

**Objective:**

The objective of this program is to introduce the use of numpy. This is accomplished by having the user enter multiple variables such as a telephone and zip code then comparing them to a regular expression format to determine if it is in correct format, if not then ask to re-enter. Next the user is asked if they would like to re-enter or continue. If continuing then the user will input a 9 integer array that is used to make a matrix. This is accomplished with numpy. After the user enters the values for the first and second matrix they are then given a menu of choices on how to handle the matrices. They have the option to add, subtract, multiply or multiply by elements. All will return the value and if use would like to change their values and run the matrix calculator again. If not then it will exit to the original prompt if the user would like to use the application.

**Test cases:**

Below are some test cases that were used to test the program.

Test Cases:	User Input:	Predicted Output:	Actual Output:	Pass/Fail:
1	random key input	Please verify input, reprompt	Please verify input, reprompt	pass, fig 1
2	n, no, No, nO, NO	Thank you for using application	Thank you for using application	pass, fig 1
3	y, yes, yEs, Yes, YES	Please enter your phone number	Please enter your phone number	pass, fig 2
4	random key input	Please verify phone number	Please verify phone number	pass, fig 2
5	1111111111,111-1111111	Please verify phone number	Please verify phone number	pass, fig 2
6	(123)456-7890, 123-456-7890	Please enter your zipcode	Please enter your zipcode	pass, fig 2
7	123456789, 1234-56789	Please verify zipcode	Please verify zipcode	pass, fig 2
8	12345, 12345-6789	Display user phone and zipcode, prompt to re-enter	Display user phone and zipcode, prompt to re-enter	pass, fig 2
9	random key input	Invalid entry, reprompt	Invalid entry, reprompt	pass, fig 2
10	y, yes, yEs, Yes, YES	Please enter phone number	Please enter phone number	pass, fig 2
11	(123)456-7890, 123-456-7890	Please enter your zipcode	Please enter your zipcode	pass, fig 2
12	123456789, 1234-56789	Please verify zipcode	Please verify zipcode	pass, fig 2

13	12345, 12345-6789	Display user phone and zip code, prompt to re-enter	Display user phone and zip code, prompt to re-enter	pass, fig 2
14	n, no, No, nO, NO	Please enter your first 3x3 matrix values	Please enter your first 3x3 matrix values	pass, fig 2
15	123456789	Please verify input, want to use the app?	Please verify input, want to use the app?	pass, fig 2
16	1,2,3,4,5,6,7,8,9	Please enter second 3x3 matrix	Please enter second 3x3 matrix	pass, fig 3
17	random key input	Please verify input, want to use the app?	Please verify input, want to use the app?	pass, fig 3
18	1234567890	Please verify input, want to use the app?	Please verify input, want to use the app?	pass, fig 4
19		Please enter your first 3x3 matrix values	Please enter your first 3x3 matrix values	pass, fig 5
20	1,2,3,4,5,6,7,8,9	Please enter your second 3x3 matrix values	Please enter your second 3x3 matrix values	pass, fig 6
21	2,2,2,2,2,2,2,2	Matrix operation menu, select operation	Matrix operation menu, select operation	pass, fig 7
22	random key entry	Not valid selection, enter new matrix?	Not valid selection, enter new matrix?	pass, fig 7
23	y, yes, yEs, Yes, YES	Please enter your first 3x3 matrix values	Please enter your first 3x3 matrix values	pass, fig 7
24	1,2,3,4,5,6,7,8,9	Please enter your second 3x3 matrix values	Please enter your second 3x3 matrix values	pass, fig 7
25	2,2,2,2,2,2,2,2	Matrix operation menu, select operation	Matrix operation menu, select operation	pass, fig 7
26	1(addition)	display matrix addition, transpose, mean, enter another matrix	display matrix addition, transpose, mean, enter another matrix	pass, fig 7
	2(subtraction)	display matrix subtraction, transpose, mean, enter another matrix	display matrix subtraction, transpose, mean, enter another matrix	pass, fig 8

	3(multiply)	display matrix multiplication ,transpose, mean, enter another matrix	display matrix multiplication ,transpose, mean, enter another matrix	pass, fig 9
	4(multiply elements)	display matrix multiplication ,transpose, mean, enter another matrix	display matrix multiplication ,transpose, mean, enter another matrix	pass, fig 10
28	random key entry	Not valid selection, return to start	Not valid selection, return to start	pass, fig 11
29	no, n	Do you want to use app?	Do you want to use app?	pass, fig 11
30	no, n	Thank you for using application	Thank you for using application	pass, fig 11

Below are the images associated with the above test cases.

The screenshot shows a PyCharm IDE with a Python script named `practice_prog.py` and its execution output in the Run console.

**Script Code:**

```

45     matrix_menu()
46     else:
47         print('\n Invalid entry.')
48         try_again = 'y'
49         # if no then exit
50         elif game_continue in ['n', 'no']:
51             print('\n * 80')
52             print('### Thank you for using the application. ###')
53         else:
54             raise ValueError
55     except ValueError:
56         print('### Please verify input. ###')
57         game_continue = ''
58
59
60 def user_phone():
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

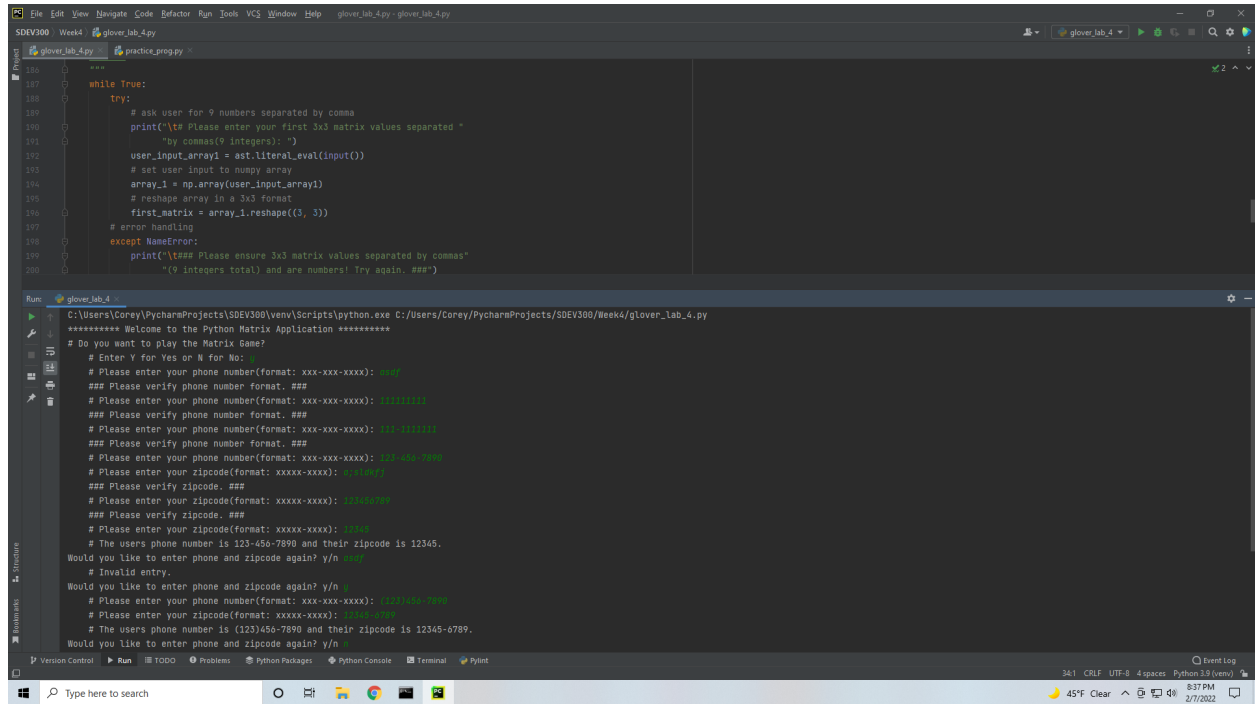
**Run Console Output:**

```

Run: glover_lab_4
main() while game_continue not in ['n', 'no']: try: elif game_continue in ['n', 'no...
***** Welcome to the Python Matrix Application *****
# Do you want to play the Matrix Game?
# Enter Y for Yes or N for No: y
### Please verify input. ###
# Do you want to play the Matrix Game?
# Enter Y for Yes or N for No: n
***** Thank you for using the application. *****
Process finished with exit code 0

```

Fig 1

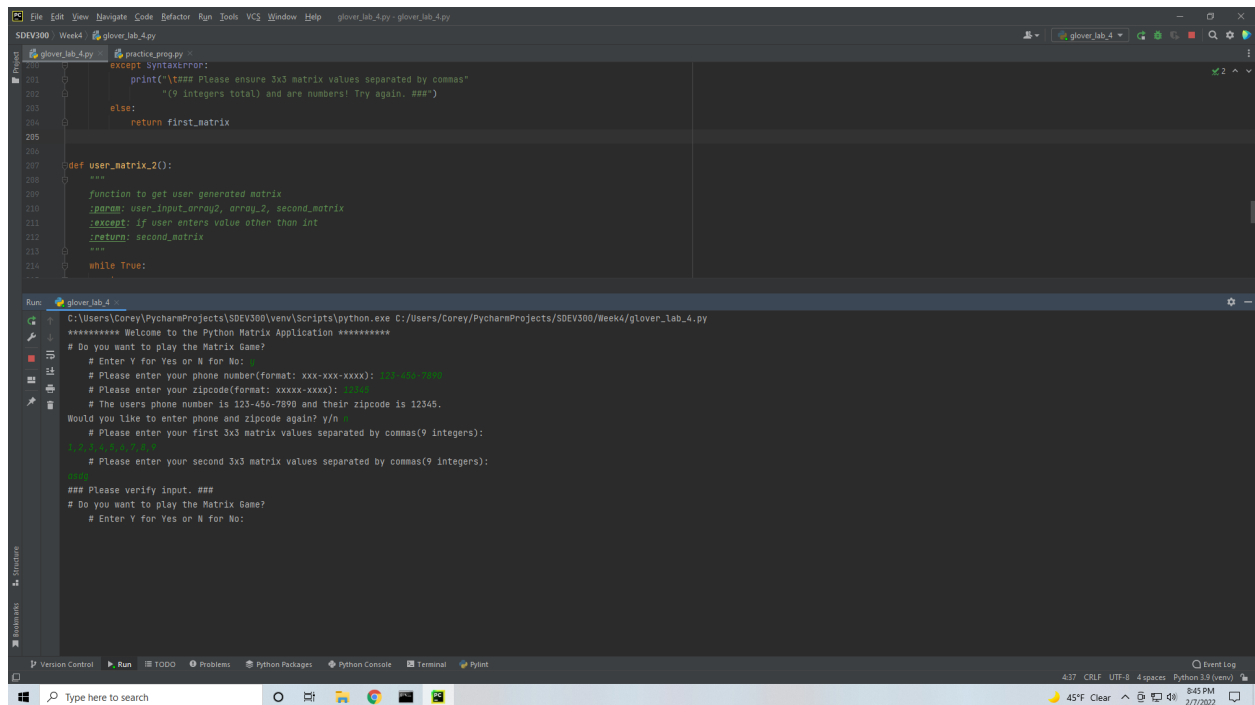


```
186: while True:
187:     try:
188:         # ask user for 9 numbers separated by comma
189:         print("\n Please enter your first 3x3 matrix values separated "
190:               "by commas(9 integers): ")
191:         user_input_array1 = ast.literal_eval(input())
192:         # set user input to numpy array
193:         array_1 = np.array(user_input_array1)
194:         # reshape array in a 3x3 format
195:         first_matrix = array_1.reshape((3, 3))
196:         # error handling
197:     except NameError:
198:         print("\n Please ensure 3x3 matrix values separated by commas"
199:               "\n(9 integers total) and are numbers! Try again. ##")
200:
```

Run: C:\Users\Corey\PycharmProjects\SDEV300\venv\Scripts\python.exe C:\Users\Corey\PycharmProjects\SDEV300\Week4\glover\_lab\_4.py

```
***** Welcome to the Python Matrix Application *****
# Do you want to play the Matrix Game?
# Enter Y for Yes or N for No: y
# Please enter your phone number(format: xxx-xxx-xxxx): 123-456-789
## Please verify phone number format. ##
# Please enter your phone number(format: xxx-xxx-xxxx): 123-456-789
## Please verify phone number format. ##
# Please enter your phone number(format: xxx-xxx-xxxx): 123-456-789
## Please verify phone number format. ##
# Please enter your phone number(format: xxx-xxx-xxxx): 123-456-789
# Please enter your zipcode(format: xxxxx-xxxx): 12345
## Please verify zipcode. ##
# Please enter your zipcode(format: xxxxx-xxxx): 12345
## Please verify zipcode. ##
# Please enter your zipcode(format: xxxxx-xxxx): 12345
# The users phone number is 123-456-7890 and their zipcode is 12345.
Would you like to enter phone and zipcode again? y/n n
# Invalid entry.
Would you like to enter phone and zipcode again? y/n n
# Please enter your phone number(format: xxx-xxx-xxxx): 123-456-789
# Please enter your zipcode(format: xxxxx-xxxx): 12345
# The users phone number is (123)456-7890 and their zipcode is 12345-0789.
Would you like to enter phone and zipcode again? y/n n
```

Fig 2

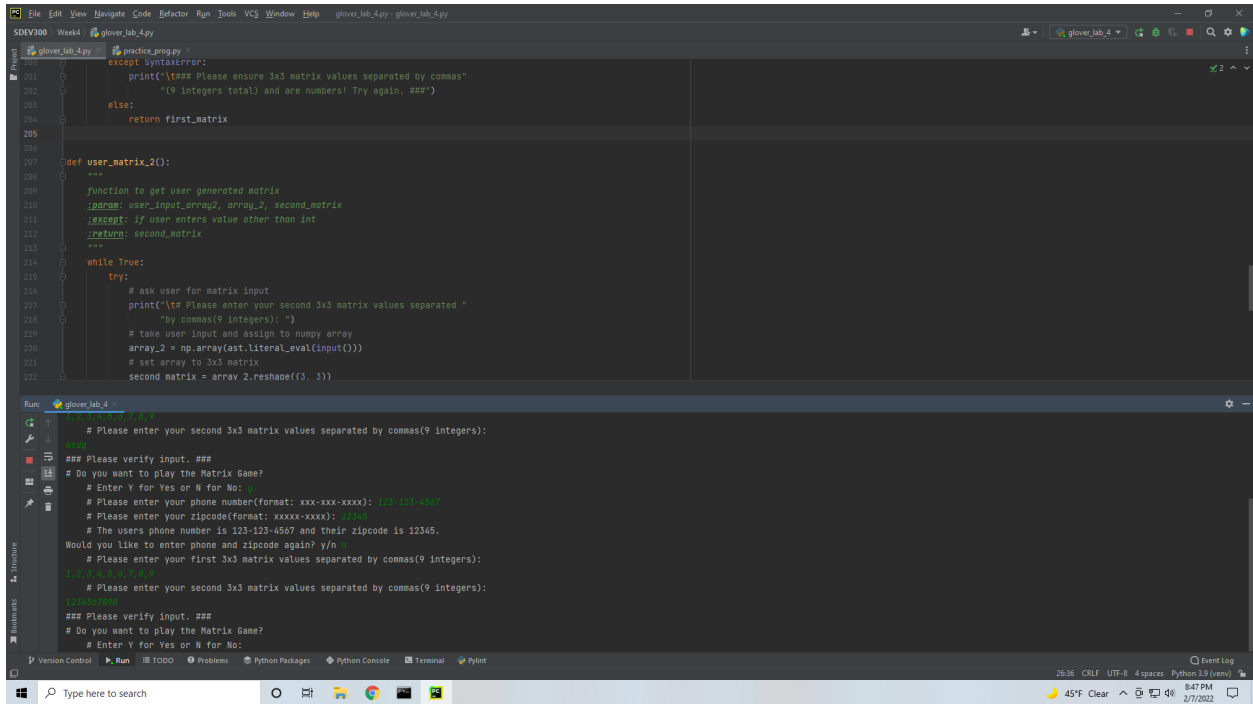


```
200: except SyntaxError:
201:     print("\n Please ensure 3x3 matrix values separated by commas"
202:           "\n(9 integers total) and are numbers! Try again. ##")
203: else:
204:     return first_matrix
205:
206:
207: def user_matrix_2():
208:     """
209:     function to get user generated matrix
210:     :param: user_input_array2, array_2, second_matrix
211:     :except: if user enters value other than int
212:     :return: second_matrix
213:     """
214:     while True:
215:
```

Run: C:\Users\Corey\PycharmProjects\SDEV300\venv\Scripts\python.exe C:\Users\Corey\PycharmProjects\SDEV300\Week4\glover\_lab\_4.py

```
***** Welcome to the Python Matrix Application *****
# Do you want to play the Matrix Game?
# Enter Y for Yes or N for No: y
# Please enter your phone number(format: xxx-xxx-xxxx): 123-456-789
# Please enter your zipcode(format: xxxxx-xxxx): 12345
# The users phone number is 123-456-7890 and their zipcode is 12345.
Would you like to enter phone and zipcode again? y/n n
# Please enter your first 3x3 matrix values separated by commas(9 integers):
123
# Please enter your second 3x3 matrix values separated by commas(9 integers):
456
## Please verify input. ##
# Do you want to play the Matrix Game?
# Enter Y for Yes or N for No: n
```

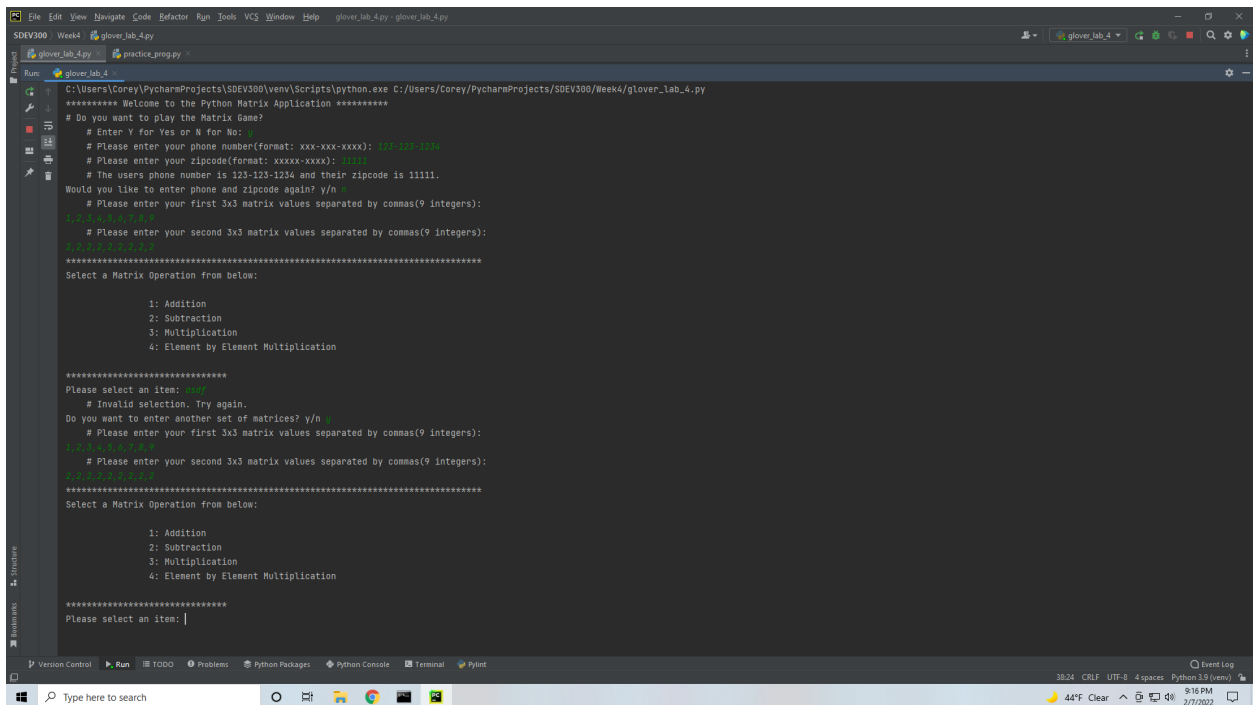
Fig 3



```
File Edit View Extensions Code Refactor Run Tools VCS Window Help - glover_lab_4.py - glover_lab_4.py
SDEV300 Week4 glover_lab_4.py
glover_lab_4.py
def user_matrix_2():
    """
    function to get user generated matrix
    :param: user_input_array2, array_2, second_matrix
    :except: if user enters value other than int
    :return: second_matrix
    """
    while True:
        try:
            # ask user for matrix input
            print("\n Please enter your second 3x3 matrix values separated by commas(9 integers): ")
            # by commas(9 integers):
            # take user input and assign to numpy array
            array_2 = np.array(ast.literal_eval(input()))
            # set array to 3x3 matrix
            second_matrix = array_2.reshape((3, 3))

Run
glover_lab_4
# Please enter your second 3x3 matrix values separated by commas(9 integers):
### Please verify input. ###
# Do you want to play the Matrix Game?
# Enter Y for Yes or N for No:
# Please enter your phone number(format: xxx-xxx-xxxx):
# Please enter your zipcode(format: xxxxx-xxxx):
# The users phone number is 123-123-4567 and their zipcode is 12345.
Would you like to enter phone and zipcode again? y/n
# Please enter your first 3x3 matrix values separated by commas(9 integers):
# Please enter your second 3x3 matrix values separated by commas(9 integers):
### Please verify input. ###
# Do you want to play the Matrix Game?
# Enter Y for Yes or N for No:
Version Control Run Python Packages Python Console Terminal Pylint
26.36 CRLF UTF-8 4 spaces Python 3.9 (venv)
45°F Clear 8:47 PM 2/7/2022
```

Fig 4



```
File Edit View Extensions Code Refactor Run Tools VCS Window Help - glover_lab_4.py - glover_lab_4.py
SDEV300 Week4 glover_lab_4.py
glover_lab_4.py
def user_matrix_2():
    """
    function to get user generated matrix
    :param: user_input_array2, array_2, second_matrix
    :except: if user enters value other than int
    :return: second_matrix
    """
    while True:
        try:
            # ask user for matrix input
            print("\n Please enter your second 3x3 matrix values separated by commas(9 integers): ")
            # by commas(9 integers):
            # take user input and assign to numpy array
            array_2 = np.array(ast.literal_eval(input()))
            # set array to 3x3 matrix
            second_matrix = array_2.reshape((3, 3))

Run
glover_lab_4
C:\Users\Corey\PycharmProjects\SDEV300\venv\Scripts\python.exe C:\Users\Corey\PycharmProjects\SDEV300\Week4\glover_lab_4.py
***** Welcome to the Python Matrix Application *****
# Do you want to play the Matrix Game?
# Enter Y for Yes or N for No:
# Please enter your phone number(format: xxx-xxx-xxxx):
# Please enter your zipcode(format: xxxxx-xxxx):
# The users phone number is 123-123-1234 and their zipcode is 11111.
Would you like to enter phone and zipcode again? y/n
# Please enter your first 3x3 matrix values separated by commas(9 integers):
# Please enter your second 3x3 matrix values separated by commas(9 integers):
*****
Select a Matrix Operation from below:
1: Addition
2: Subtraction
3: Multiplication
4: Element by Element Multiplication
Please select an item:
# Invalid selection. Try again.
Do you want to enter another set of matrices? y/n
# Please enter your first 3x3 matrix values separated by commas(9 integers):
# Please enter your second 3x3 matrix values separated by commas(9 integers):
*****
Select a Matrix Operation from below:
1: Addition
2: Subtraction
3: Multiplication
4: Element by Element Multiplication
Please select an item:
Version Control Run Python Packages Python Console Terminal Pylint
38.24 CRLF UTF-8 4 spaces Python 3.9 (venv)
44°F Clear 9:16 PM 2/7/2022
```

Fig 5

```
File Edit View Settings Code Refactor Run Tools VCS Window Help glover_lab_4py - glover_lab_4py
SDEV300 Week4 glover_lab_4py
glover_lab_4py practice_prog.py
Run glover_lab_4py

3: Multiplication
4: Element by Element Multiplication

*****
Please select an item:
# Invalid selection. Try again.
Do you want to enter another set of matrices? y/n
# Please enter your first 3x3 matrix values separated by commas(9 integers):
1,2,3,4,5,6,7,8,9
# Please enter your second 3x3 matrix values separated by commas(9 integers):
2,2,2,2,2,2,2,2,2
*****
Select a Matrix Operation from below:

1: Addition
2: Subtraction
3: Multiplication
4: Element by Element Multiplication

*****
Please select an item:
# The first matrix entered was:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
# The second matrix entered was
[[2 2]
 [2 2]
 [2 2]]
# The sum of the two matrix is:
[[ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
#The transpose is
[[ 1  4  7]
 [ 2  5  8]
 [ 3  6  9]]
The row mean is [0.7, 8.]
The column mean is [ 4.  7. 10.]
Do you want to enter another set of matrices? y/n

Version Control Run TODO Problems Python Packages Python Console Terminal Pylint
57.51 CRLF UTF-8 4 spaces Python 3.8 (venv)
Type here to search 44°F Clear 9:17 PM 2/7/2022
```

Fig 6

```
File Edit View Settings Code Refactor Run Tools VCS Window Help glover_lab_4py - glover_lab_4py
SDEV300 Week4 glover_lab_4py
glover_lab_4py practice_prog.py
Run glover_lab_4py

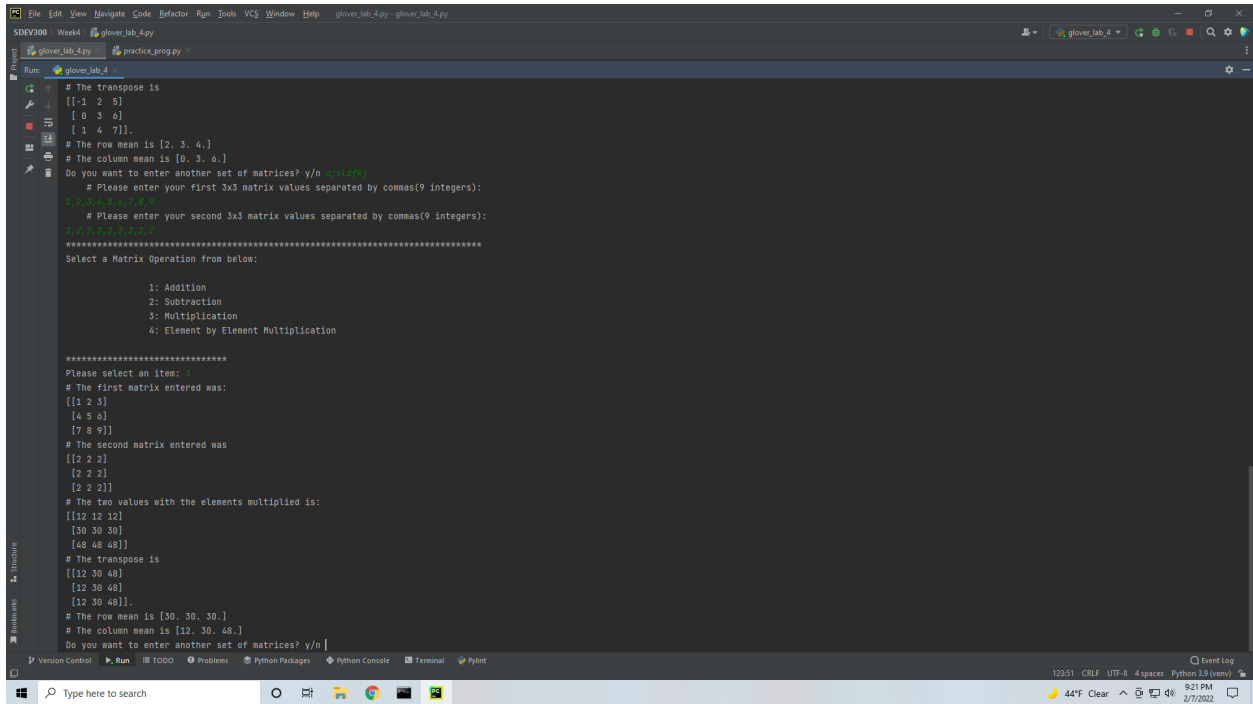
[ 4  7 10]
[ 5  8 11]]
The row mean is [0.7, 8.]
The column mean is [ 4.  7. 10.]
Do you want to enter another set of matrices? y/n
# Please enter your first 3x3 matrix values separated by commas(9 integers):
4,7,10,5,8,11,2,2,2
# Please enter your second 3x3 matrix values separated by commas(9 integers):
2,2,2,2,2,2,2,2,2
*****
Select a Matrix Operation from below:

1: Addition
2: Subtraction
3: Multiplication
4: Element by Element Multiplication

*****
Please select an item:
# The first matrix entered was:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
# The second matrix entered was
[[2 2]
 [2 2]
 [2 2]]
# The difference of the two matrix is:
[[-1  0  1]
 [ 2  3  4]
 [ 5  6  7]]
# The transpose is
[[-1  2  5]
 [ 0  3  6]
 [ 1  4  7]]
# The row mean is [2.3, 4.]
# The column mean is [0.3, 6.]
Do you want to enter another set of matrices? y/n
# Please enter your first 3x3 matrix values separated by commas(9 integers):

Version Control Run TODO Problems Python Packages Python Console Terminal Pylint
52.1 CRLF UTF-8 4 spaces Python 3.8 (venv)
Type here to search 44°F Clear 9:21 PM 2/7/2022
```

Fig 7



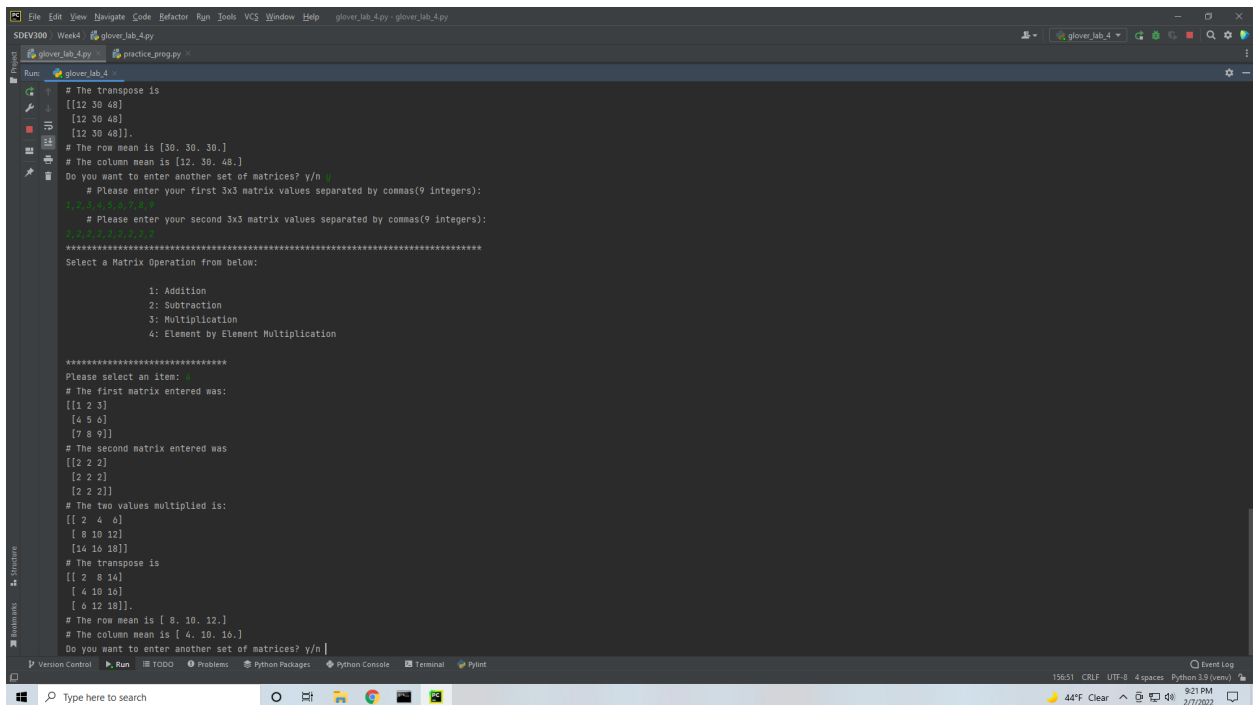
```
File Edit View Extensions Code Refactor Run Tools VCS Window Help glover_lab_4py - glover_lab_4py
SDEV300 Week4 glover_lab_4py
glover_lab_4py practice_prog.py
Run glover_lab_4
# The transpose is
[[1 2 5]
 [0 5 6]
 [1 4 7]]
# The row mean is [2.3 4.]
# The column mean is [0.3 6.]
Do you want to enter another set of matrices? y/n y
# Please enter your first 3x3 matrix values separated by commas(9 integers):
1,2,3,4,5,6,7,8,9
# Please enter your second 3x3 matrix values separated by commas(9 integers):
1,2,3,4,5,6,7,8,9
*****
Select a Matrix Operation from below:

1: Addition
2: Subtraction
3: Multiplication
4: Element by Element Multiplication

*****
Please select an item: 1
# The first matrix entered was:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
# The second matrix entered was
[[2 2 2]
 [2 2 2]
 [2 2 2]]
# The two values with the elements multiplied is:
[[12 12 12]
 [30 30 30]
 [48 48 48]]
# The transpose is
[[12 30 48]
 [12 30 48]
 [12 30 48]].
# The row mean is [30. 30. 30.]
# The column mean is [12. 30. 48.]
Do you want to enter another set of matrices? y/n |

Version Control Run TODO Problems Python Packages Python Console Terminal Pylint
123.51 CRLF UTF-8 4 spaces Python 3.8 (venv)
Type here to search 44°F Clear 9:21 PM 2/7/2022
```

Fig 8



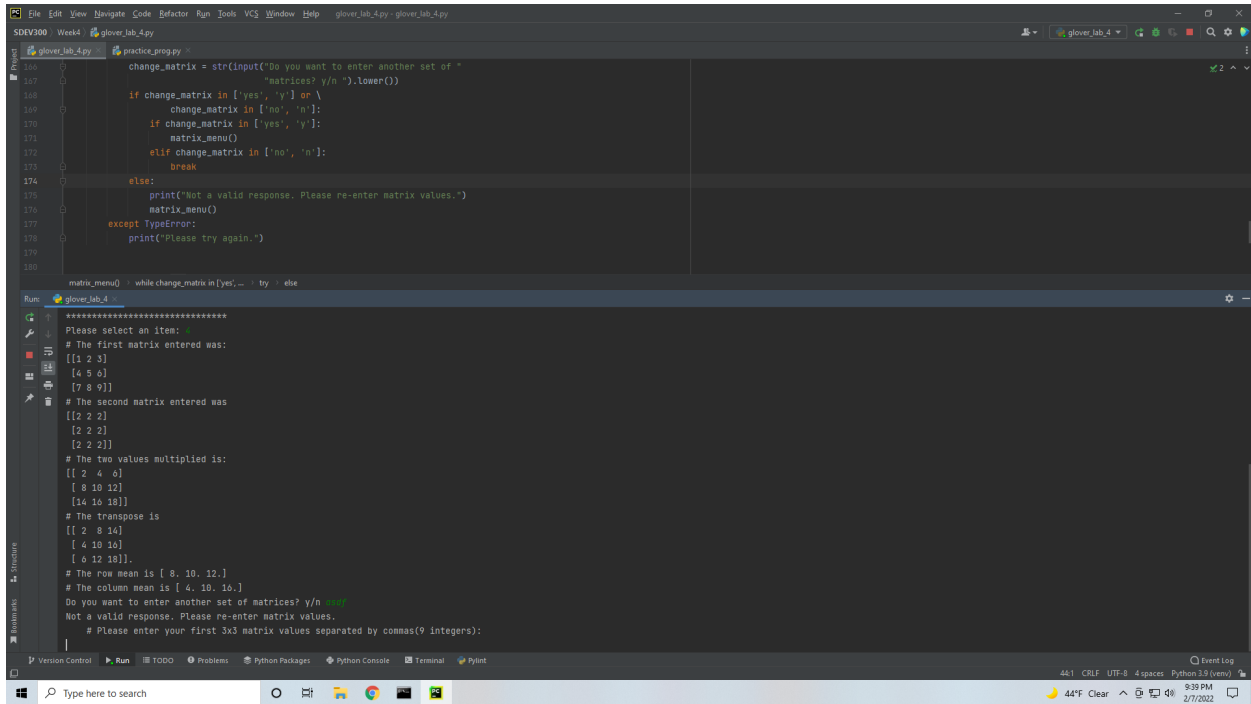
```
File Edit View Extensions Code Refactor Run Tools VCS Window Help glover_lab_4py - glover_lab_4py
SDEV300 Week4 glover_lab_4py
glover_lab_4py practice_prog.py
Run glover_lab_4
# The transpose is
[[12 30 48]
 [12 30 48]
 [12 30 48]].
# The row mean is [30. 30. 30.]
# The column mean is [12. 30. 48.]
Do you want to enter another set of matrices? y/n
# Please enter your first 3x3 matrix values separated by commas(9 integers):
1,2,3,4,5,6,7,8,9
# Please enter your second 3x3 matrix values separated by commas(9 integers):
1,2,3,4,5,6,7,8,9
*****
Select a Matrix Operation from below:

1: Addition
2: Subtraction
3: Multiplication
4: Element by Element Multiplication

*****
Please select an item: 1
# The first matrix entered was:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
# The second matrix entered was
[[2 2 2]
 [2 2 2]
 [2 2 2]]
# The two values multiplied is:
[[ 2  4  6]
 [ 8 10 12]
 [14 16 18]]
# The transpose is
[[ 2  8 16]
 [ 4 10 18]
 [ 6 12 18]].
# The row mean is [ 8. 10. 12.]
# The column mean is [ 4. 10. 16.]
Do you want to enter another set of matrices? y/n |

Version Control Run TODO Problems Python Packages Python Console Terminal Pylint
156.51 CRLF UTF-8 4 spaces Python 3.8 (venv)
Type here to search 44°F Clear 9:21 PM 2/7/2022
```

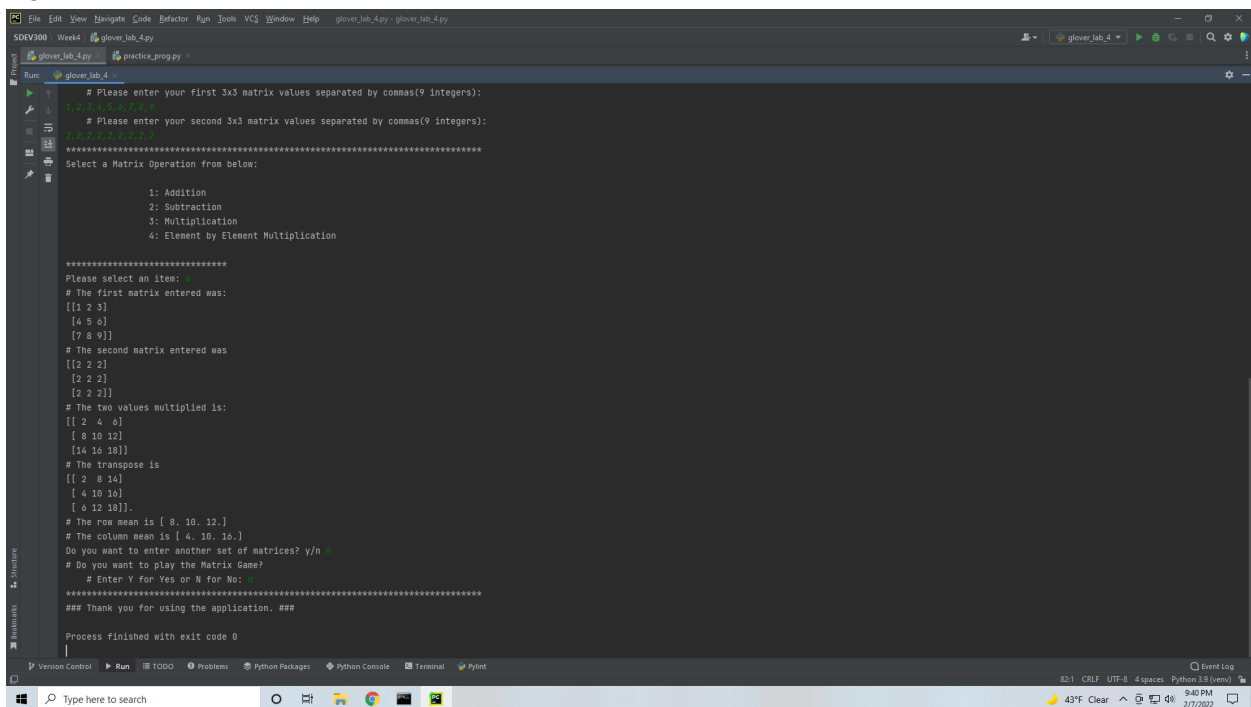
Fig 9



```
156 change_matrix = str(input("Do you want to enter another set of "
157 "matrices? y/n ").lower())
158 if change_matrix in ['yes', 'y'] or \
159 change_matrix in ['no', 'n']:
160     if change_matrix in ['yes', 'y']:
161         matrix_menu()
162     elif change_matrix in ['no', 'n']:
163         break
164 else:
165     print("Not a valid response. Please re-enter matrix values.")
166     matrix_menu()
167 except TypeError:
168     print("Please try again.")
169
170 matrix_menu() while change_matrix in ['yes', 'y']: try: else
```

```
*****
Please select an item: 1
# The first matrix entered was:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
# The second matrix entered was
[[2 2 2]
 [2 2 2]
 [2 2 2]]
# The two values multiplied is:
[[ 2  4  6]
 [ 8 10 12]
 [14 16 18]]
# The transpose is
[[ 2  8 14]
 [ 4 10 16]
 [ 6 12 18]]
# The row mean is [ 8. 10. 12.]
# The column mean is [ 4. 10. 16.]
Do you want to enter another set of matrices? y/n yes
Not a valid response. Please re-enter matrix values.
# Please enter your first 3x3 matrix values separated by commas(9 integers):
```

Fig 10



```
# Please enter your first 3x3 matrix values separated by commas(9 integers):
1,2,3,4,5,6,7,8,9
# Please enter your second 3x3 matrix values separated by commas(9 integers):
2,2,2,2,2,2,2,2,2
*****
Select a Matrix Operation from below:

1: Addition
2: Subtraction
3: Multiplication
4: Element by Element Multiplication

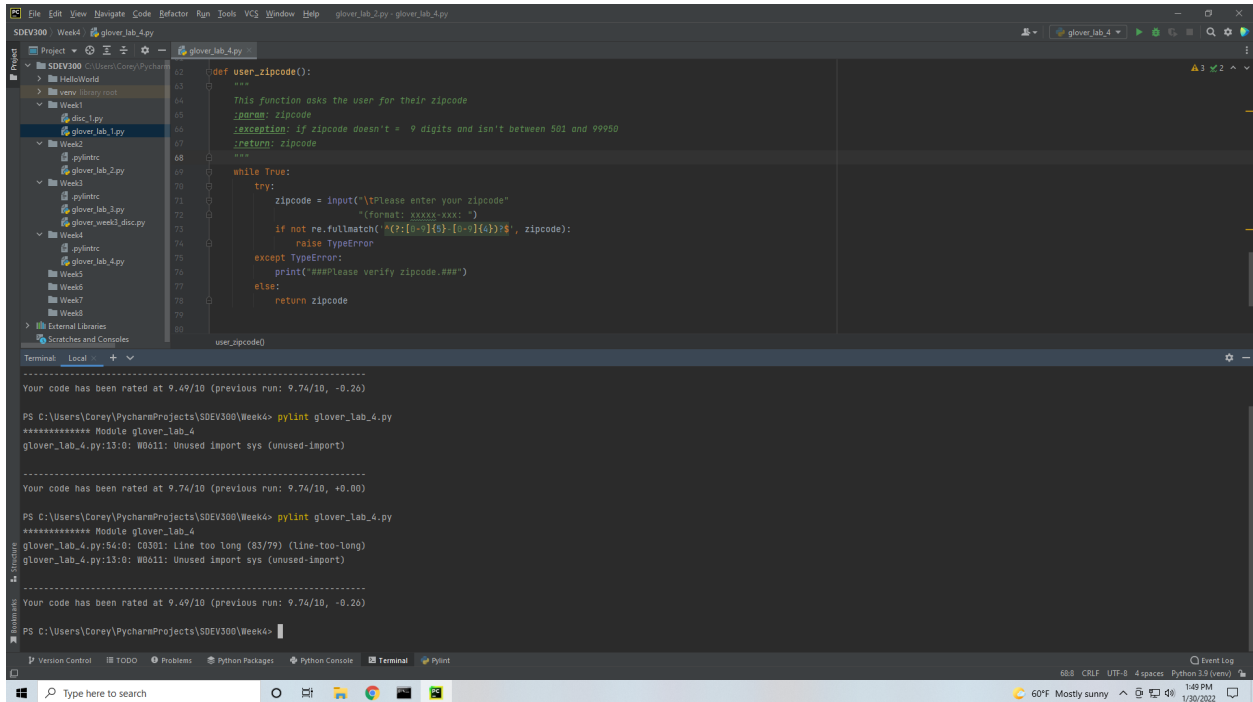
Please select an item: 4
# The first matrix entered was:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
# The second matrix entered was
[[2 2 2]
 [2 2 2]
 [2 2 2]]
# The two values multiplied is:
[[ 2  4  6]
 [ 8 10 12]
 [14 16 18]]
# The transpose is
[[ 2  8 14]
 [ 4 10 16]
 [ 6 12 18]]
# The row mean is [ 8. 10. 12.]
# The column mean is [ 4. 10. 16.]
Do you want to enter another set of matrices? y/n
# Do you want to play the Matrix Game?
Enter Y for Yes or N for No:
*****
### Thank you for using the application. ###

Process finished with exit code 0
```

Fig 11

Below are some screenshots of pylint running on the program along with comments on how they were corrected.





```
def user_zipcode():
    """
    This function asks the user for their zipcode
    """
    zipcode = input("Please enter your zipcode\n(format: xxxx-xxx: ")
    if not re.fullmatch(r"(?!(0+)(0-9)(0-9)(4)?$", zipcode):
        raise ValueError
    except TypeError:
        print("Please verify zipcode.##")
    else:
        return zipcode

user_zipcode()
```

```
PS C:\Users\Corey\PycharmProjects\SDEV300\Week4> pylint glover_lab_4.py
***** Module glover_lab_4
glover_lab_4.py:13:0: W0611: Unused import sys (unused-import)

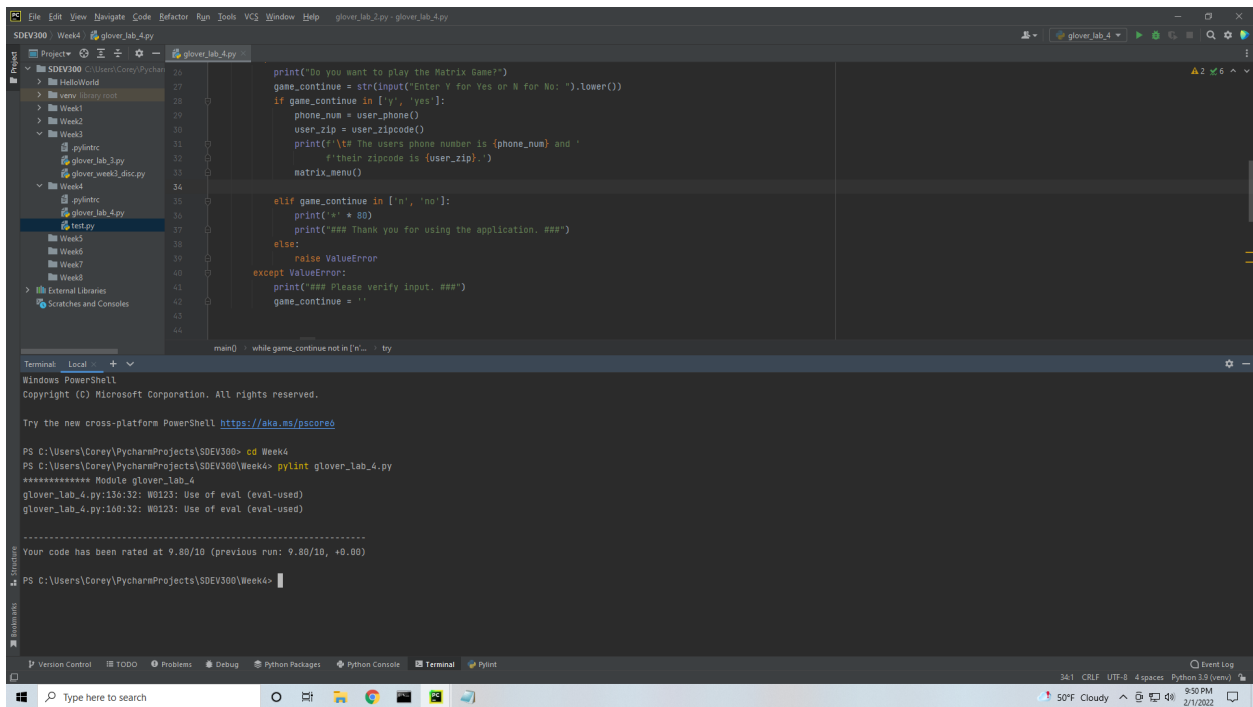
Your code has been rated at 9.74/10 (previous run: 9.74/10, +0.00)

PS C:\Users\Corey\PycharmProjects\SDEV300\Week4> pylint glover_lab_4.py
***** Module glover_lab_4
glover_lab_4.py:54:0: C0301: Line too long (83/79) (line-too-long)
glover_lab_4.py:13:0: W0611: Unused import sys (unused-import)

Your code has been rated at 9.49/10 (previous run: 9.74/10, -0.26)

PS C:\Users\Corey\PycharmProjects\SDEV300\Week4>
```

To correct the above issue the unused import was deleted and the line was backspaced until it was on the previous line then returned so Pycharm would auto-format



```
print("Do you want to play the Matrix Game?")
game_continue = str(input("Enter Y for Yes or N for No: ").lower())
if game_continue in ['y', 'yes']:
    phone_num = user_phone()
    user_zip = user_zipcode()
    print(f"The users phone number is {phone_num} and "
          f"their zipcode is {user_zip}.")
    matrix_menu()

elif game_continue in ['n', 'no']:
    print(" " * 80)
    print("## Thank you for using the application. ##")
else:
    raise ValueError
except ValueError:
    print("## Please verify input. ##")
game_continue = ''

main() while game_continue not in ['n', 'no']:
```

```
PS C:\Users\Corey\PycharmProjects\SDEV300> cd Week4
PS C:\Users\Corey\PycharmProjects\SDEV300\Week4> pylint glover_lab_4.py
***** Module glover_lab_4
glover_lab_4.py:136:32: W0123: Use of eval (eval-used)
glover_lab_4.py:160:32: W0123: Use of eval (eval-used)

Your code has been rated at 9.80/10 (previous run: 9.80/10, +0.00)

PS C:\Users\Corey\PycharmProjects\SDEV300\Week4>
```

To correct the use of eval, ast.literal\_eval was used in its place.

The screenshot shows a PyCharm IDE with a Python script named `glover_lab_4.py` and a terminal window. The script defines a menu for matrix operations (Addition, Subtraction, Multiplication, Element by Element Multiplication) and includes functions for adding, subtracting, multiplying, and transposing matrices. The terminal window shows the following output:

```

Module: glover_lab_4
glover_lab_4.py:49:8: C0301: Line too long (81/79) (line-too-long)
Your code has been rated at 9.92/10 (previous run: 10.00/10, -0.08)
PS C:\Users\Corey\PycharmProjects\SDEV300\Week4>

```

Returning a 10/10

Below are the test cases as well as some screen shots of running the encrypt program and then running the hashed passwords on a decrypt site.

Test Cases:	MD5:	SHA-256:	SHA-512:	Results:
Passw0rd	d41e98d1eafa6d6011d3a70f1a5b92f0	ab38eadaeb746599f2c1ee90f8267f31f467347462764a24d71ac1843ee77fe3	fe0d8456dd3f1a0f68cde11860c34bddce97dcbc20f389f534af8c4c49e225f6307ca16e414ac04c8d67b80821690edceb86f8de0d5286dd37ee562e3dcf2e80	Exact Match
s0ngb1rd5	e3f838a629d3246fc6fb053b09b8d271	f345d4437843f40d028f1be4c843290652a810263608b21c1c43b2d05daa3364	3c8d8ae09d0e5c1b13747bfcd1289144ecd78b42129edee7fb74c8d9a3bffdbcf638a71e400f9f4341384c45bb070b8b52944fa84d104a1946	Not Found

			04855d4310d57b	
password12345	365d38c60c4e98c a5ca6dbc02d396e 53	3700adf1f25fab 8202c1343c4b 0b4e3fec706d5 7cad57408646 7b8b3ddf273ec	fb997d5c01ebc f962d820b3b0e 7f8bfeeb7f4bd3 37cc83682f2af 90d252c20c5d 85744b7c6bb9 4f48139f690a6 1e4ad317d610 7e4310efc016d 9287266b5172 b	Exact Match
Rosco1981	3f8dde1819f1b06a 4a4e0962bdf07f9f	66083a1a680b 32812b6e62b4 96798487263c 01de058ff8948 b45f4c0fc701d 94	8c3023ba55f61 a011a391f2d2e 46f0153420a9d 564317ef2ea52 2e3cbac3f351c c5521683f10e5 76b2c1f18dece fd0081bf450c8 8f819a11728a6 c0940b4b12f	Not Found
Sara1105	efe874b3fd81dcc2 4a423526861500e 4	c90a9b9af23b1 d61b79ebaf03d caa02c5e5b69 ec1f286c54c5a b4fd399bb5060	e18e2c5e37a6 0f331f98a1cfeb b1fff639ab3150 20e22ccf51ff00 ef0e8cd1cac4a 3e7eb830962e d03596f3b24cf 29c980ea37a2 59989361f0d13 6af9c944a41	Not Found
R@nd0mP@55W0rd	d05685a5443097a c8c92437b943bf05 d	f4b11821dc70f 557f8bc95648b e7d2f829844aa d379028faad6a dc015266535c	2835228cf268e 03aae2db345f1 3f68e6108d086 183a61b2483d cc80241a5f27c ba3e67d09454 2fc999a2f774df b2d949c8c8c2 36c5fe23456d2 0fbe1ecdf919e	Not Found

123password456	b0693f29e3018e5 4886fb6f41abac11 4	389137315eeb a714d066ddb b64e0418d6f14 ed2ce6639ded 3773c29ba601 61d	f2054079c77d0 192f1bca53557 6a08236dd568 24920ef0ebea0 815055d010d6 8c5f2787fc765 792d0fd0fddd4 bff08d62e8e0c 6302ba879862f 4b0b0f69b87a0	Exact Match
Str0ngP@ssW0rd	25a86b775faaa19 d7e41efdd7864f51 6	11a53d30308c 69b489d9dfde8 4df84dcea3c23 47d4a3080589 1d5e15ab7720 91	077c17648ac4 9e47d78dfdeac 2f4c1b9ffdd771 fe6d3bd19e2a2 978feaf9f14ac 9a15deec3701 47af9a76c4157 5dd3650c351fe a38039ca3a55 d975787fcf90	Not Found
Birthday1980	c47f5cdf6461c291 8bf0671b4b192d3 a	989d346b2ef87 fbd85fd82194c 49d106dcb794 1d7dec976530 c56cc950c803b 0	a7d5a778ab60 1e12458c0203 5760d9fffee8a4 cfa2cd6fe4f8c8 ad1dfbf34a685 9e1d541c03e0 c0b312f85f5d7 d6efd51ca9b33 e26553d5ca55 c3d745a25866 2	Not Found
DogName10111998	212603722388b94 a7ba62f1a7f5c962 2	51cd279cdebd 23d8637f15cf0 1de56dd8d0a7 8db0993d37a3 3ed5f2126f04b 84	07faf70ed99a9 133156d6c45b 97d0c92945d7 726be22b26b9 93d31a248f2dd 0e52f4375f147 d019941d8f827 d8a0a3e5f3709 cf5c8dda0e3eb 78441e9dc3a8 66	Not Found

Below are some screenshots of the above passwords and the use of Crackstation.net.

Name: Corey Glover

SDEV300-6382

Date: Jan. 31, 2022

CrackStation - Online Password - x

crackstation.net

# CrackStation

Defuse.ca Twitter

CrackStation Password Hashing Security Defuse Security

## Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

c98a989af23b1d61b79ebaf93dca02c5e5b69ecf286c54c5ab4fd3999b5060

I'm not a robot

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-hex, sha1, sha224, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+ (sha1{sha1\_bin}), QubesV3.1BackupDefaults

Hash	Type	Result
c98a989af23b1d61b79ebaf93dca02c5e5b69ecf286c54c5ab4fd3999b5060	Unknown	Not found

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

[Download CrackStation's Wordlist](#)

### How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

CrackStation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list we could find. We also applied intelligent word mangling (brute force hybrid) to our wordlists to make them much more effective. For MD5 and SHA1 hashes, we have a 190GB, 15-billion-entry lookup table, and for other hashes, we have a 19GB 1.5-billion-entry lookup table.

You can download CrackStation's dictionaries [here](#), and the lookup table implementation (PHP and C) is available [here](#).

Last Modified: May 27, 2019, 8:19am UTC  
Page Hits: 40951984  
Unique Hits: 7899526

Untitled spreadsheet - Google Sheets

Assignments - SDEV 300 E

lab\_4\_report - Google Docs

doc.google.com/spreadsheets/d/182RQzTJ08ICuMhVKeApDHFqByMfHal-94d8-lUXY/edit#

File Edit View Insert Format Data Tools Extensions Help Last edit...

100% 123- Default (Alt...) 10 B I T A

	A	B	C	D	E	F	G
5			66083a1a680b	2a3bac3f351c			
6			c09a989af23b1d61b79ebaf93dca02c5e5b69ecf286c54c5ab4fd3999b5060	283c228e268e			
7			ef6874b3d81d0cc	4a423526861500e			
8							
9							

CrackStation - Online Password - x

crackstation.net

# CrackStation

Defuse.ca Twitter

CrackStation Password Hashing Security Defuse Security

## Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

ab38eadeab746599f2c1ee90f8267f31f467347462764a24d71ac1843ee77fe3

I'm not a robot

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-hex, sha1, sha224, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+ (sha1{sha1\_bin}), QubesV3.1BackupDefaults

Hash	Type	Result
ab38eadeab746599f2c1ee90f8267f31f467347462764a24d71ac1843ee77fe3	SHA256	Password

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

[Download CrackStation's Wordlist](#)

### How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

CrackStation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list we could find. We also applied intelligent word mangling (brute force hybrid) to our wordlists to make them much more effective. For MD5 and SHA1 hashes, we have a 190GB, 15-billion-entry lookup table, and for other hashes, we have a 19GB 1.5-billion-entry lookup table.

You can download CrackStation's dictionaries [here](#), and the lookup table implementation (PHP and C) is available [here](#).

Last Modified: May 27, 2019, 8:19am UTC  
Page Hits: 40951986  
Unique Hits: 7899521

Untitled spreadsheet - Google Sheets

Assignments - SDEV 300 E

lab\_4\_report - Google Docs

doc.google.com/spreadsheets/d/182RQzTJ08ICuMhVKeApDHFqByMfHal-94d8-lUXY/edit#

File Edit View Insert Format Data Tools Extensions Help Last edit...

100% 123- Default (Alt...) 10 B I T A

	A	B	C	D	E	F	G
1	Test Cases:	MD5:	SHA 256:	SHA 512:			
2							
3							
4							
5							

Name: Corey Glover

SDEV300-6382

Date: Jan. 31, 2022

CrackStation - Online Password

crackstation.net

Defuse.ca

Twitter

CrackStation Password Hashing Security Defuse Security

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

e3f838a629d3246fc6fb053b09b8d271

I'm not a robot

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-hex, sha1, sha224, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+ (sha1[sha1\_bin]), QubesV3.1BackupDefaults

Hash

Type

Result

e3f838a629d3246fc6fb053b09b8d271

Unknown

Not Found

Color Codes: Exact match, Partial match, Not found.

Download CrackStation's Wordlist

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

CrackStation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list we could find. We also applied intelligent word mangling (brute force hybrid) to our wordlists to make them much more effective. For MD5 and SHA1 hashes, we have a 190GB, 15-billion-entry lookup table, and for other hashes, we have a 19GB 1.5-billion-entry lookup table.

You can download CrackStation's dictionaries [here](#), and the lookup table implementation (PHP and C) is available [here](#).

Last Modified: May 27, 2019, 8:19am UTC

Page Hits: 40951966

Unique Hits: 7899524

Untitled spreadsheet - Google

Assignments - SDEV 300 E

lab\_4\_report - Google Docs

Share

File Edit View Insert Format Data Tools Extensions Help Last edit...

100% 123+ Default (Aut...) 10 B I A

Test Cases:

MD5:

SHA 256:

SHA 512:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1