Name - Ravi kumar

Reg - 2020Pgcaca72

In [1]:

```
pip install -U scikit-fuzzy
```

```
Collecting scikit-fuzzy
  Downloading scikit-fuzzy-0.4.2.tar.gz (993 kB)
Requirement already satisfied: numpy>=1.6.0 in c:\users\student\anaconda3\li
b\site-packages (from scikit-fuzzy) (1.20.3)
Requirement already satisfied: scipy>=0.9.0 in c:\users\student\anaconda3\li
b\site-packages (from scikit-fuzzy) (1.7.1)
Requirement already satisfied: networkx>=1.9.0 in c:\users\student\anaconda3
\lib\site-packages (from scikit-fuzzy) (2.6.3)
Building wheels for collected packages: scikit-fuzzy
  Building wheel for scikit-fuzzy (setup.py): started
  Building wheel for scikit-fuzzy (setup.py): finished with status 'done'
  Created wheel for scikit-fuzzy: filename=scikit_fuzzy-0.4.2-py3-none-any.w
hl size=894089 sha256=ca3df78b6b9815c47ce6b4244cd15de5ae02e92924fe756ab29213
87a74a0952
  Stored in directory: c:\users\student\appdata\local\pip\cache\wheels\32\2c
\a1\a90a7d7dd8448ec029f298a61f3490275e99b17aa348be675c
Successfully built scikit-fuzzy
Installing collected packages: scikit-fuzzy
Successfully installed scikit-fuzzy-0.4.2
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
import numpy as np
import skfuzzy as fuz
import matplotlib.pyplot as plt
```

In [7]:

```
x = np.linspace(start = 0, stop=75, num =75,endpoint = True, retstep = False)
a1 =[0,25,50]
b1 = [25,50,75]
```

Tringular membership function

In [11]:

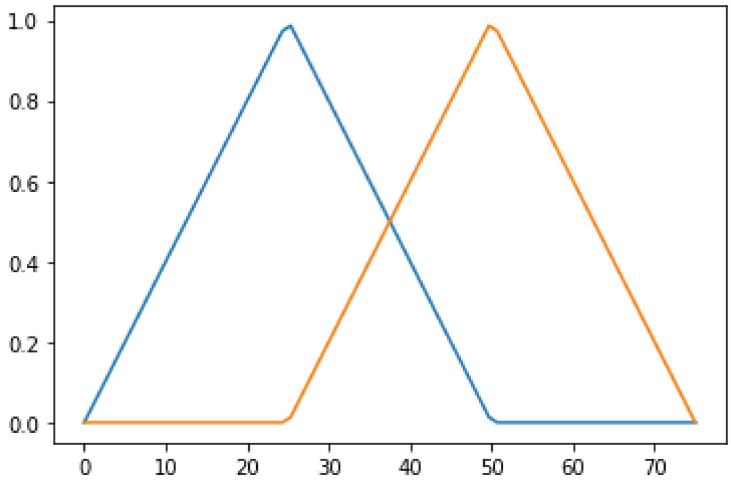
```
tri_a1 = fuz.membership.trimf(x,a1)
tri_b1 = fuz.membership.trimf(x,b1)
one= np.ones(75)
zero=np.zeros(75)
union= fuz.fuzzy_or(x,tri_a1,x,tri_b1)
intersection = fuz.fuzzy_and(x,tri_a1,x,tri_b1)
complement = fuz.fuzzy_not(tri_a1)
alg_prod = tri_a1 * tri_b1
```

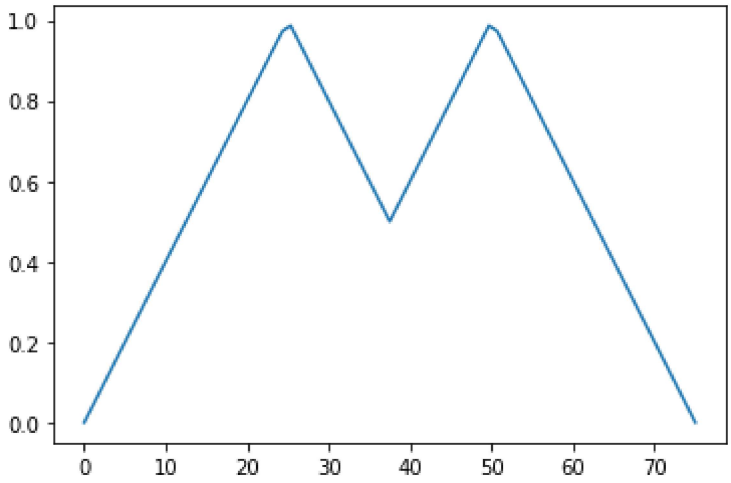
In [19]:

```
plt.figure()
plt.plot(x,tri_a1)
plt.plot(x,tri_b1)
plt.figure()
plt.plot(x,union[1])
```

Out[19]:

```
[<matplotlib.lines.Line2D at 0x4a44409af0>]
```





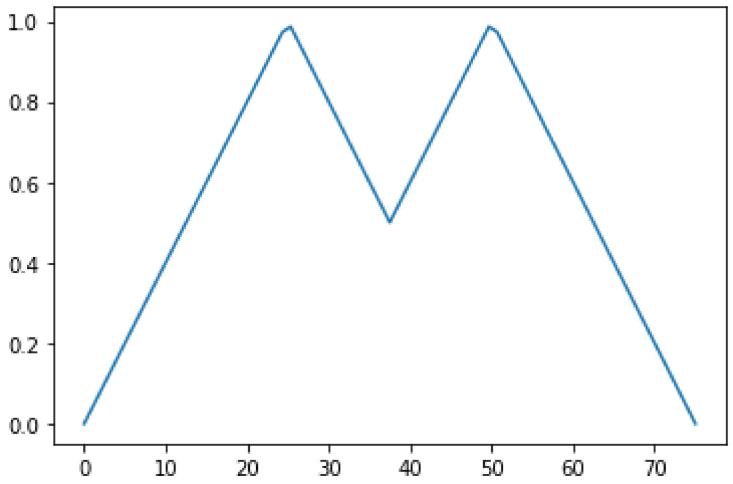
In [ ]:

In [14]:

```
plt.plot(x,union[1])
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x4a44209fd0>]
```

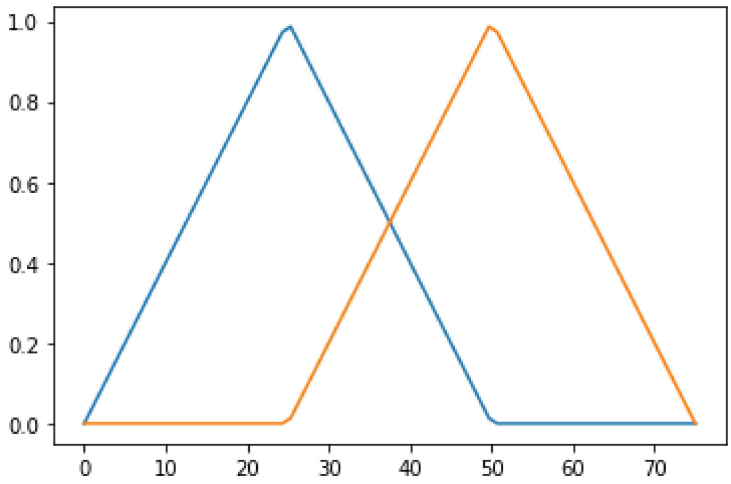


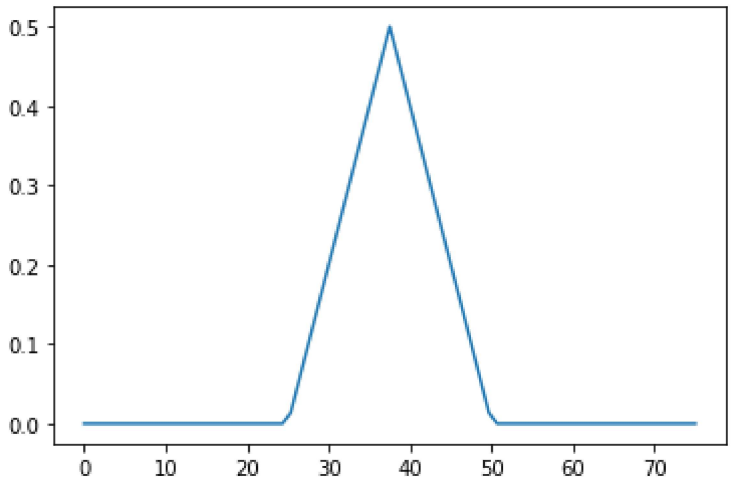
In [18]:

```
plt.plot(x,tri_a1)
plt.plot(x,tri_b1)
plt.figure()
plt.plot(x,intersection[1])
```

Out[18]:

```
[<matplotlib.lines.Line2D at 0x4a44399970>]
```





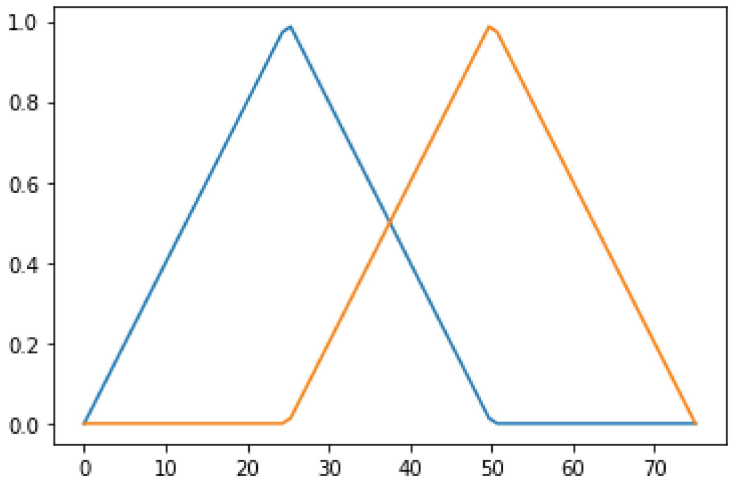
In [ ]:

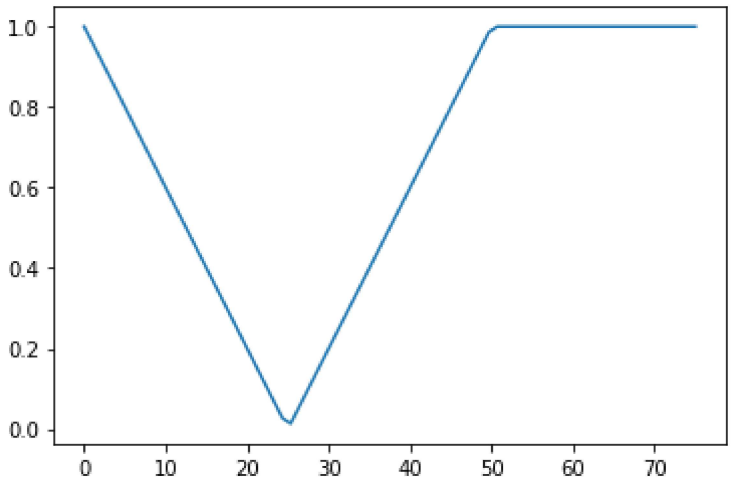
In [20]:

```
plt.plot(x,tri_a1)
plt.plot(x,tri_b1)
plt.figure()
plt.plot(x,complement)
```

Out[20]:

```
[<matplotlib.lines.Line2D at 0x4a44477970>]
```



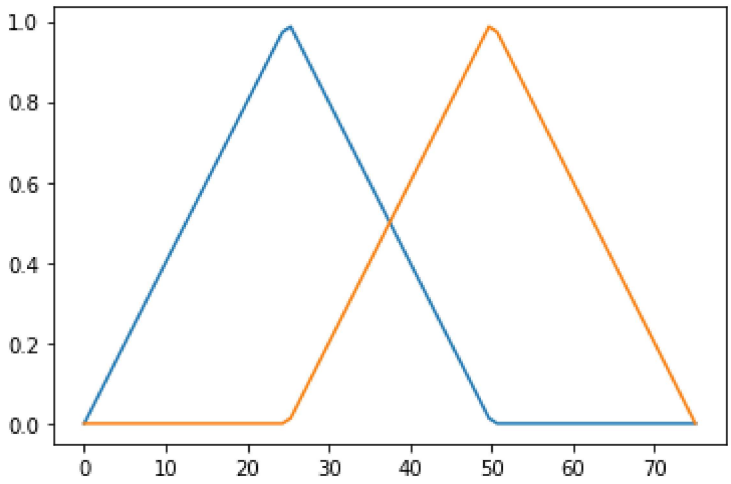


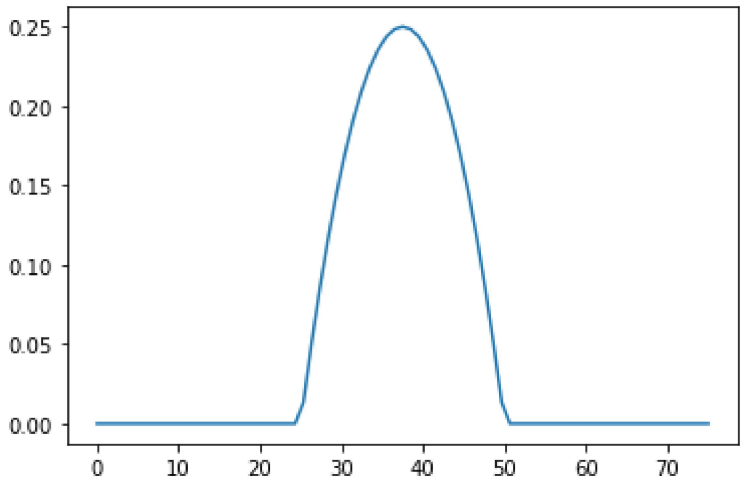
In [21]:

```
plt.plot(x,tri_a1)
plt.plot(x,tri_b1)
plt.figure()
plt.plot(x,alg_prod)
```

Out[21]:

```
[<matplotlib.lines.Line2D at 0x4a454a6970>]
```





Pi- function

In [22]:

In [51]:

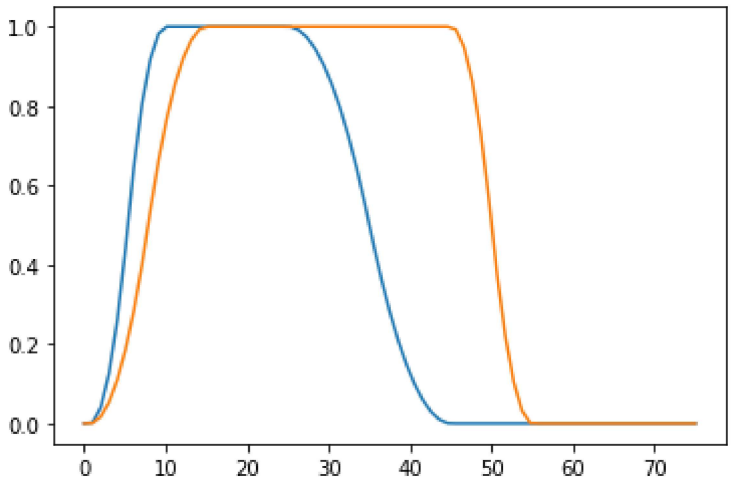
```
#diff = fuz.membership.dsigmf(x,b1,c1,b2,c2)
#prod = fuz.membership.psigmf(x,b1,c1,b2,c2)
#bell = fuz.membership.gbellmf(x,a,b,c)

a=0.7
b=10
c=25
d=45

a1=0.7
b1=15
c1=45
d1=55
a_pie =   fuz.membership.pimf(x,a,b,c,d)
b_pie =   fuz.membership.pimf(x,a1,b1,c1,d1)

union= fuz.fuzzy_or(x,a_pie,x,b_pie)
intersection = fuz.fuzzy_and(x,a_pie,x,b_pie)
complement = fuz.fuzzy_not(a_pie)
alg_prod = a_pie * b_pie

#basic =fuz.membership.sigmf(x,b,c)
#s = fuz.membership.smf(x,a,b)
#z = fuz.membership.zmf(x,a,b)
```

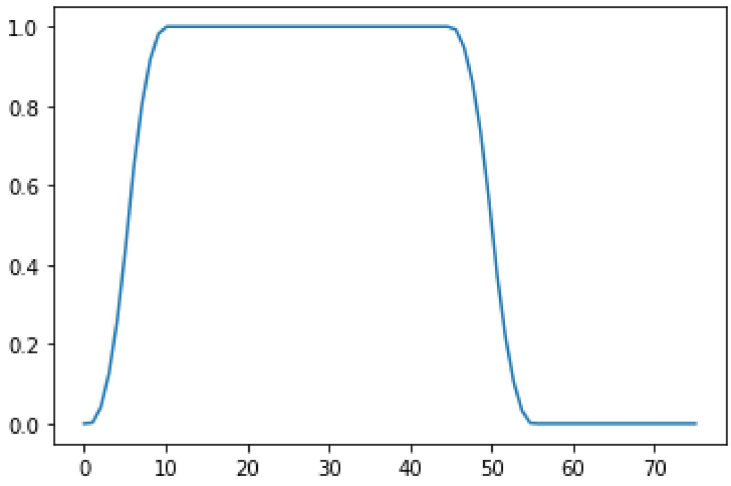
In [52]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,union[1])
```

Out[52]:

```
[<matplotlib.lines.Line2D at 0x4a45c1e4f0>]
```



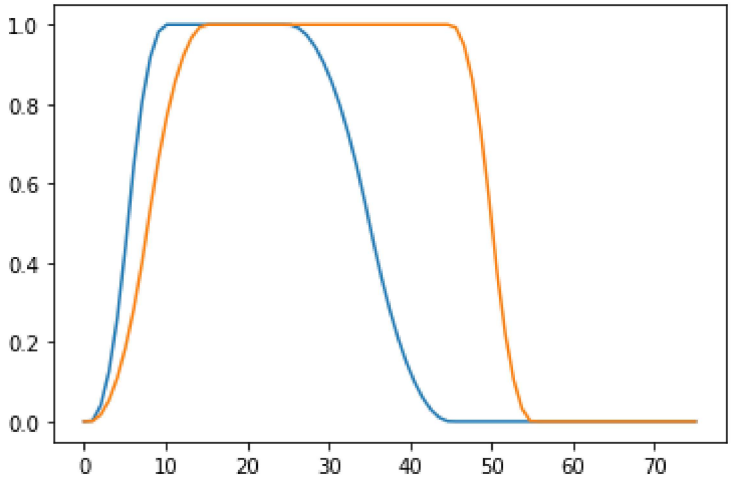


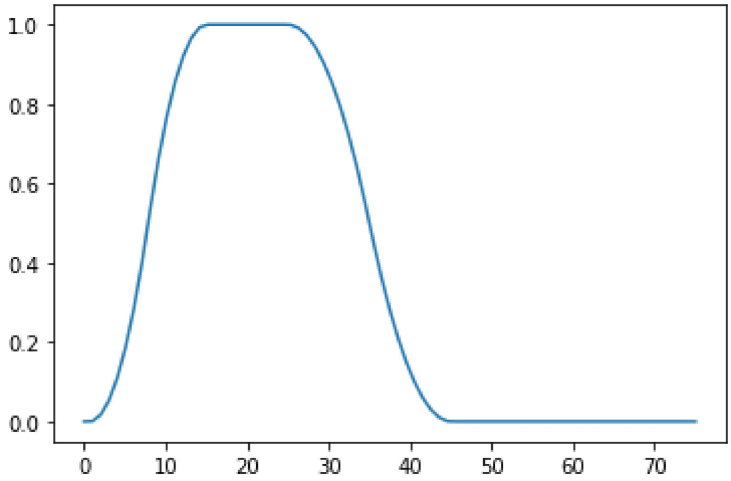
In [54]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,intersection[1])
```

Out[54]:

```
[<matplotlib.lines.Line2D at 0x4a45d7e430>]
```



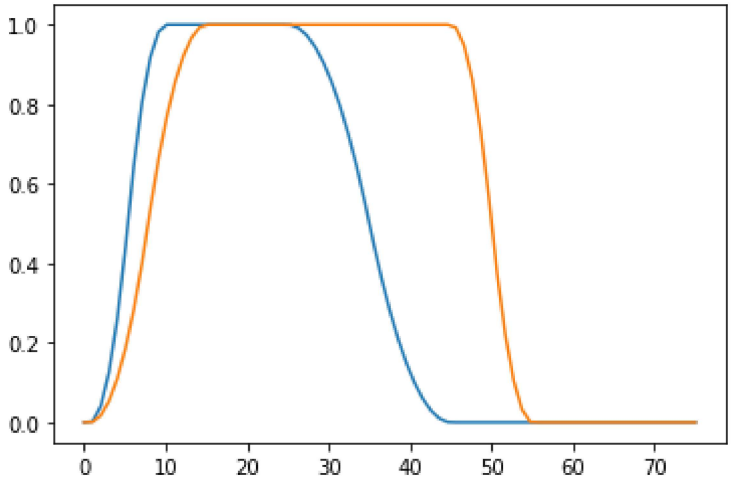


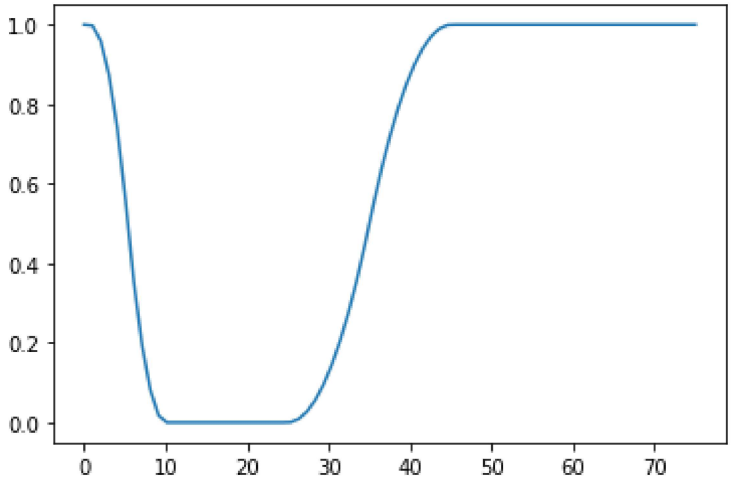
In [56]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,complement)
```

Out[56]:

```
[<matplotlib.lines.Line2D at 0x4a46ef7a00>]
```



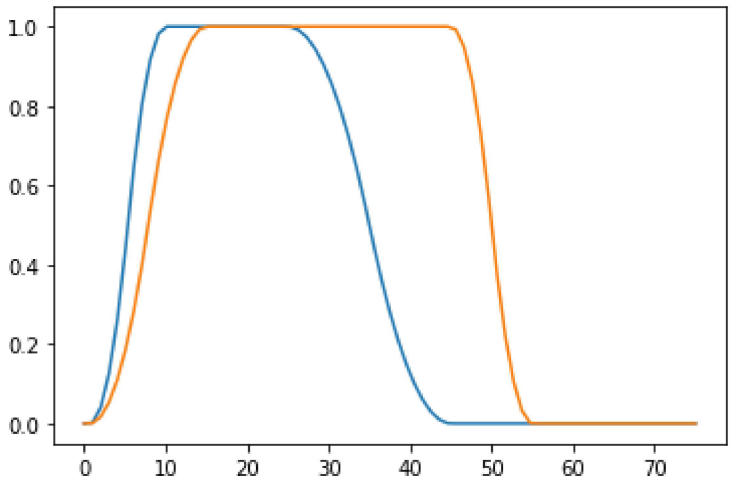


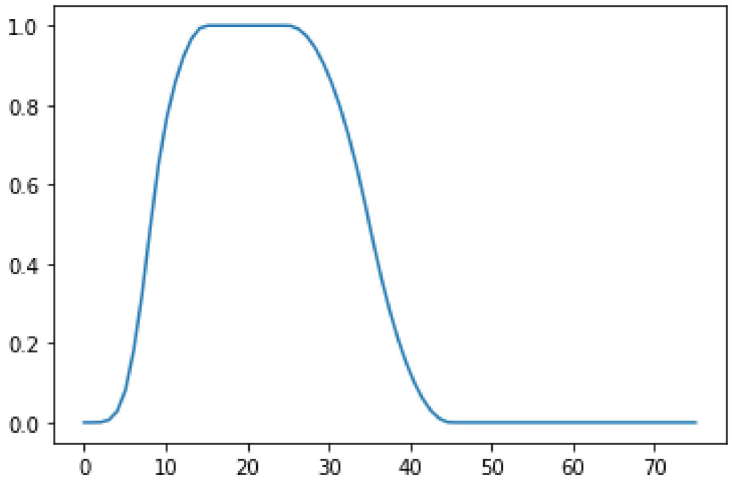
In [57]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,alg_prod)
```

Out[57]:

```
[<matplotlib.lines.Line2D at 0x4a46fcde20>]
```





Gaussian member function

In [64]:

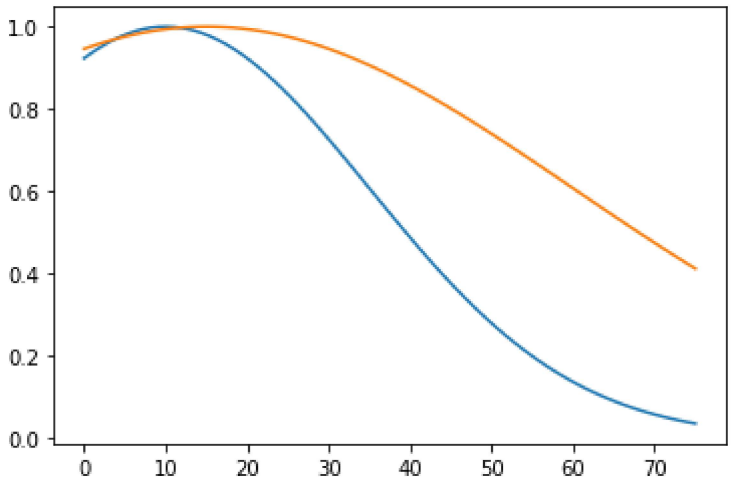
```
b=10
c=25

b1=15
c1=45

a_pie = fuz.membership.gaussmf(x, b, c)

b_pie = fuz.membership.gaussmf(x, b1, c1)

union= fuz.fuzzy_or(x,a_pie,x,b_pie)
intersection = fuz.fuzzy_and(x,a_pie,x,b_pie)
complement = fuz.fuzzy_not(a_pie)
alg_prod = a_pie * b_pie
```

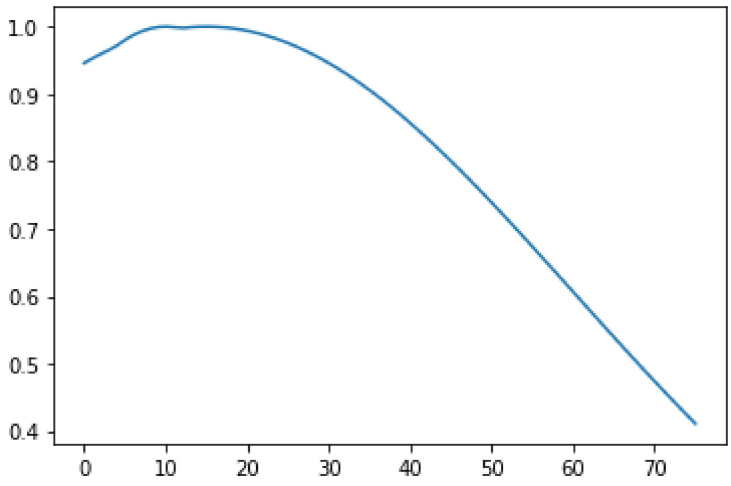
In [65]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,union[1])
```

Out[65]:

```
[<matplotlib.lines.Line2D at 0x4a46f0b130>]
```



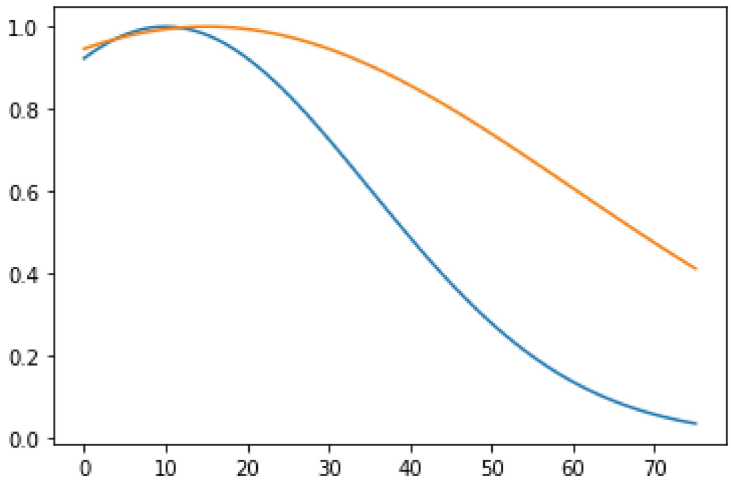


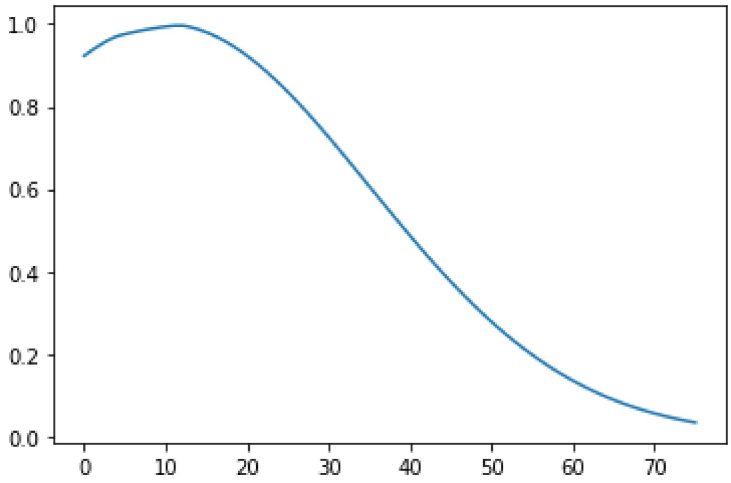
In [66]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,intersection[1])
```

Out[66]:

```
[<matplotlib.lines.Line2D at 0x4a45c6c280>]
```



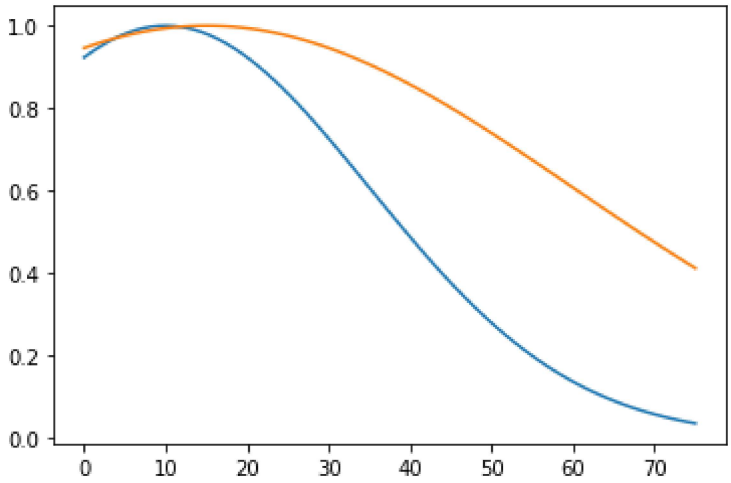


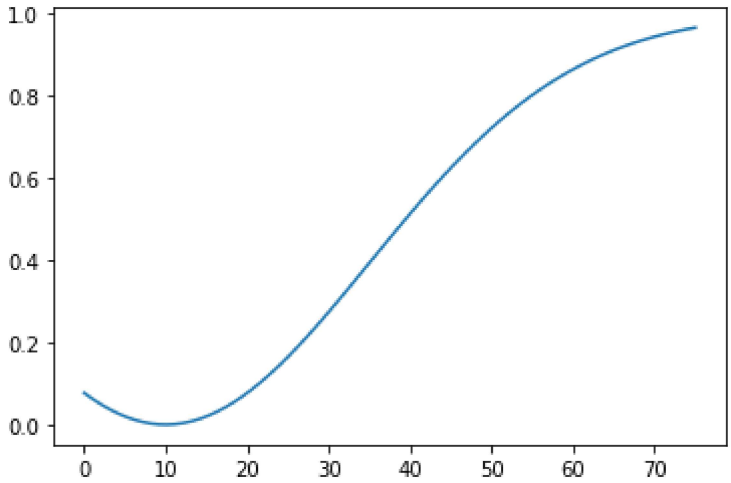
In [67]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,complement)
```

Out[67]:

```
[<matplotlib.lines.Line2D at 0x4a4572db20>]
```



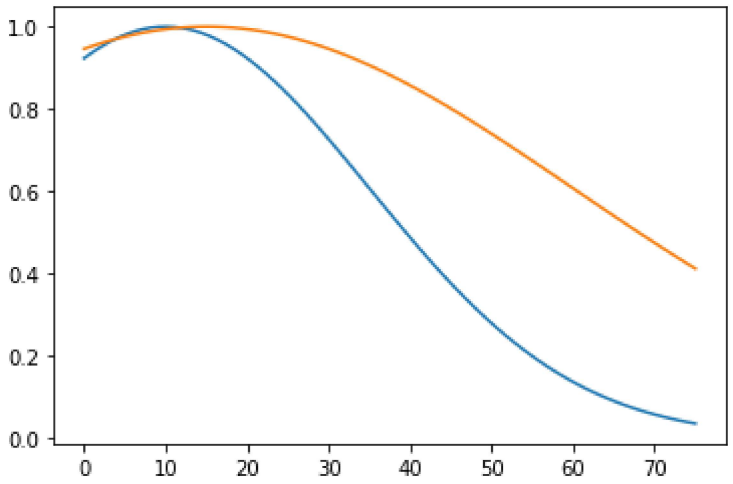


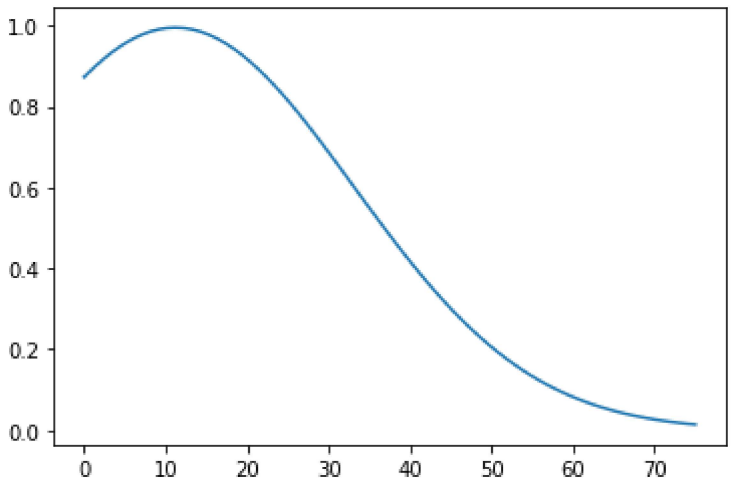
In [68]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,alg_prod)
```

Out[68]:

```
[<matplotlib.lines.Line2D at 0x4a45c5b700>]
```





Generalised bell

In [70]:

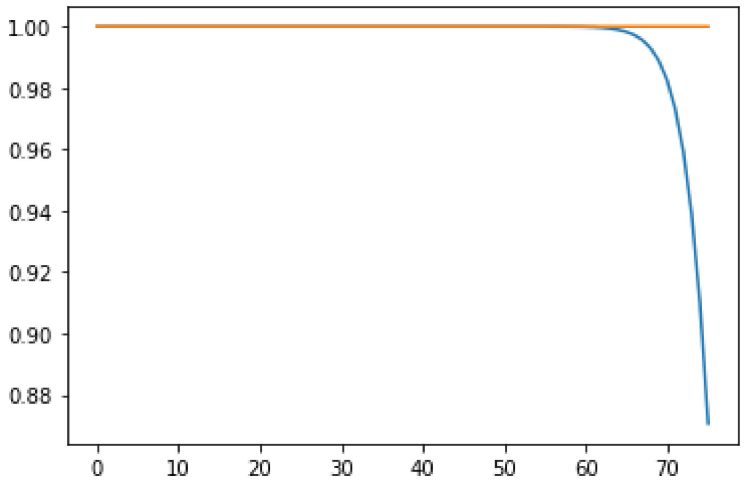
```
a=55
b=10
c=25

a1=65
b1=15
c1=45

a_pie = fuz.membership.gbellmf(x, a, b, c)

b_pie = fuz.membership.gbellmf(x, a1, b1, c1)

union= fuz.fuzzy_or(x,a_pie,x,b_pie)
intersection = fuz.fuzzy_and(x,a_pie,x,b_pie)
complement = fuz.fuzzy_not(a_pie)
alg_prod = a_pie * b_pie
```

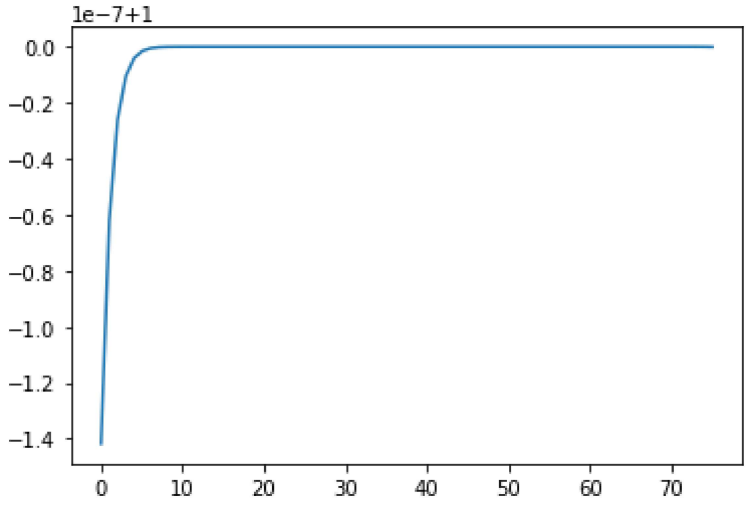
In [71]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,union[1])
```

Out[71]:

```
[<matplotlib.lines.Line2D at 0x4a47106940>]
```



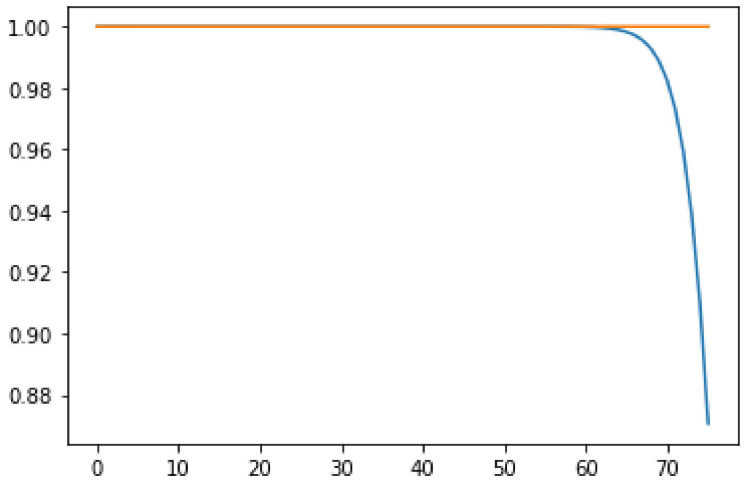


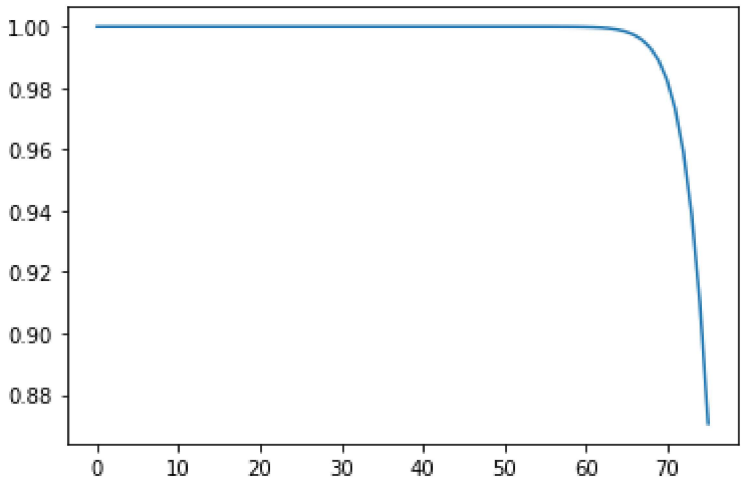
In [72]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,intersection[1])
```

Out[72]:

```
[<matplotlib.lines.Line2D at 0x4a471ef250>]
```



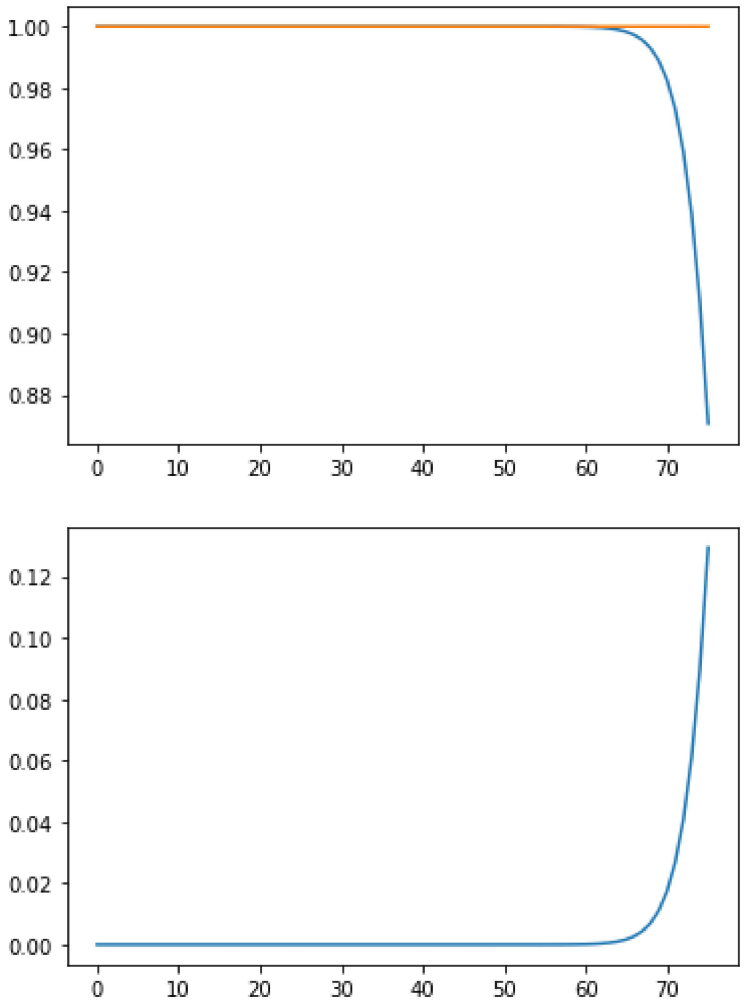


In [73]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,complement)
```

Out[73]:

```
[<matplotlib.lines.Line2D at 0x4a472ce1f0>]
```



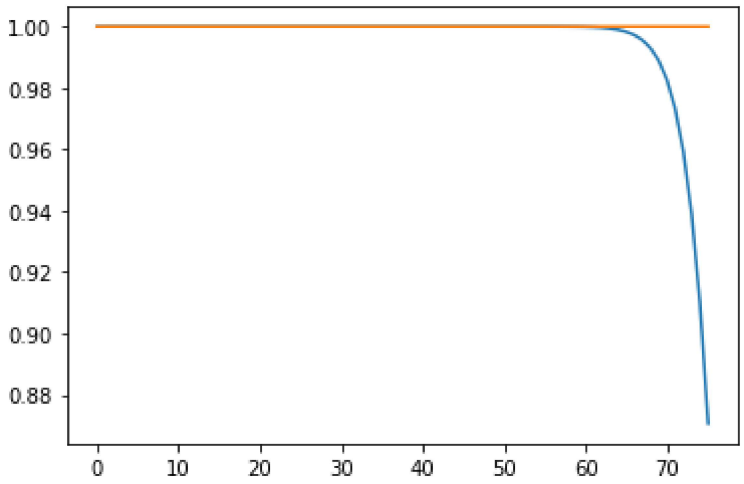
In [ ]:

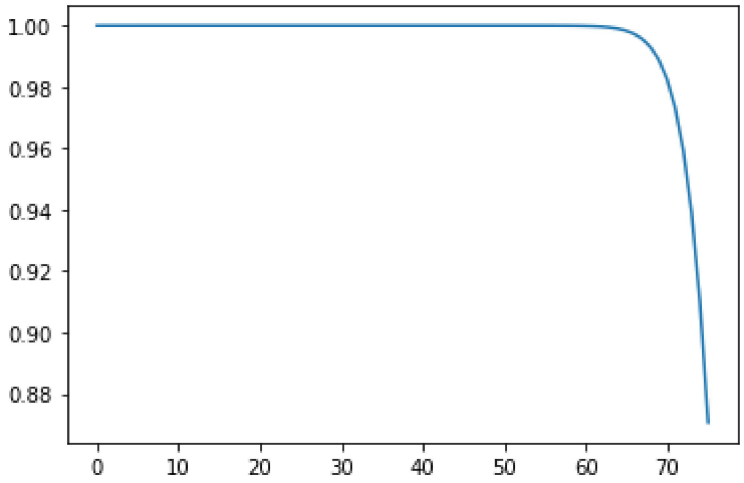
In [74]:

```
plt.plot(x,a_pie)
plt.plot(x,b_pie)
plt.figure()
plt.plot(x,alg_prod)
```

Out[74]:

```
[<matplotlib.lines.Line2D at 0x4a473a9220>]
```





In [ ]:

```
```

In [ ]:

```
```