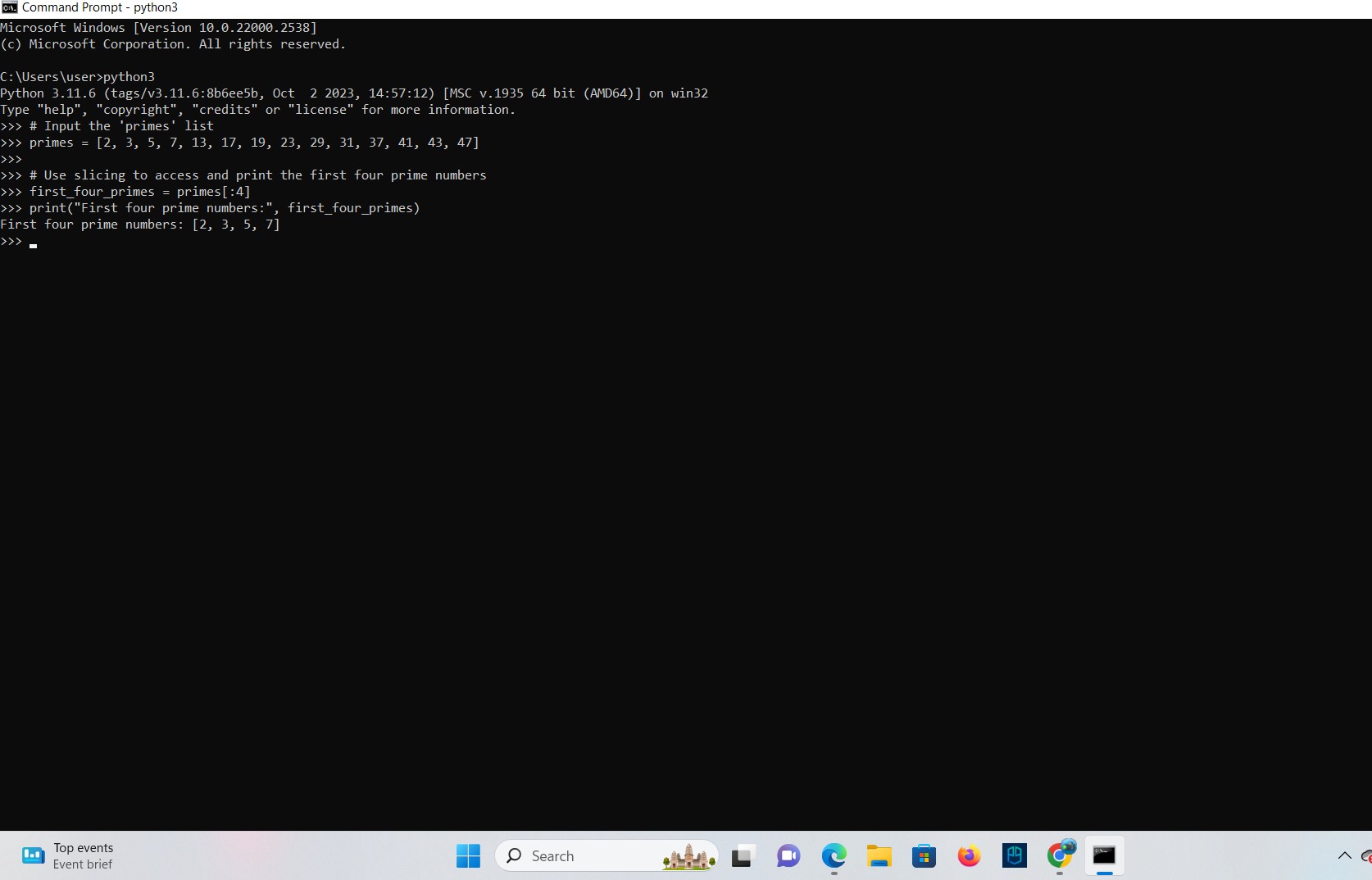
for char in surname[:-1]: print(char,end=’’)

## Introducing Lists

primes = [2, 3, 5, 7, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

**TASK**: Write code that uses slicing to access then print the first four prime numbers defined within the ‘primes’ list given above. Note: you will have to input that list first for testing Purposes.

**Ans**

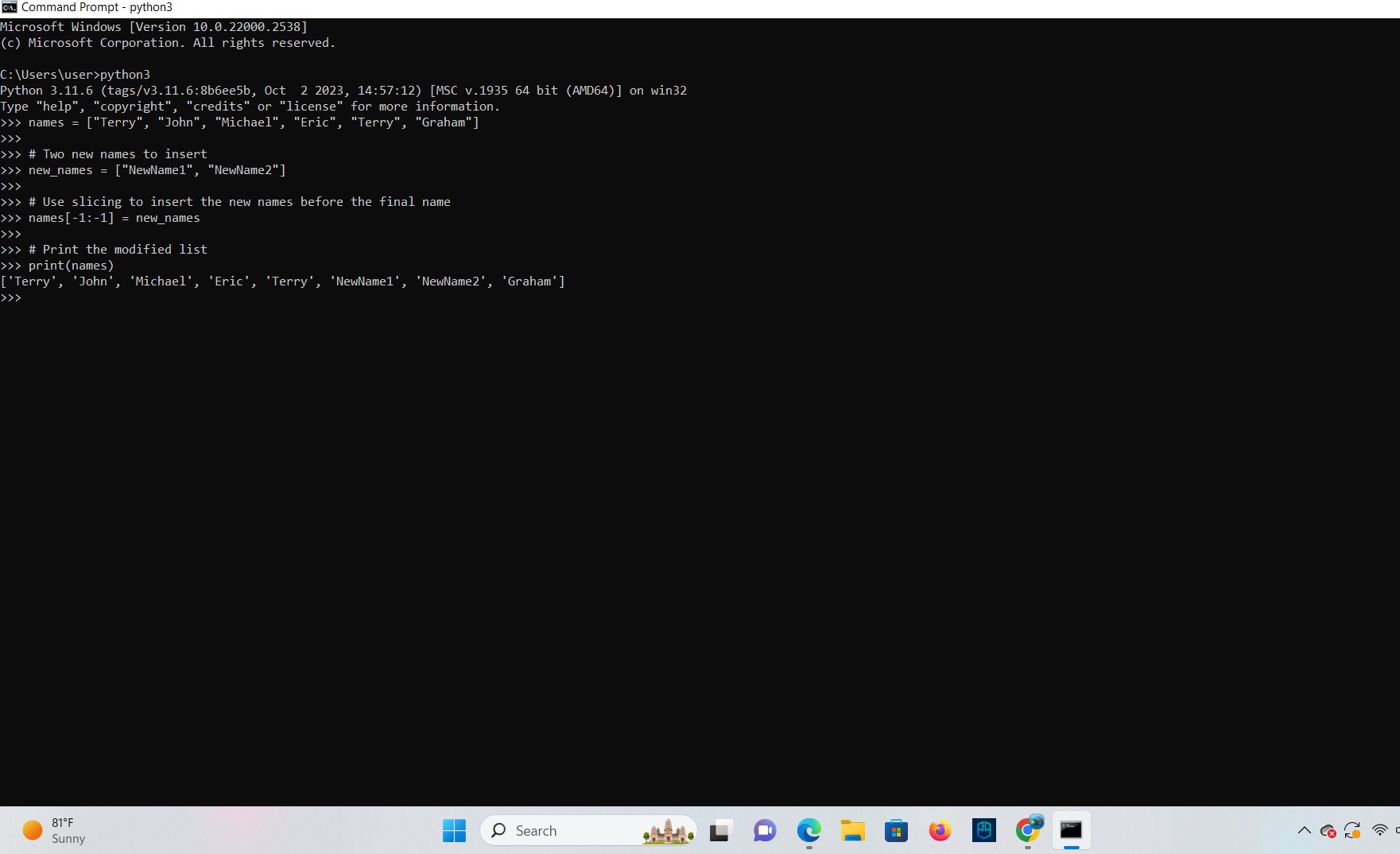


## Mutable and Immutable types

names = [“Terry”, “John”, “Michael”, “Eric”, “Terry”, “Graham”]

**TASK**: Write code that uses slicing to insert two new names just before the final name within the ‘names’ list.

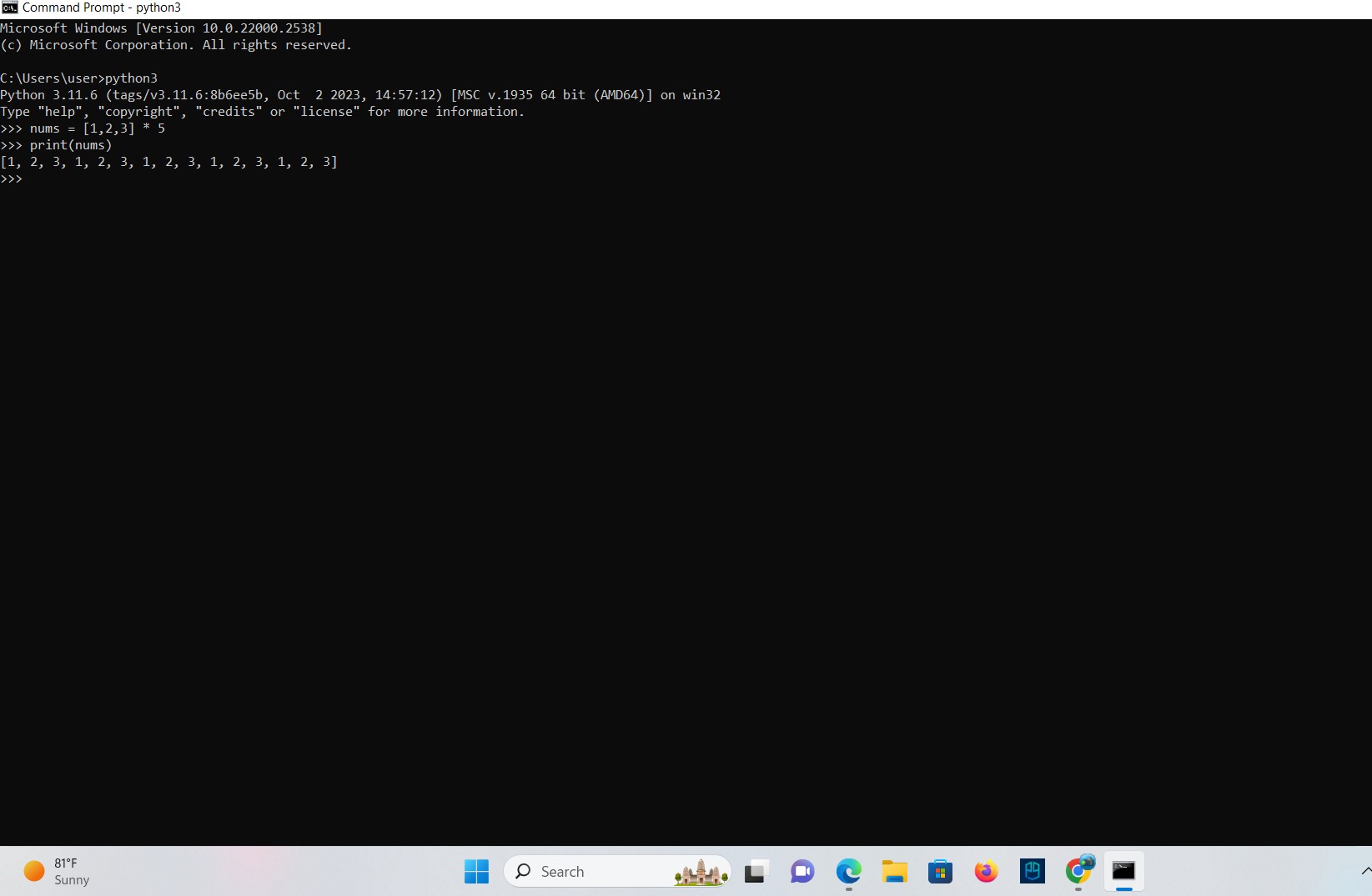
### Ans



**TASK**: Work out in your head what the contents of the ‘nums’ list would be, then check this using the Python interpreter.

nums = [1,2,3] \* 5

**Ans**



## Key Terminology

### Assignment

**Ans** The operators which is used to assign value to a variable.

### Data-type

**Ans** It represents the kind of value that tells what operations can be performed on data.

### Argument

**Ans** An argument is a value that is passed to a function when it is called.

### Indexing

**Ans** Indexing is the process of accessing an element in a sequence.

### Slicing

**Ans** It enables users to access the specific range of elements by mentioning their indices.

### Mutable

**Ans** Those who values can be modified once they are created.

### Immutable

**Ans** Those who values cannot be modified once they are created.