In [16]:
```python
# Question 2a

import numpy as np
A = np.array([[25,0,1],[20,1,2],[40,1,6]])
b = np.array([[110],[110],[210]])

# To see rank, use:
# np.linalg.matrix_rank(A)

# To invert a matrix, use:
# np.linalg.inv(A)

# To multiply matrices in Python 3, use:
# A@B


# Check if A is invertible
if np.linalg.matrix_rank(A) == A.shape[0]:
    # Solve using direct inversion
    x = np.linalg.inv(A) @ b
    print("Solution:\n", x)
```

```
Solution:
 [[ 4.25]
 [17.5 ]
 [ 3.75]]
```

In [15]:
```python
# Question 2b
import numpy as np
A = np.array([[25,15,10,0,1],[20,12,8,1,2],[40,30,10,1,6], [30,15,15,0,3], [
b = np.array([[104],[97],[193],[132],[174]])

print(np.linalg.matrix_rank(A))

# Note: you can use np.hstack() to concatinate vectors, for example np.hstac
print(np.linalg.matrix_rank(np.hstack((A,b))))
print("There is an exact solution")
# Note: you can select all the columns, except the first of a matrix A as:
print('There is not a unique solution because the rank of A is not 5')

A = np.array([[20,12,8,1,2],[40,30,10,1,6], [30,15,15,0,3], [35,20,15,2,4]])
b = np.array([[97],[193],[132],[174]])

print(np.linalg.matrix_rank(A))

print("Unique solution")
print(np.linalg.inv(A.T@A)@A.T@b)
```

```
4
4
There is an exact solution
There is not a unique solution because the rank of A is not 5
4
Unique solution
[[ 1.21371651]
 [ 7.63039488]
 [14.31664595]
 [ 9.        ]
 [ 4.        ]]
```