M.Akshara

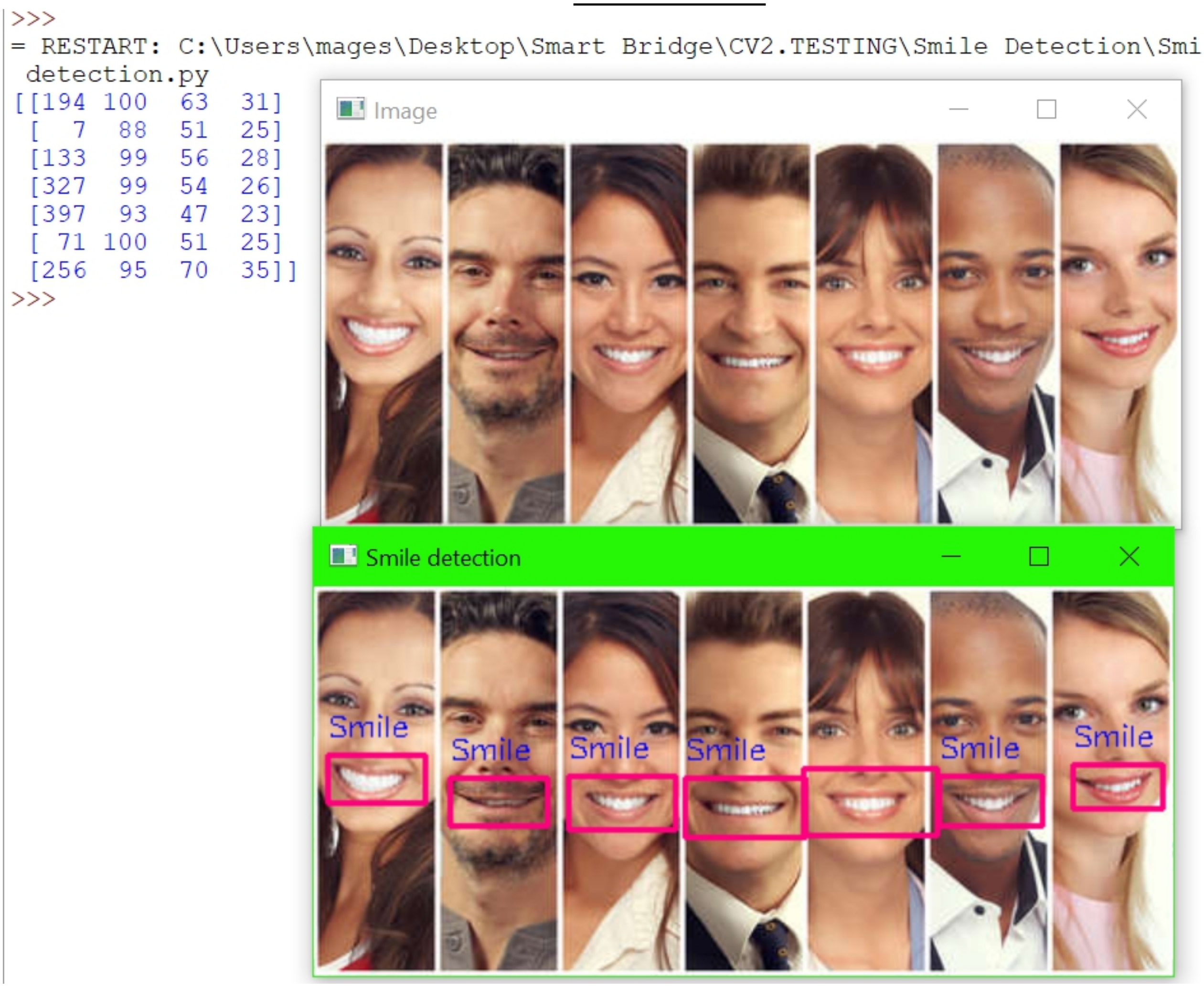
ASSIGNMENT 6

Develop a python code to detect any object using Haar cascade classifier.

PYTHON CODE(IMAGE)

```
import cv2
smile_classifier=cv2.CascadeClassifier("haarcascade_smile.xml")
image=cv2.imread('Smile.jpg')
cv2.imshow('Image',image)
smiles = smile_classifier.detectMultiScale(image,1.3,5)
print(smiles)
for(x,y,w,h) in smiles:
   cv2.rectangle(image, (x,y), (x+w,y+h), (127,0,255), 2)
   cv2.imshow('Smile detection', image)
   cv2.putText(image, 'Smile',(x,y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,0,0), 1)
   Key=cv2.waitKey(1)
   if Key==ord('q'):
      #release the camera
      video.release()
      #destroy all windows
      cv2.destroyAllWindows()
      break
```

SCREENSHOT

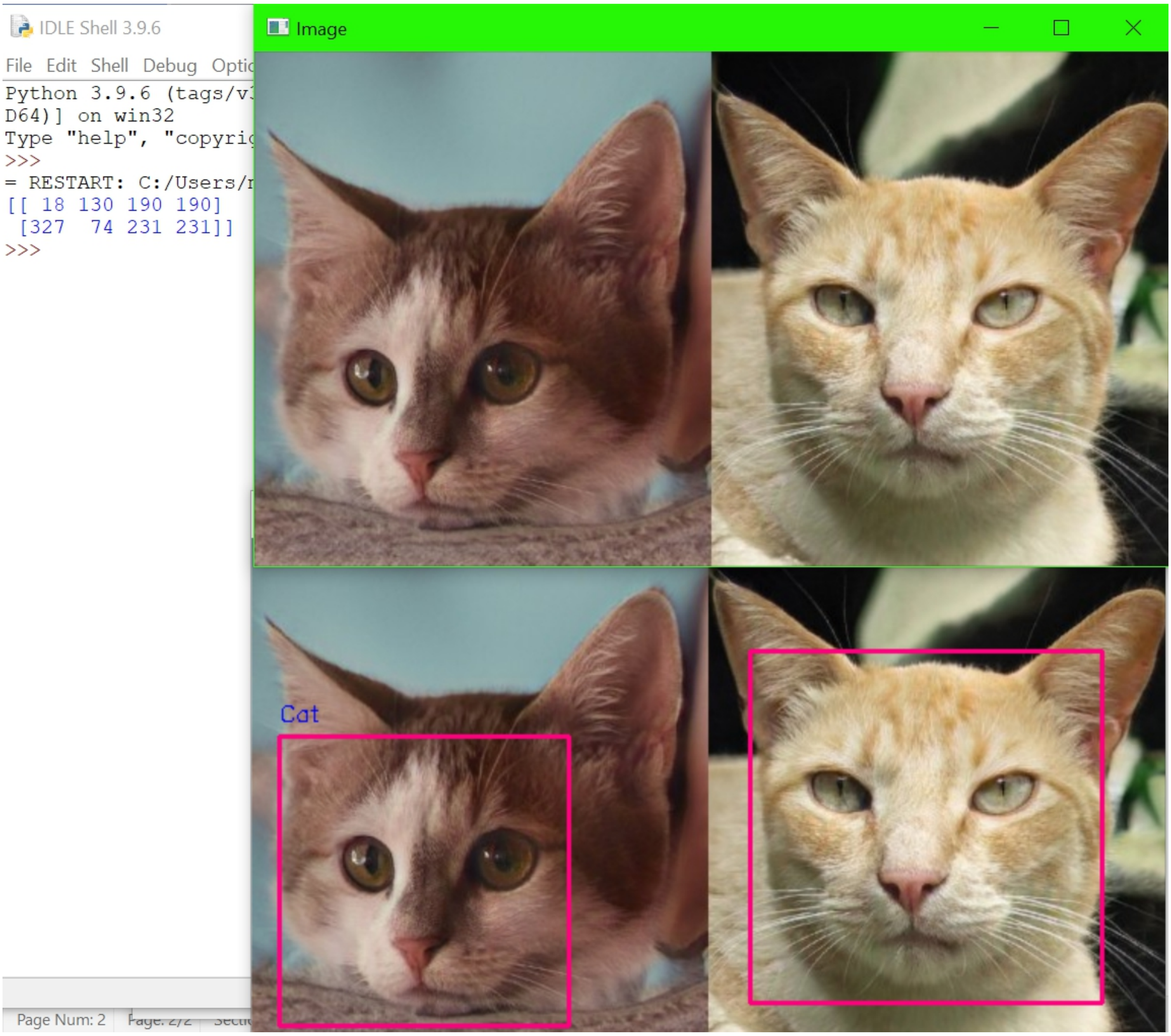


```
>>>
= RESTART: C:\Users\mages\Desktop\Smart Bridge\CV2.TESTING\Smile Detection\Smi
 detection.py
[[194 100  63  31]
 [  7  88  51  25]
 [133  99  56  28]
 [327  99  54  26]
 [397  93  47  23]
 [ 71 100  51  25]
 [256  95  70  35]]
>>>
```

PYTHON CODE(IMAGE)

```
import cv2
cat_classifier=cv2.CascadeClassifier("haarcascade_frontalcatface_extended.xml")
image=cv2.imread('Cats.jpg')
cv2.imshow('Image',image)
cats = cat_classifier.detectMultiScale(image,1.3,5)
print(cats)
for(x,y,w,h) in cats:
   cv2.rectangle(image, (x,y), (x+w,y+h), (127,0,255), 2)
   cv2.imshow('Cat detection', image)
   cv2.putText(image, 'Cat',(x,y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,0,0), 1)
   Key=cv2.waitKey(1)
   if Key==ord('q'):
      #release the camera
      video.release()
      #destroy all windows
      cv2.destroyAllWindows()
      break
```

SCREENSHOT

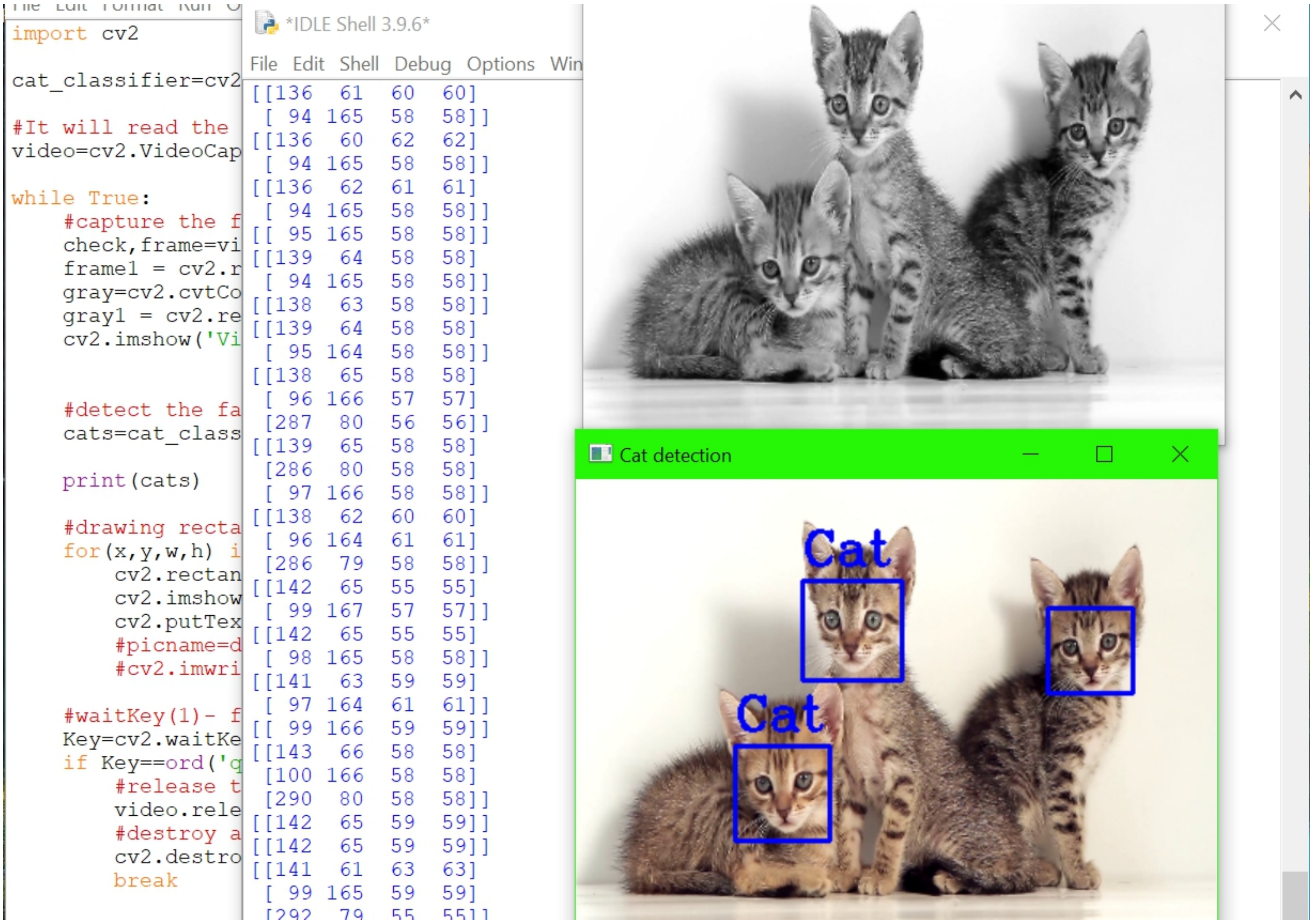


```
Python 3.9.6 (tags/v
D64)] on win32
Type "help", "copyric
>>>
= RESTART: C:/Users/r
[[ 18 130 190 190]
 [327  74 231 231]]
>>>
```

PYTHON CODE(VIDEO)

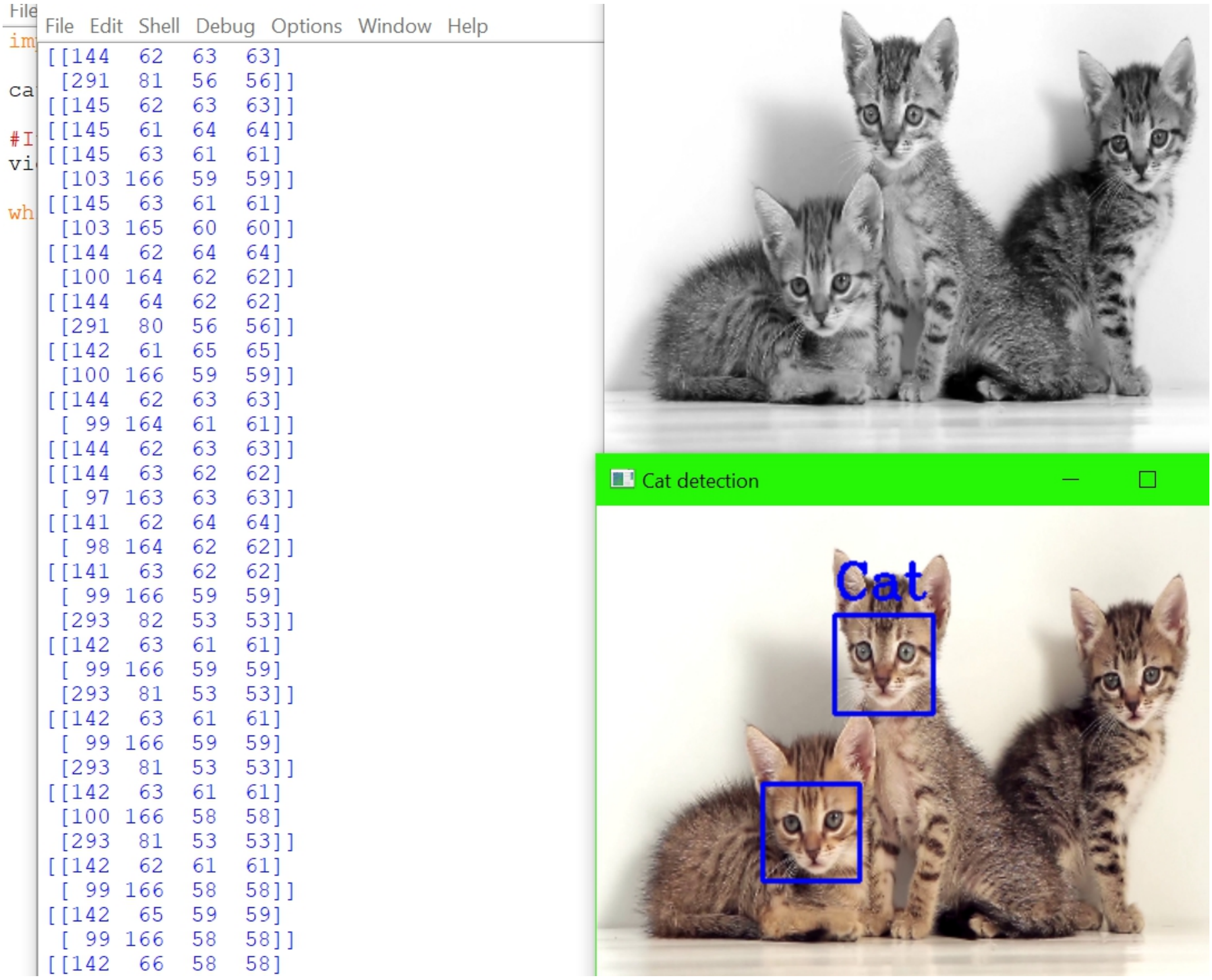
```
import cv2
cat_classifier=cv2.CascadeClassifier("haarcascade_frontalcatface_extended.xml")
#It will read the first frame/image of the video
video=cv2.VideoCapture('Cats4.mp4')
while True:
   #capture the first frame
   check,frame=video.read()
   frame1 = cv2.resize(frame, (400, 300))
   gray=cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
   gray1 = cv2.resize(gray, (400, 300))
   cv2.imshow('Video',gray1)
   #detect the faces from the video using detectMultiScale function
   cats=cat_classifier.detectMultiScale(gray1,1.3,5)
   print(cats)
   #drawing rectangle boundries for the detected face
   for(x,y,w,h) in cats:
      cv2.rectangle(frame1, (x,y), (x+w,y+h), (255,0,0), 2)
      cv2.imshow('Cat detection', frame1)
      cv2.putText(frame1, 'Cat',(x,y-10), cv2.FONT_HERSHEY_COMPLEX, 1, (255,0,0), 2)
      #picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
      #cv2.imwrite(picname+".jpg",frame)
   #waitKey(1)- for every 1 millisecond new frame will be captured
   Key=cv2.waitKey(25)
   if Key==ord('q'):
      #release the camera
      video.release()
      #destroy all windows
      cv2.destroyAllWindows()
      break
```

SCREENSHOT



```
import cv2

cat_classifier=cv2

#It will read the
video=cv2.VideoCap

while True:
    #capture the f
    check,frame=vi
    frame1 = cv2.r
    gray=cv2.cvtCo
    gray1 = cv2.re
    cv2.imshow('Vi


    #detect the fa
    cats=cat_class

    print(cats)

    #drawing recta
    for(x,y,w,h) i
        cv2.rectan
        cv2.imshow
        cv2.putTex
        #picname=d
        #cv2.imwri

    #waitKey(1)- f
    Key=cv2.waitKe
    if Key==ord('q
        #release t
        video.rele
        #destroy a
        cv2.destro
        break
```

```
[[136  61  60  60]
 [ 94 165  58  58]]
[[136  60  62  62]
 [ 94 165  58  58]]
[[136  62  61  61]
 [ 94 165  58  58]]
[[ 95 165  58  58]]
[[139  64  58  58]
 [ 94 165  58  58]]
[[138  63  58  58]]
[[139  64  58  58]
 [ 95 164  58  58]]
[[138  65  58  58]
 [ 96 166  57  57]
 [287  80  56  56]]
[[139  65  58  58]
 [286  80  58  58]
 [ 97 166  58  58]]
[[138  62  60  60]
 [ 96 164  61  61]
 [286  79  58  58]]
[[142  65  55  55]
 [ 99 167  57  57]]
[[142  65  55  55]
 [ 98 165  58  58]]
[[141  63  59  59]
 [ 97 164  61  61]]
[[ 99 166  59  59]]
[[143  66  58  58]
 [100 166  58  58]
 [290  80  58  58]]
[[142  65  59  59]]
[[142  65  59  59]]
[[141  61  63  63]
 [ 99 165  59  59]
 [292  79  55  55]]
```



```
[[144  62  63  63]
 [291  81  56  56]]
[[145  62  63  63]]
[[145  61  64  64]]
[[145  63  61  61]
 [103 166  59  59]]
[[145  63  61  61]
 [103 165  60  60]]
[[144  62  64  64]
 [100 164  62  62]]
[[144  64  62  62]
 [291  80  56  56]]
[[142  61  65  65]
 [100 166  59  59]]
[[144  62  63  63]
 [ 99 164  61  61]]
[[144  62  63  63]]
[[144  63  62  62]
 [ 97 163  63  63]]
[[141  62  64  64]
 [ 98 164  62  62]]
[[141  63  62  62]
 [ 99 166  59  59]
 [293  82  53  53]]
[[142  63  61  61]
 [ 99 166  59  59]
 [293  81  53  53]]
[[142  63  61  61]
 [ 99 166  59  59]
 [293  81  53  53]]
[[142  63  61  61]
 [100 166  58  58]
 [293  81  53  53]]
[[142  62  61  61]
 [ 99 166  58  58]]
[[142  65  59  59]
 [ 99 166  58  58]]
[[142  66  58  58]
```