```python
import cv2
import numpy as np
#TASK 5 omar Rashad salem

#import reference
ref_10p = cv2.imread("ref_10p.jpg")
ref_5p = cv2.imread("ref_5p.jpg")

#import test samples
dhs10_tst = cv2.imread("10dhs_tst.jpg")
folded5p_tst = cv2.imread("5p_folded_tst.jpg")
old5p_tst = cv2.imread("old5p_tst.jpg")
many10p_tst = cv2.imread("10p_tst2.jpg")
good10p_tst = cv2.imread("10p_tst3.jpg")

sift_obj = cv2.SIFT_create()

#do SIFT for the reference images
kp_10p_ref , desc_10p_ref = sift_obj.detectAndCompute(ref_10p , None)
kp_5p_ref , desc_5p_ref = sift_obj.detectAndCompute(ref_5p , None)

#do sift for the test samples
kp_fold5p , desc_fold5p = sift_obj.detectAndCompute(folded5p_tst , None)
kp_old5p , desc_old5p = sift_obj.detectAndCompute(old5p_tst , None)
kp_many10p , desc_many10p = sift_obj.detectAndCompute(many10p_tst , None)
kp_good10p , desc_good10p = sift_obj.detectAndCompute(good10p_tst , None)

kp_tst_ls = [kp_fold5p , kp_old5p , kp_many10p , kp_good10p] #or use dict
desc_tst_ls = [ desc_fold5p , desc_old5p , desc_many10p , desc_good10p  ,
desc_5p_ref ,desc_10p_ref]

#create match objects
mathcer = cv2.BFMatcher(cv2.NORM_L2)
# matched_with_5p = mathcer.match( desc_5p_ref , desc_5p_ref )
# matched_with_10p = mathcer.match( desc_10p_ref , desc_10p_ref )

mathcer2 = cv2.BFMatcher(cv2.NORM_L2)

#TESTING WITH DIFF SAMPLES
for i in range(len(desc_tst_ls)):

    #get matches with the ref 5p
    matched_with_5p = mathcer.match( desc_5p_ref , desc_tst_ls[i] )
    matched_with_5p = sorted( matched_with_5p , key = lambda x : x.queryIdx )
```

```python
    #get the matches with the ref 10p
    matched_with_10p = mathcer.match( desc_10p_ref , desc_tst_ls[i] )
    matched_with_10p = sorted( matched_with_10p , key = lambda x : x.queryIdx )

    #smaller sum is the supposed righ match
    sum_10p , sum_5p = 0 , 0
    for j in range(len(matched_with_5p)): #take same number of features from both
        if matched_with_10p[j].distance < matched_with_5p[j].distance :
            sum_5p += 1
        elif matched_with_10p[j].distance > matched_with_5p[j].distance :
            sum_10p += 1
        else :
            sum_10p += 1
            sum_5p += 1

    # compare to find the supposed right  match
    if sum_10p > sum_5p :
        print (f"test {i} : is a 5 pounds bill")
    elif sum_10p < sum_5p :
        print(f"test {i} : is a 10 pounds bill")
    else:
        print(f"test {i} couldn't decide")

        #TODO : add thresh hold later do detect it neither of them

# cv2.waitKey(0)
#END TASK 5 omar rashad salem
```

O\P + used images

```
E:\omar-work\tasks\multimedia\sheet6
λ py sheet6_task5.py
test 0 : is a 5 pounds bill
test 1 : is a 5 pounds bill
test 2 : is a 10 pounds bill
test 3 : is a 10 pounds bill
test 4 : is a 5 pounds bill
test 5 : is a 10 pounds bill
```