Find the output of the following program. If there is any error, correct the error. But if there is no error, you don’t need to change anything in the program

```
import numpy as np
import random

aa = [1,2,3]
bb = [1, 2, 3, 4, 5, 6]

bb = np.array(bb)
aa = aa * 2
print(aa)

mult = aa * bb
print(bb)

num = random.randrange(5, 10, 10)

mult = num * mult
np.reshape(bb, (2,1,3))
print(bb)
print(mult)

aa.clear()

if aa:
    bb = bb * 2
    print(bb)

if bb.any():
    bb = bb * 2
    print(bb)
```

Write a program that will generate a matrix of size 1000 X 20 with random values from 0 to 9. Each row of the matrix indicates a trial. For each of the trial, you need to find the sum of the same digits. You also need to find the largest sum for each trial. Part of the program is done for you. You need to complete the rest of the part.

```
import numpy as np
import random

def findBiggestSum(sumOfSameDigitsVec):
    """ This function takes some of Same digits as input
    and it returns the tuple that has largest sum"""

def findSumOfSameDigits(numVec):
    """ This function takes a number of vector as parameter
    and it finds the sum of th same digits. For example:
    if the vector is [5, 5, 2, 3], it finds the sum of 5 is
    10, sum of 2 is 2 and sum of 3 is 3. It returns a list
    where each element in the list is a tuple with the
digit
    and its corresponding sum."""



def main():
    numbersMat = np.random.randint(1,9, (1000, 20))

    for numVec in numbersMat:
        sumOfSameDigits = findSumOfSameDigits(numVec)
        for item in sumOfSameDigits:
            print('Total Sum of {:d}: {:d}'.format(item[0],
item[1]))
        biggestSum = findBiggestSum(sumOfSameDigits)
        print('The largest sum with digit {:d} is
{:d}'.format(biggestSum[0], biggestSum[0]))

        print()
        print()
```

```
if __name__ == '__main__':
    main()
```