Assignment 1

21AIE303

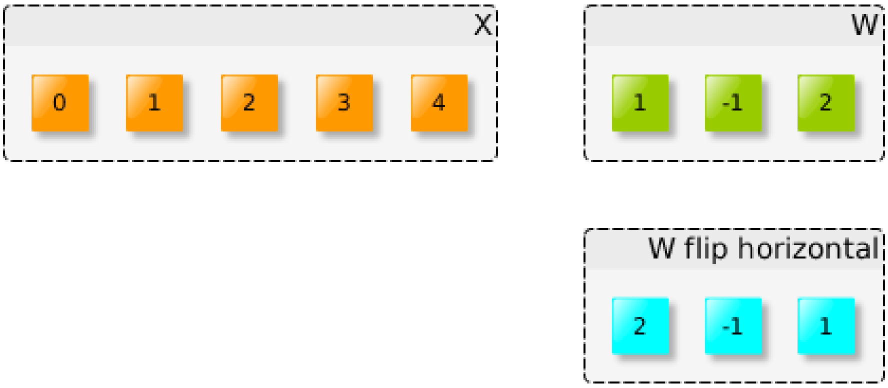
Signal and Image Processing – SEM-V

Professor – Dr. Sachin Kumar

Submitted By: Vikhyat Bansal CB.EN.U4AIE21076

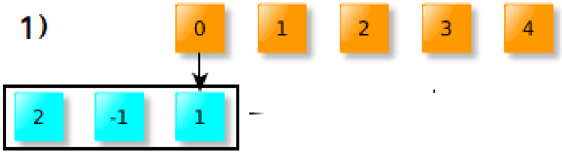
AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Idea of Implementation: Question



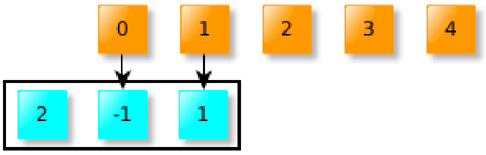
Answer:

1)



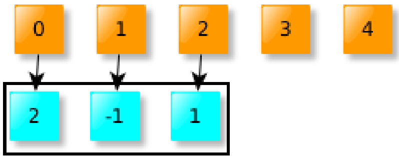
(1x0) = 0

2)



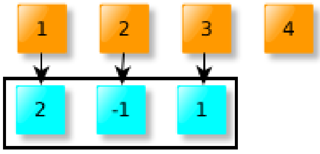
(1x1)+(-1x0) = 1

3)



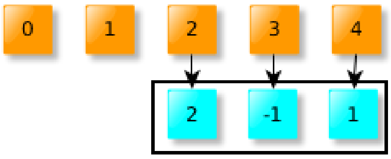
(1x2)+(-1x1)+(2x0) = 1

4)



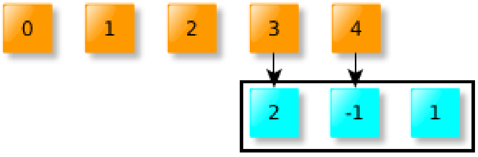
(1x3)+(-1x2)+(2x1) = 3

5)



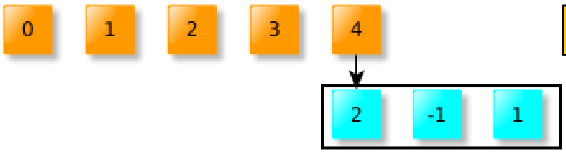
(1x4)+(-1x3)+(2x2) = 5

6)



(2x3)+(-1x4) = 2

7)



(2x4) = 8

1. CODE: {Verfication using In-Built – Python}

```python
import numpy as np
np.convolve(list1,list2)
```

CODE: {Verfication using In-Built – MATLAB}

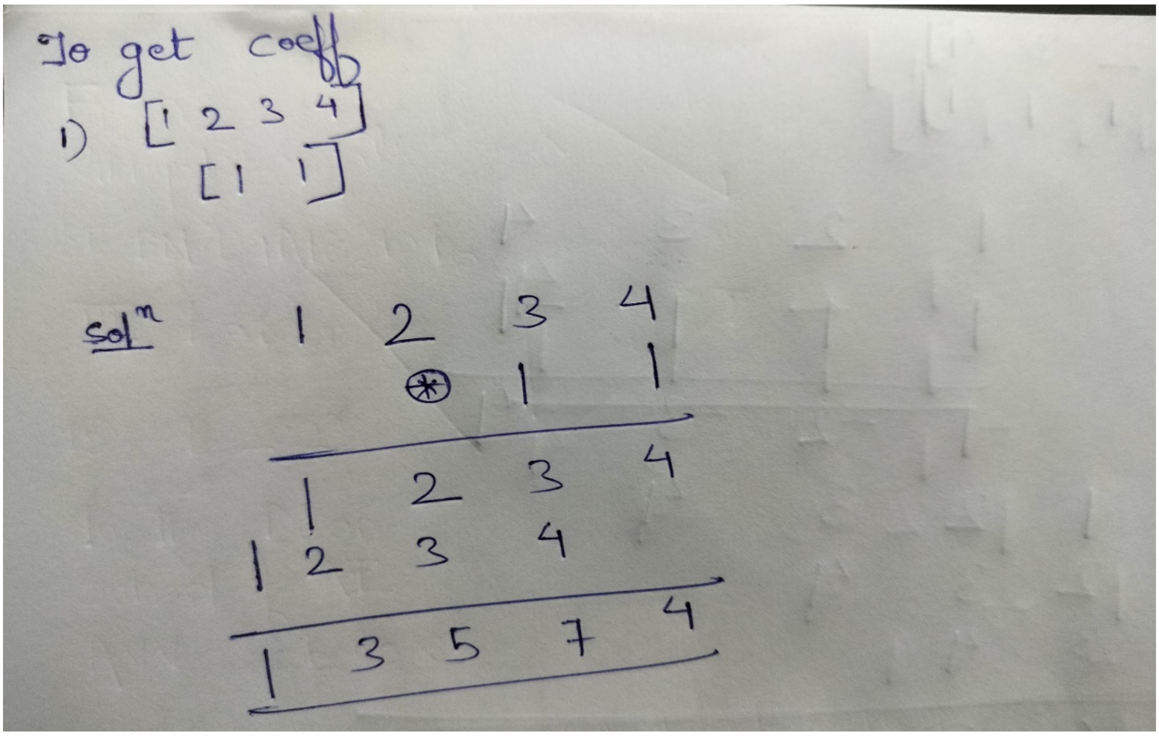
```matlab
conv(list1,list2)
```

Verify the workout examples using InBuilt Function:

a.) List 1 = [1 2 3 4]

List 2 = [1 1]

Handwritten:

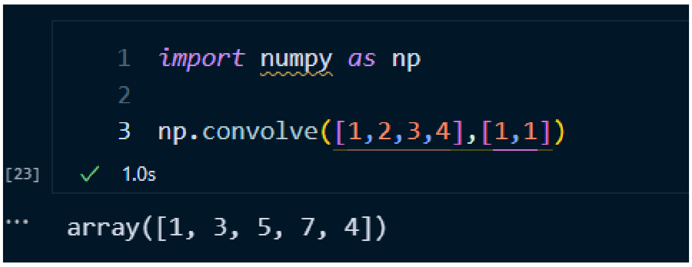


To get coeff

1) [1 2 3 4]
   [1 1]

```
Solⁿ        1   2   3   4
                ⊛   1   1
         ------------------
            1   2   3   4
        1   2   3   4
       --------------------
        1   3   5   7   4
```

Python:

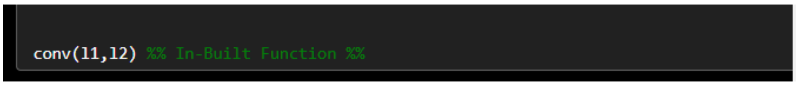


```python
import numpy as np

np.convolve([1,2,3,4],[1,1])
```

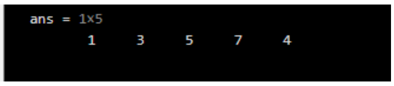
[23] ✓ 1.0s

```
array([1, 3, 5, 7, 4])
```

MATLAB:



```matlab
conv(l1,l2) %% In-Built Function %%
```

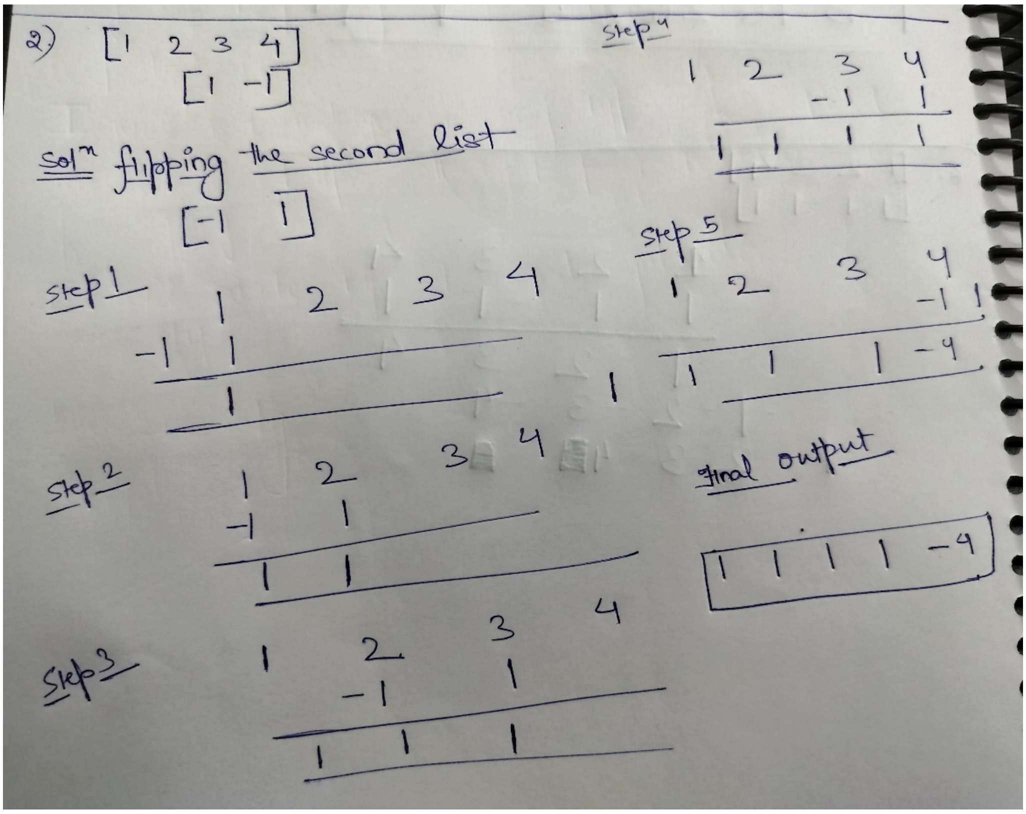


```
ans = 1x5
     1     3     5     7     4
```

b.) List 1 = [1 2 3 4]

List 2 = [1 -1]

Handwritten:



2) [1 2 3 4]
[1 -1]

Sol^n flipping the second list

[-1 1]

Step 1

```
    1  2  3  4
-1  1
    1
```

Step 2

```
 1  2  3  4
-1  1
 1  1
```

Step 3

```
 1  2  3  4
   -1  1
 1  1  1
```

Step 4

```
 1  2  3  4
      -1  1
 1  1  1  1
```

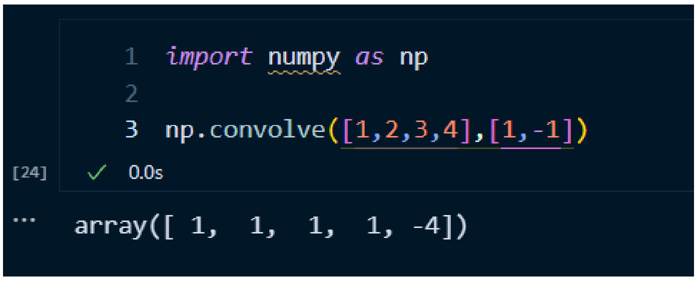
Step 5

```
 1  2  3  4
         -1  1
 1  1  1  1  -4
```

final output

```
1  1  1  1  -4
```

Python:

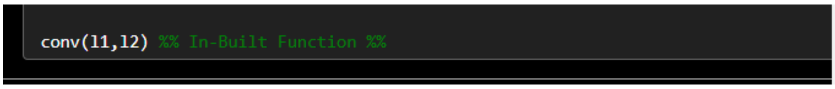


```
import numpy as np

np.convolve([1,2,3,4],[1,-1])
```

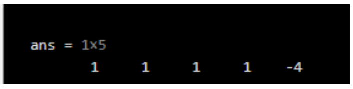
```
[24] ✓ 0.0s
```

```
array([ 1,  1,  1,  1, -4])
```

MATLAB:



```
conv(l1,l2) %% In-Built Function %%
```

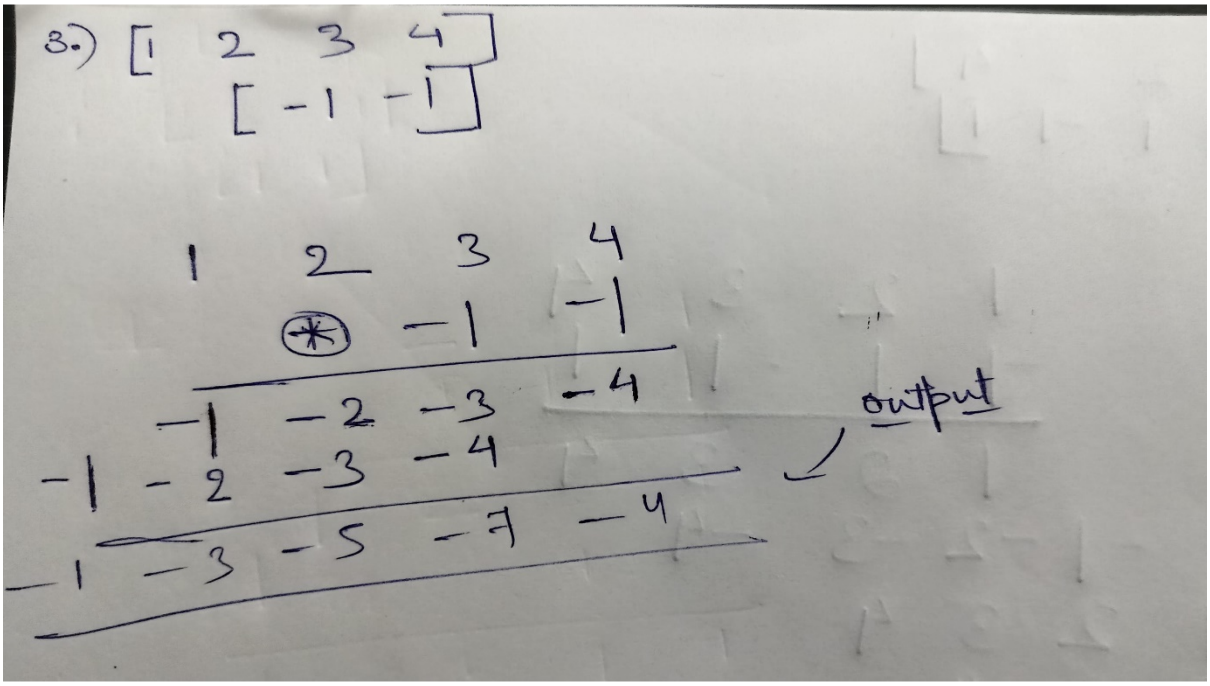


```
ans = 1x5
     1     1     1     1    -4
```

c.) List 1 = [1 2 3 4]

List 2 = [-1 -1]

Handwritten:



3.) [1 2 3 4]

[-1 -1]

1 2 3 4

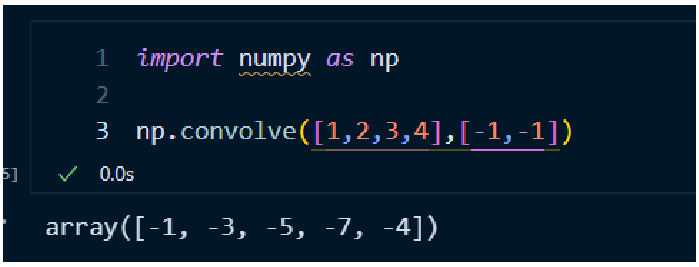
⊛ -1 -1

-1 -2 -3 -4

-1 -2 -3 -4

-1 -3 -5 -7 -4 ✓ output

Python:

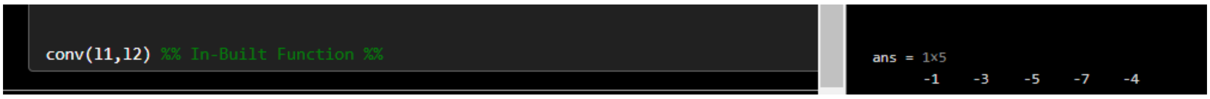


```
import numpy as np

np.convolve([1,2,3,4],[-1,-1])
```

✓ 0.0s

```
array([-1, -3, -5, -7, -4])
```

MATLAB:

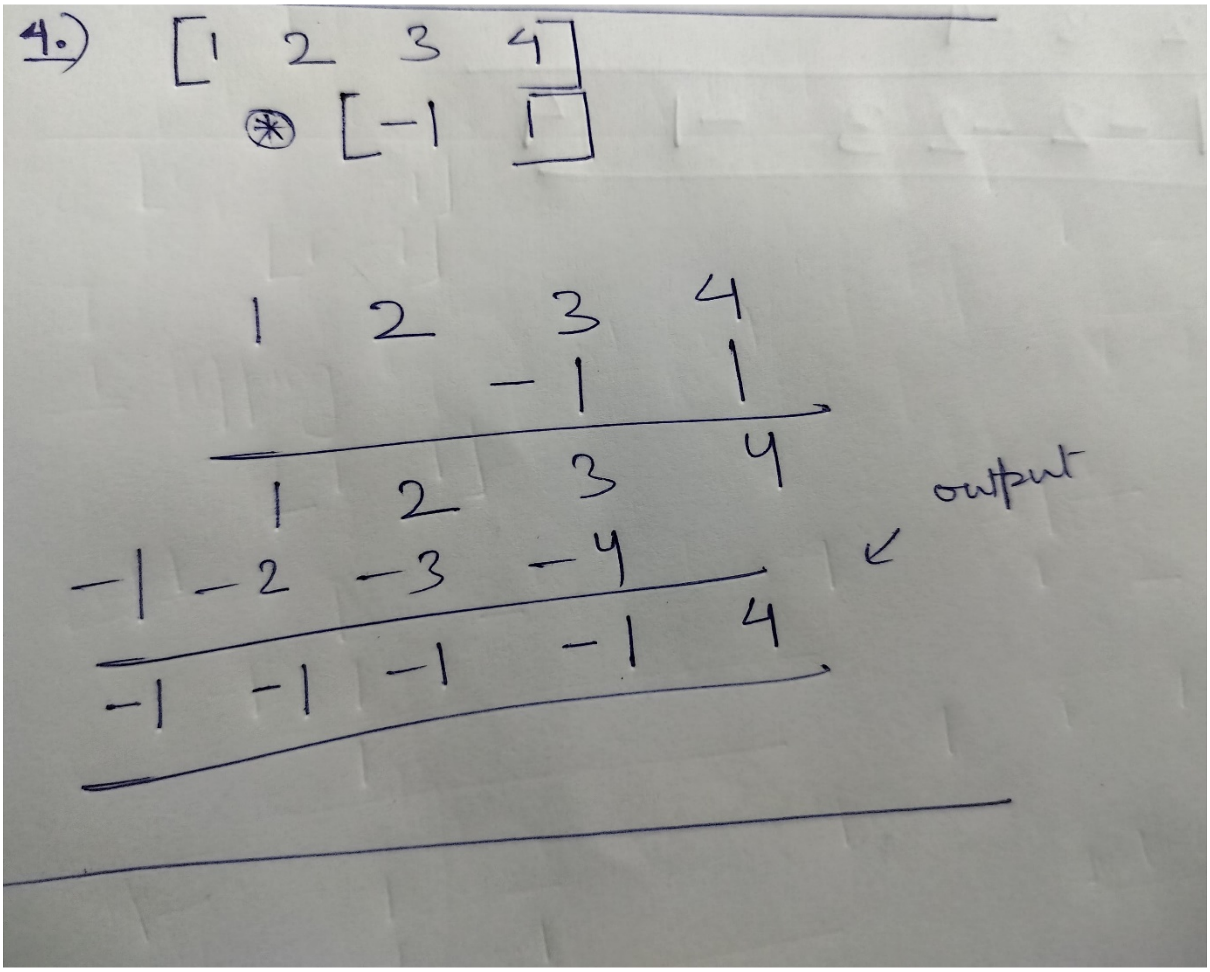


```
conv(l1,l2) %% In-Built Function %%
```

```
ans = 1x5
    -1    -3    -5    -7    -4
```

d.) List 1 = [1 2 3 4]
List 2 = [-1 1]

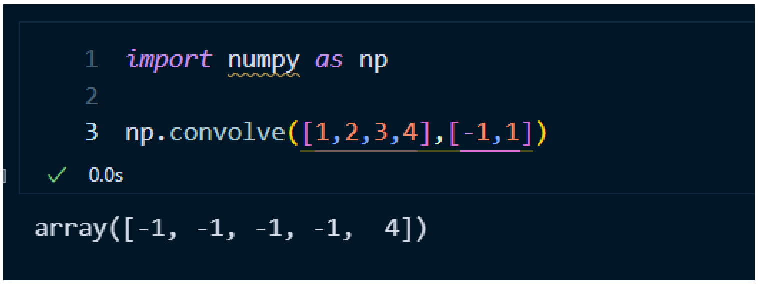
Handwritten:



```
4.)   [1   2   3   4]
        ⊛ [-1   1]

         1    2    3    4
                  -1    1
      -----------------------
         1    2    3    4
   -1   -2   -3   -4
   ---------------------------
   -1   -1   -1   -1    4
```

↙ output

Python:

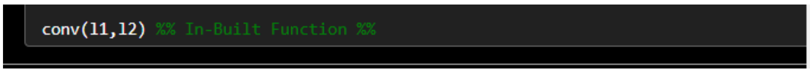


```
import numpy as np

np.convolve([1,2,3,4],[-1,1])
```

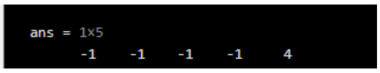
✓ 0.0s

```
array([-1, -1, -1, -1,  4])
```

MATLAB:



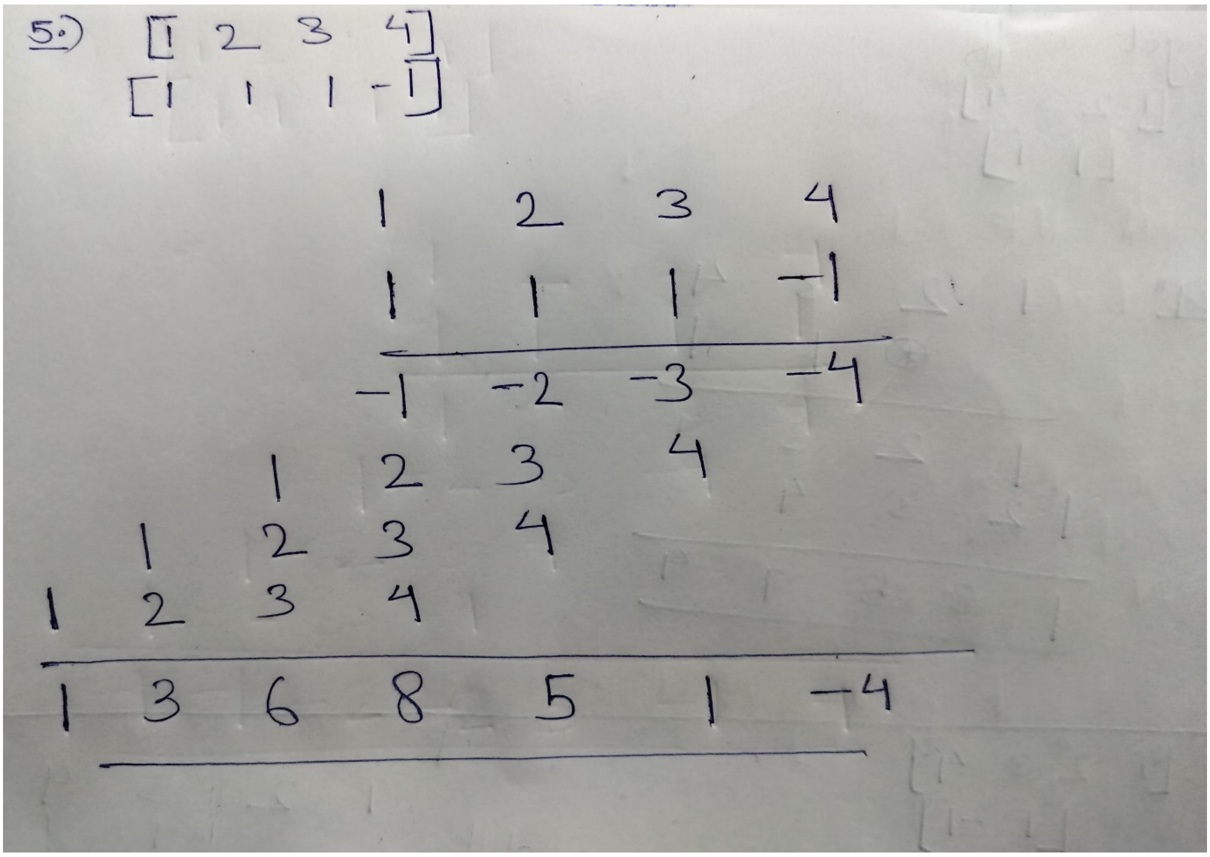
```
conv(l1,l2) %% In-Built Function %%
```



```
ans = 1x5
    -1    -1    -1    -1     4
```

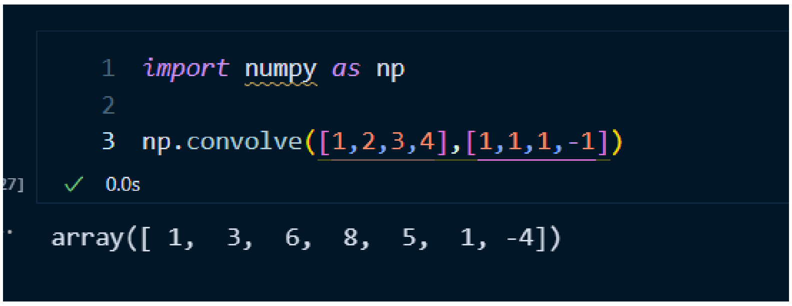
e.) List 1 = [1 2 3 4]
List 2 = [1 1 1 -1]

Handwritten:



```
5.)  [1  2  3  4]
    [1  1  1  -1]

                 1    2    3    4
                 1    1    1   -1
               ----------------------
                -1   -2   -3   -4
            1    2    3    4
       1    2    3    4
  1    2    3    4
 ------------------------------------
  1    3    6    8    5    1   -4
 ------------------------------------
```

Python:

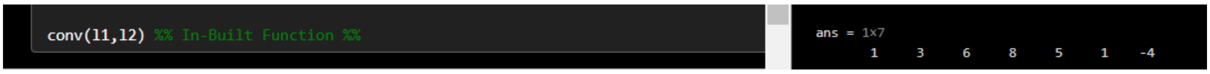


```
import numpy as np

np.convolve([1,2,3,4],[1,1,1,-1])
```

```
✓ 0.0s
```

```
array([ 1,  3,  6,  8,  5,  1, -4])
```

MATLAB:

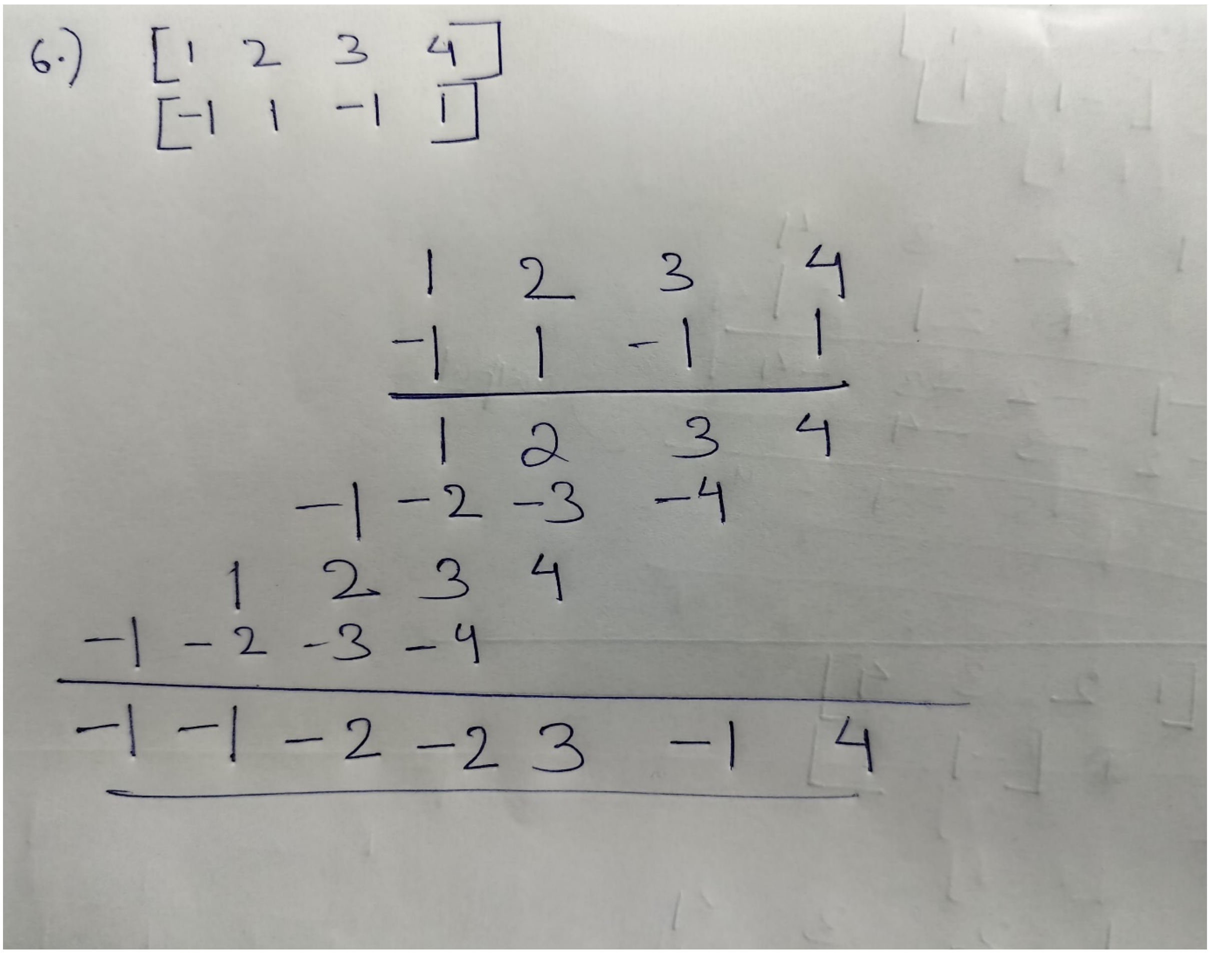


```
conv(l1,l2) %% In-Built Function %%
```

```
ans = 1x7
     1     3     6     8     5     1    -4
```

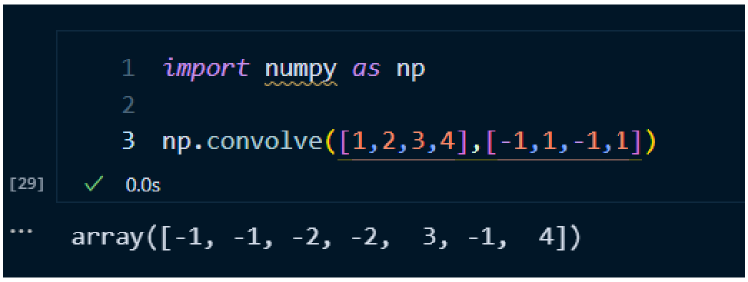
f.) List 1 = [1 2 3 4]
List 2 = [-1 1 -1 1]

Handwritten:



```
6.)  [1  2  3  4]
     [-1  1  -1  1]

                1    2    3    4
               -1    1   -1    1
              ---------------------
                1    2    3    4
          -1   -2   -3   -4
      1    2    3    4
 -1  -2   -3   -4
--------------------------------------
 -1  -1   -2   -2    3   -1    4
```

Python:

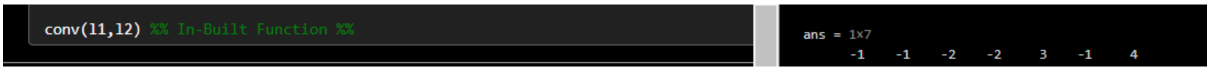


```
import numpy as np

np.convolve([1,2,3,4],[-1,1,-1,1])
```

[29] ✓ 0.0s

```
array([-1, -1, -2, -2,  3, -1,  4])
```

MATLAB:



```
conv(l1,l2) %% In-Built Function %%
```

```
ans = 1x7
    -1    -1    -2    -2     3    -1     4
```

2. Implement code for convolution and verify using examples given.

CODE: {Self Implemented – Python}

```
def convolution(list1,list2):
    # Getiing the length of the resultant list an flipping the
second list
    n = len(list1) + len(list2) -1
    coeffs = [0]*n
    list2 = list2[::-1]

    # Creating a new list with entries as zero of length(list2)-1
    newlist1 = [0]*(len(list2)-1)

# Adding the initial input list1 to the newlist1 just to get the
correct entries.
    for i in range(len(list1)):
        newlist1.append(list1[i])

    # Adding more zeroes to newlist just to deal with protruding
#part of list2 when flipped.

    for j in range(len(list2)-1):
        newlist1.append(0)

    # Using Flipping Convolution
    for i in range(n):
        for j in range(len(list2)):
            coeffs[i] += newlist1[i+j]*list2[j]

    return coeffs
```

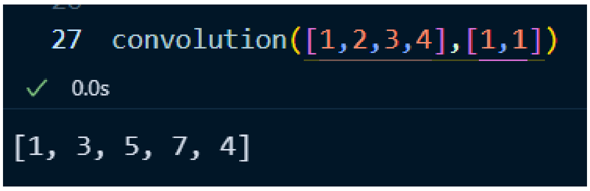
Code {Self Implemented – MATLAB}

```
function coefs = convolution(l1,l2)

    n = length(l1) + length(l2) - 1;  %% Length of final list %%
    coefs = zeros(1,n);               %% Padding of zeros on empty list %%
    l2 = flip(l2);                    %% Flipping second list %%

    newlist1 = zeros(1,length(l2)-1);  %% Same as python code%%

    for i=1:length(l1)
        newlist1(length(l2)-1+i) = l1(i);
    end

    for j=1:length(l2)-1
        newlist1(length(newlist1)+j) = 0;
    end

    for i=1:n    %% Traversing list1 using list2 and updating final coefs list
        for j=1:length(l2)
            coefs(i) = coefs(i) + newlist1(i+j-1)*l2(j);
        end
    end

end
```

Verify the implemented code using Examples:

a.) List 1 = [1 2 3 4]
List 2 = [1 1]

Python:

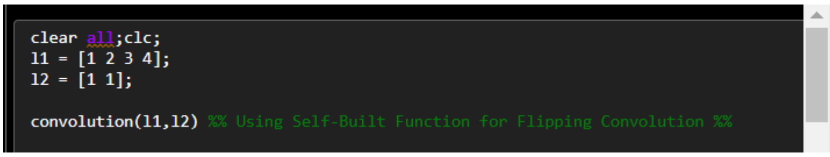


```
convolution([1,2,3,4],[1,1])
```

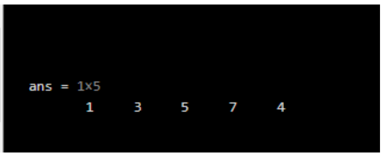
✓ 0.0s

```
[1, 3, 5, 7, 4]
```

MATLAB:



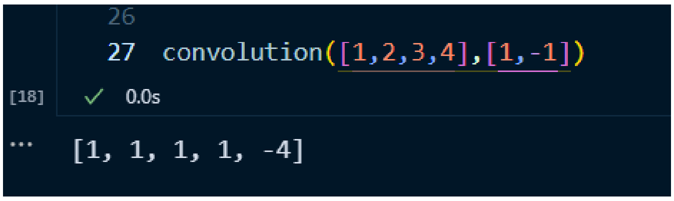
```
clear all;clc;
l1 = [1 2 3 4];
l2 = [1 1];

convolution(l1,l2) %% Using Self-Built Function for Flipping Convolution %%
```



```
ans = 1x5
     1     3     5     7     4
```

b.) List 1 = [1 2 3 4]
List 2 = [1 -1]

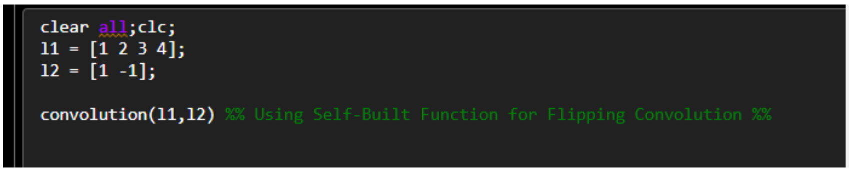
Python:



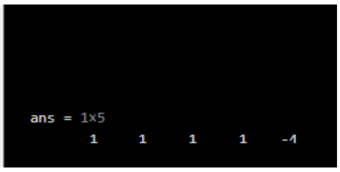
```
convolution([1,2,3,4],[1,-1])
```

```
[1, 1, 1, 1, -4]
```

MATLAB:



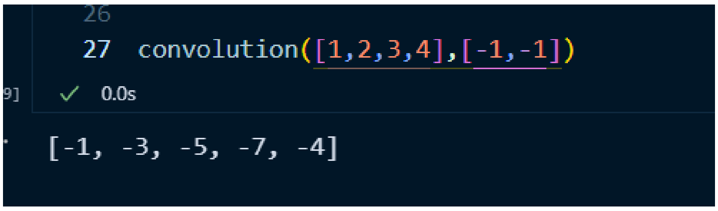
```
clear all;clc;
l1 = [1 2 3 4];
l2 = [1 -1];

convolution(l1,l2) %% Using Self-Built Function for Flipping Convolution %%
```



```
ans = 1x5
     1     1     1     1    -4
```

c.) List 1 = [1 2 3 4]
List 2 = [-1 -1]

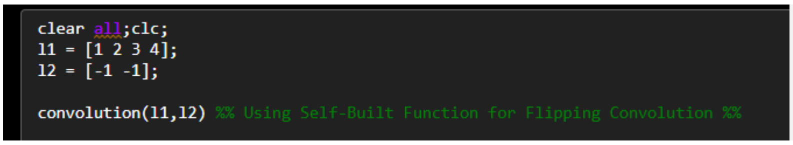
Python:



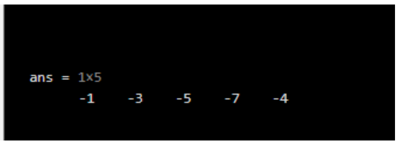
```
convolution([1,2,3,4],[-1,-1])
```

```
[-1, -3, -5, -7, -4]
```

MATLAB:



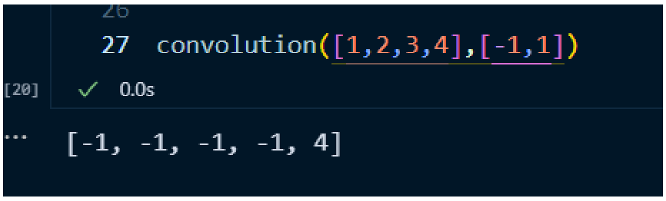
```
clear all;clc;
l1 = [1 2 3 4];
l2 = [-1 -1];

convolution(l1,l2) %% Using Self-Built Function for Flipping Convolution %%
```



```
ans = 1x5
    -1    -3    -5    -7    -4
```

d.) List 1 = [1 2 3 4]
List 2 = [-1 1]

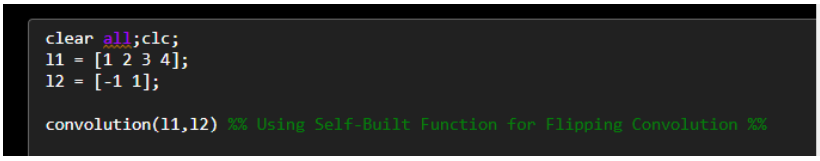
Python:



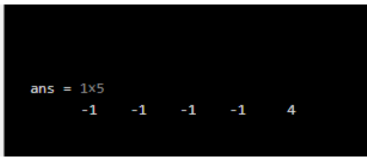
```
convolution([1,2,3,4],[-1,1])
```

```
[-1, -1, -1, -1, 4]
```

MATLAB:



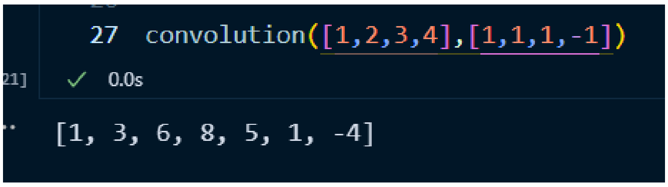
```
clear all;clc;
l1 = [1 2 3 4];
l2 = [-1 1];

convolution(l1,l2) %% Using Self-Built Function for Flipping Convolution %%
```



```
ans = 1x5
    -1    -1    -1    -1     4
```

e.) List 1 = [1 2 3 4]
List 2 = [1 1 1 -1]

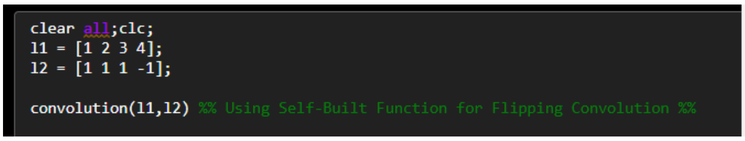
Python:



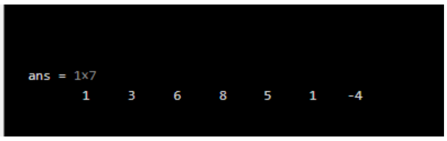
```
convolution([1,2,3,4],[1,1,1,-1])
```

```
[1, 3, 6, 8, 5, 1, -4]
```

MATLAB:



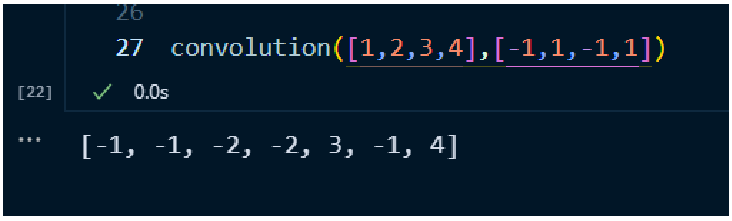
```
clear all;clc;
l1 = [1 2 3 4];
l2 = [1 1 1 -1];

convolution(l1,l2) %% Using Self-Built Function for Flipping Convolution %%
```



```
ans = 1x7
     1     3     6     8     5     1    -4
```

f.) List 1 = [1 2 3 4]
List 2 = [-1 1 -1 1]

Python:

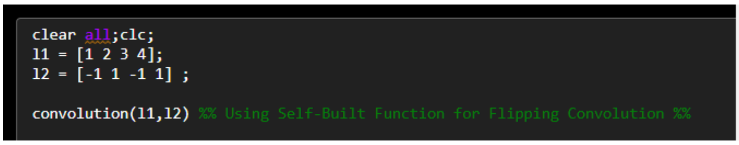


```
convolution([1,2,3,4],[-1,1,-1,1])
```

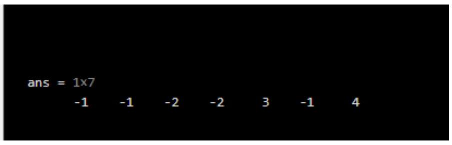
[22] ✓ 0.0s

```
[-1, -1, -2, -2, 3, -1, 4]
```

MATLAB:



```
clear all;clc;
l1 = [1 2 3 4];
l2 = [-1 1 -1 1] ;

convolution(l1,l2) %% Using Self-Built Function for Flipping Convolution %%
```



```
ans = 1x7
    -1    -1    -2    -2     3    -1     4
```

THANK YOU