In [5]:
```
import names
import pandas as pd
from math import sqrt
```

In [4]:
```
pip install names
```

```
Collecting namesNote: you may need to restart the kernel to use updated packages.

  Using cached names-0.3.0.tar.gz (789 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: names
  Building wheel for names (setup.py): started
  Building wheel for names (setup.py): finished with status 'done'
  Created wheel for names: filename=names-0.3.0-py3-none-any.whl size=803682 sha256=e
ee1e0b9dbec58f171e1366ddadefef670bc36708c2c749cfef976537956d3e3
  Stored in directory: c:\users\kiran\appdata\local\pip\cache\wheels\f1\bc\04\55ab949
9ea02359ece8b02b4169ebb30aa52d82b84c13fc506
Successfully built names
Installing collected packages: names
Successfully installed names-0.3.0
```

In [89]:
```
class Euclidean:
    def __init__(self,a,b,c,d,e,f,g,h,i):
        self.a=a
        self.b=b
        self.c=c
        self.d=d
        self.e=e
        self.f=f
        self.g=g
        self.h=h
        self.i=i

    def clusters(self):
        self.cluster1=[self.a,self.b,self.c]
        self.cluster2=[self.d,self.e,self.f]
        self.cluster3=[self.g,self.h,self.i]
        return(f"A{self.a}, Cluster1: {self.cluster1}",f"Cluster2:{self.cluster2}",f"C
    def cluster_a(self):
        self.dis_a_x=1.0 #distance from A to x
        self.dis_a_y=0.9 #distance from A to u
        dis_b_x=1.0 #distance from b to x
        dis_b_y=1.7 #distance from c to x
        dis_c_x=1.3 #distance from c to x
        dis_c_y=1.5  #distance from c to y
        index=self.cluster1
        data={'x':[self.dis_a_x,dis_b_x,dis_c_x],'y':[self.dis_a_y,dis_b_y,dis_c_y]}
        self.df1=pd.DataFrame(data=data,index=index)
        euclidean_a_b=round(sqrt((dis_b_x-self.dis_a_x)**2+(dis_b_y-self.dis_a_y)**2),
        euclidean_a_c=round(sqrt((dis_c_x-self.dis_a_x)**2+(dis_c_y-self.dis_a_y)**2),
        self.alpha=round((euclidean_a_b+euclidean_a_c)/2,2)
        print(f"Euclidean Distance {self.a} to {self.b}: {euclidean_a_b}")
        print(f"Euclidean Distance {self.a} to {self.c}: {euclidean_a_c}")
        print(f"Shilhouette Alpha:{self.alpha}")
        return self.df1

    def cluster_b(self):
```

```
        dis_d_x=1.3 #distance from d to x
        dis_d_y=3.9 #distance from d to y
        dis_e_x=1.5 #distance from e to x
        dis_e_y=4.3  #distance from e to y
        dis_f_x=1.6  #distance from f to x
        dis_f_y=3.7  #distance from f to y
        index=self.cluster2
        data={'x':[dis_d_x,dis_e_x,dis_f_x],'y':[dis_d_y,dis_e_y,dis_f_y]}
        self.df2=pd.DataFrame(data=data,index=index)
        euclidean_a_d=round(sqrt((dis_d_x-self.dis_a_x)**2+(dis_d_y-self.dis_a_y)**2),
        euclidean_a_e=round(sqrt((dis_e_x-self.dis_a_x)**2+(dis_e_y-self.dis_a_y)**2),
        euclidean_a_f=round(sqrt((dis_f_x-self.dis_a_x)**2+(dis_f_y-self.dis_a_y)**2),
        self.avg_dis_a_c2=round((euclidean_a_d+euclidean_a_e+euclidean_a_f)/3,2)
        print(f"Euclidean Distance {self.a} to {self.d}: {euclidean_a_d}")
        print(f"Euclidean Distance {self.a} to {self.e}: {euclidean_a_e}")
        print(f"Euclidean Distance {self.a} to {self.f}: {euclidean_a_f}")
        print(f"Average Distance:{self.a} to Cluster 2:{self.avg_dis_a_c2}")
        print('\n')
        print(self.df1.iloc[0].to_frame())
        return self.df2

    def cluster_c(self):
        dis_g_x=2.1 #distance from g to x
        dis_g_y=0.8 #distance from g to y
        dis_h_x=2.2 #distance from h to x
        dis_h_y=1.5  #distance from h to y
        dis_i_x=2.5  #distance from i to x
        dis_i_y=0.9  #distance from i to y
        index=self.cluster3
        data={'x':[dis_g_x,dis_h_x,dis_i_x],'y':[dis_g_y,dis_h_y,dis_i_y]}
        self.df3=pd.DataFrame(data=data,index=index)
        euclidean_a_g=round(sqrt((dis_g_x-self.dis_a_x)**2+(dis_g_y-self.dis_a_y)**2),
        euclidean_a_h=round(sqrt((dis_h_x-self.dis_a_x)**2+(dis_h_y-self.dis_a_y)**2),
        euclidean_a_i=round(sqrt((dis_i_x-self.dis_a_x)**2+(dis_i_y-self.dis_a_y)**2),
        self.avg_dis_a_c3=round((euclidean_a_g+euclidean_a_h+euclidean_a_i)/3,2)
        print(f"Euclidean Distance {self.a} to {self.g}: {euclidean_a_g}")
        print(f"Euclidean Distance {self.a} to {self.h}: {euclidean_a_h}")
        print(f"Euclidean Distance {self.a} to {self.i}: {euclidean_a_i}")
        print(f"Average Distance:{self.a} to Cluster 3:{self.avg_dis_a_c3}")
        print('\n')
        print(self.df1.iloc[0].to_frame())
        return self.df3

    def silhouette(self):
        beta=min([self.avg_dis_a_c2,self.avg_dis_a_c3])
        max_alpha_beta=max([beta,self.alpha])
        silhouette=(beta-self.alpha)/max_alpha_beta
        print(f"Alpha :{self.alpha}")
        print(f"Beta:  {beta}")
        return f"Silhouette Score of {self.a}:{silhouette}"
```

In [90]:
```
test=Euclidean(*[names.get_first_name() for i in range(0,9)])
```

In [91]:
```
test.clusters()
```

Out[91]:
```
("ACarl, Cluster1: ['Carl', 'Brenda', 'Timothy']",
 "Cluster2:['Ernesto', 'Mark', 'Donna']",
 "Cluster3:['Bertha', 'Richard', 'Jerry']")
```

In [92]:
```
test.cluster_a()
```

```
Euclidean Distance Carl to Brenda: 0.8
Euclidean Distance Carl to Timothy: 0.67
Shilhouette Alpha:0.74
```

Out[92]:

	x	y
Carl	1.0	0.9
Brenda	1.0	1.7
Timothy	1.3	1.5

In [93]:
```
test.cluster_b()
```

```
Euclidean Distance Carl to Ernesto: 3.01
Euclidean Distance Carl to Mark: 3.44
Euclidean Distance Carl to Donna: 2.86
Average Distance:Carl to Cluster 2:3.1


   Carl
x   1.0
y   0.9
```

Out[93]:

	x	y
Ernesto	1.3	3.9
Mark	1.5	4.3
Donna	1.6	3.7

In [94]:
```
test.cluster_c()
```

```
Euclidean Distance Carl to Bertha: 1.1
Euclidean Distance Carl to Richard: 1.34
Euclidean Distance Carl to Jerry: 1.5
Average Distance:Carl to Cluster 3:1.31


   Carl
x   1.0
y   0.9
```

Out[94]:

	x	y
Bertha	2.1	0.8
Richard	2.2	1.5
Jerry	2.5	0.9

In [95]:
```
test.silhouette()
```

```
Alpha :0.74
Beta:  1.31
```

Out[95]: 'Silhouette Score of Carl:0.4351145038167939'

In [ ]:

In [ ]: