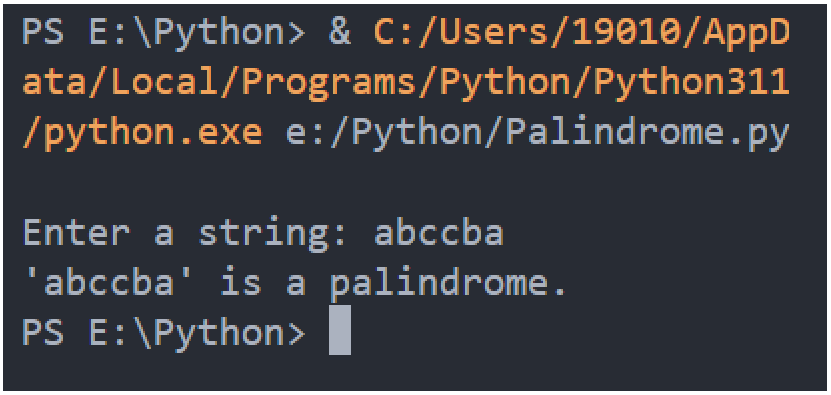
Name:Joydip Das
ID:190102

Problem 1 source code:

```
def is_palindrome(s):
    return s == s[::-1]

t = input("Enter a string: ")

if is_palindrome(t):
    print(f"'{t}' is a palindrome.")
else:
    print(f"'{t}' is not a palindrome.")
```

Output:

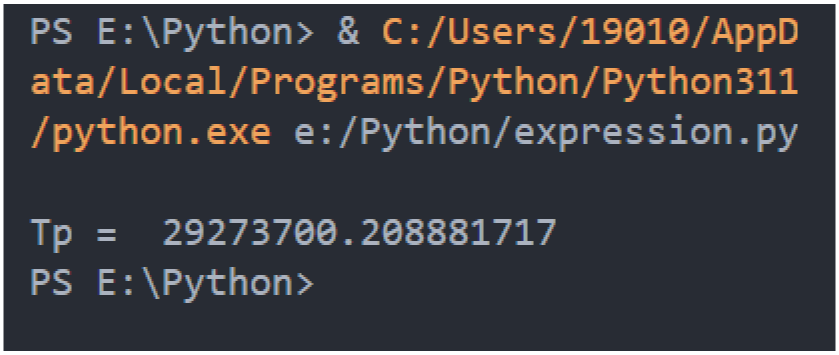


```
PS E:\Python> & C:/Users/19010/AppData/Local/Programs/Python/Python311/python.exe e:/Python/Palindrome.py

Enter a string: abccba
'abccba' is a palindrome.
PS E:\Python> 
```

Problem 2 source code:

```
alpha = 0.306
ts = 6.96e8
rs = 6.96e8
d = 1.496e11
beta = 1.2

f = (1-alpha)/beta
sec= rs * pow(f , 0.5)
th = sec / (2.0 * d)
tp = ts * pow(th , 0.5);
print("Tp = " , tp)
```

Output:

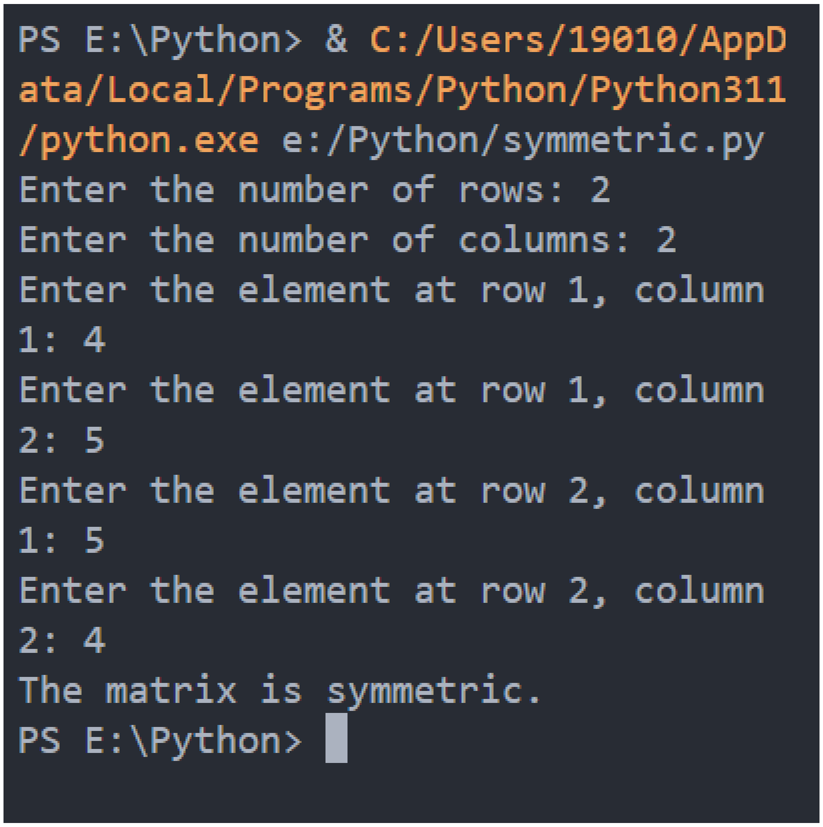


```
PS E:\Python> & C:/Users/19010/AppD
ata/Local/Programs/Python/Python311
/python.exe e:/Python/expression.py

Tp =  29273700.208881717
PS E:\Python>
```

Problem 3 source code:

```
def is_symmetric(matrix):
    return all(matrix[i][j] == matrix[j][i] for i in range(len(matrix)) for j in range(len(matrix[0])))

def input_matrix(rows, cols):
    matrix = []
    for i in range(rows):
        row = []
        for j in range(cols):
            element = float(input(f"Enter the element at row {i + 1}, column {j + 1}: "))
            row.append(element)
        matrix.append(row)
    return matrix

rows = int(input("Enter the number of rows: "))
cols = int(input("Enter the number of columns: "))

user = input_matrix(rows, cols)

if is_symmetric(user):
    print("The matrix is symmetric.")
else:
    print("The matrix is not symmetric.")
```

Output:

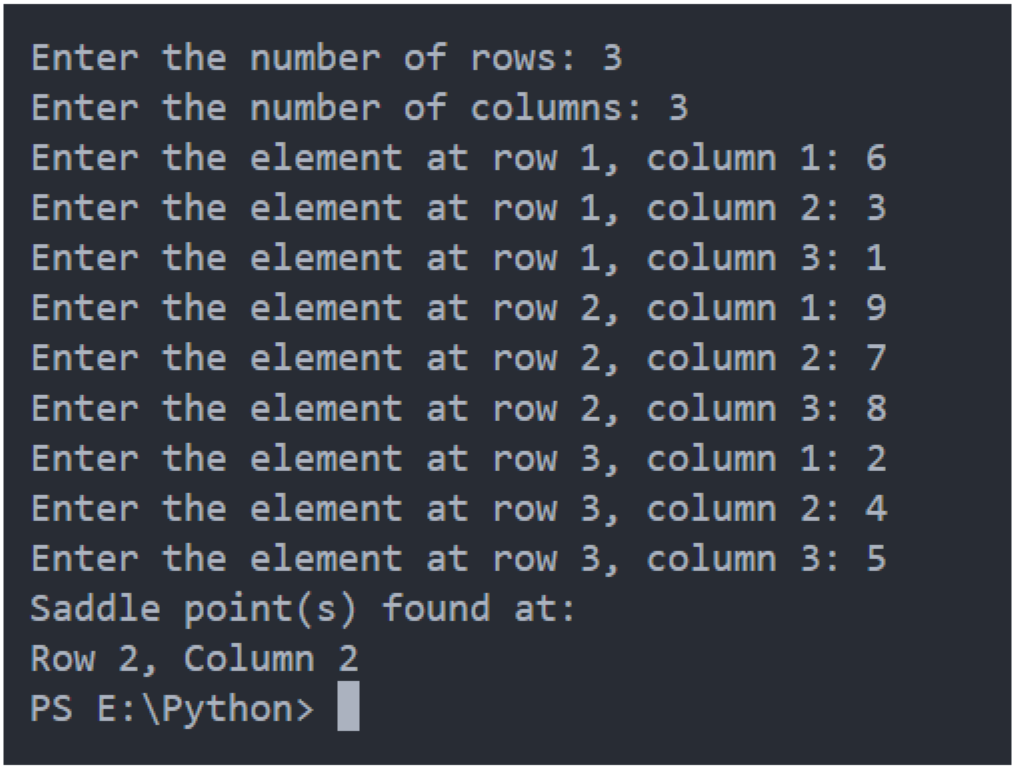


```
PS E:\Python> & C:/Users/19010/AppData/Local/Programs/Python/Python311/python.exe e:/Python/symmetric.py
Enter the number of rows: 2
Enter the number of columns: 2
Enter the element at row 1, column 1: 4
Enter the element at row 1, column 2: 5
Enter the element at row 2, column 1: 5
Enter the element at row 2, column 2: 4
The matrix is symmetric.
PS E:\Python> 
```

Problem 4 source code:

```
def find_saddle(matrix):
    saddle_points = []

    for i in range(len(matrix)):
        for j in range(len(matrix[0])):
            element = matrix[i][j]

            is_min_in_row = all(element <= matrix[i][k] for k in range(len(matrix[0])))

            is_max_in_col = all(element >= matrix[k][j] for k in range(len(matrix)))

            if is_min_in_row and is_max_in_col:
                saddle_points.append((i, j))

    return saddle_points

def input_matrix(rows, cols):
    matrix = []
    for i in range(rows):
        row = []
        for j in range(cols):
            element = int(input(f"Enter the element at row {i + 1}, column {j + 1}: "))
            row.append(element)
        matrix.append(row)
    return matrix

rows = int(input("Enter the number of rows: "))
cols = int(input("Enter the number of columns: "))

user_matrix = input_matrix(rows, cols)

saddle_points = find_saddle(user_matrix)

if saddle_points:
    print("Saddle point(s) found at:")
    for point in saddle_points:
        print(f"Row {point[0] + 1}, Column {point[1] + 1}")
else:
    print("No saddle points found in the matrix.")
```

Output:

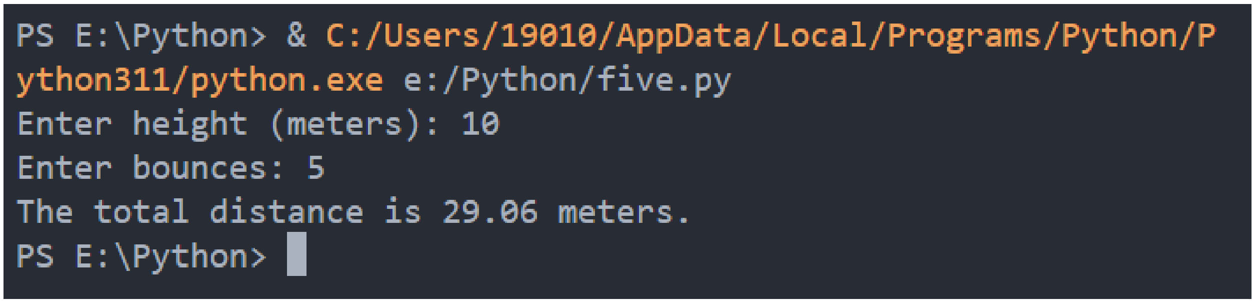


```
Enter the number of rows: 3
Enter the number of columns: 3
Enter the element at row 1, column 1: 6
Enter the element at row 1, column 2: 3
Enter the element at row 1, column 3: 1
Enter the element at row 2, column 1: 9
Enter the element at row 2, column 2: 7
Enter the element at row 2, column 3: 8
Enter the element at row 3, column 1: 2
Enter the element at row 3, column 2: 4
Enter the element at row 3, column 3: 5
Saddle point(s) found at:
Row 2, Column 2
PS E:\Python> 
```

Problem 5 Source code:

```
def total_distance_traveled(h, n):
    g = 9.8
    total_distance = 0

    for _ in range(n):
        total_distance += h
        h /= 2
        total_distance += h

    return total_distance

h = float(input("Enter height (meters): "))
n = int(input("Enter bounces: "))

if h <= 0 or n < 1:
    print("Enter positive values.")
else:
    total_dist = total_distance_traveled(h, n)
    print(f"The total distance is {total_dist:.2f} meters.")
```

Output:



```
PS E:\Python> & C:/Users/19010/AppData/Local/Programs/Python/Python311/python.exe e:/Python/five.py
Enter height (meters): 10
Enter bounces: 5
The total distance is 29.06 meters.
PS E:\Python> 
```