In [ ]:

```
a=-1

#  [a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11]
C=[]

def again():

    return a*a-b1*b1-b2*b2-b3*b3-b4*b4-b5*b5-b6*b6-b7*b7-b8*b8-b9*b9-b10*b10-b11*b11<=0 and -3*a+b1+b2+b3+b4+b5+b6+b7+b8+b
#if a*a-b1*b1-b2*b2-b3*b3-b4*b4<=0 and -3*a+b1+b2+b3+b4+a*a-b1*b1-b2*b2-b3*b3-b4*b4 == -2:
#    C.append([a,b1,b2,b3,b4,b5,b6,b7,b8])
#    break

    #if  a*a-b1*b1-b2*b2-b3*b3-b4*b4-b5*b5-b6*b6-b7*b7-b8*b8-b9*b9-b10*b10-b11*b11 ==0 and -3*a+b1+b2+b3+b4+b5+b6+b7+b8+b9
        #C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
        #break
while a<=9:
    a=a+1
    b1=-1
    b2=-1
    b3=-1
    b4=-1
    b5=-1
    b6=-1
    b7=-1
    b8=-1
    b9=-1
    b10=-1
    b11=-1
if again() == True:
        C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
        break
    while b2<=b1:
        if again() == True:
            C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
            break
        while b3<=b2:
            if again() == True:
                C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                break
            while b4<=b3:
                #b4=b4+1
                if again() == True:
                    C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                    break
                while b5<=b4:
                    if again() == True:
                        C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                        break
                    while b6<=b5:
                        if again() == True:
                            C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                            break
                        while b7<=b6:
                            if again() == True:
                                C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                                break
                            while b8<=b7:
                                if again() == True:
                                    C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                                    break
                                while b9<=b8:
                                    if again() == True:
                                        C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                                        break
                                    while b10<=b9:
                                        if again() == True:
                                            C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                                            break
                                        while b11<=b10:
                                            b11=b11+1
                                            if again() == True:
                                                C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                                            #break
                                        b10=b10+1
                                        if again() == True:
                                            C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                                        # break
                                    b9=b9+1
                                    if again() == True:
                                        C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                                #  break
                                b8=b8+1
```

```
                            if again() == True:
                                C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                        #break
                        b7=b7+1
                        if again() == True:
                            C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                                # break
                    b6=b6+1
                    if again() == True:
                        C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                            #  break
                b5=b5+1
                if again() == True:
                    C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                        #break
            b4=b4+1
            if again() == True:
                C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
                        #break
        b3=b3+1
        if again() == True:
            C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
            # break
    b2=b2+1
    if again() == True:
        C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
        #  break
b1=b1+1
if again() == True:
    C.append([a,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11])
    #  break

print(C)
```

In [47]:

```
C=[[0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  [2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0],
 [3, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0],  [4, 2, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0],  [5, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0],
 [6, 3, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0],  [7, 4, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1], [8,4,3,3,3,2,2,2,2,2,1,1],  [8,5,2,2,2,2,2,2
```

In [52]:

```
# Output the reduced exceptional classes. We only need to check those have positive area.
print(C)
```

```
[[0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0], [3,
2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0], [4, 2, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0], [5, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0], [6, 3, 2,
2, 2, 2, 2, 2, 2, 0, 0, 0], [7, 4, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1], [8, 4, 3, 3, 3, 2, 2, 2, 2, 2, 1, 1], [8, 5, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2], [9, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 0]]
```

In [53]:

```
A=[1-0.0007, -(1/2 - 0.0007), -(2/7 + 0.0001), -(2/7 + 0.0001), -(2/7 + 0.0001), -(2/7 + 0.0001), -(2/7 + 0.0001), -(2/7 +
```

In [54]:

```
import numpy as np
```

In [56]:

```
for i in range (0, 11):
    a=np.dot(A, C[i])
    print(a)
```

```
0.4993
0.49999999999999994
0.3560428571428571
0.2844142857142855
0.4262714285714284
0.5681285714285722
0.49650000000000016
0.13895714285714256
0.28081428571428635
0.35304285714285655
0.35044285714285817
```

In [ ]: