# Python Lab 5 Solutions

**Question 1.** Days, hours, minutes as a tuple.

```
total_minutes = 3456
days = total_minutes // (24*60)
extra_minutes = total_minutes % (24*60)
hours = extra_minutes // 60
minutes = extra_minutes % 60
result = (days,hours,minutes)
```

**Question 2.** First check using numbers (any different number values for a and b).

```
a = 3
b = 5
print('Before: a=',a,', b=',b)
a = a + b
b = a - b
a = a - b
print('After:  a=',a,', b=',b)
```

*Output:*

```
Before: a= 3 , b= 5
After:  a= 5 , b= 3
```

Second check using some other data type, e.g., strings. Note that the question mentions swapping "values" not necessarily "numbers".

```
a = '3'
b = '5'
# same code
```

*Output:*

```
Before: a= 3 , b= 5
TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

**Conclusion:** yes for numbers, no for strings

**Question 3.** Mutable: B (list), C (set), E (dictionary) Immutable: A (string), D (tuple)

```
A[0] = 'h'
B.append('mouse')
C.add('mouse')
D[0] = 'mouse'
E['dog'] = 'mouse'
```

**Question 4.** Replace, lower case, split and remove duplicates.

```
L = S.replace('.','').replace(',','').split()
# modified
L = S.replace('.','').replace(',','').lower().split()
L = list(set(L))
```

**Question 5.** Lots of possibilities, for example:

```
valid = ((cardA[0] in {'Hearts','Diamonds','Clubs','Spades'}) and
         (cardA[1]>=2) and (cardA[1]<=14))
flush = (cardA[0]==cardB[0])
straight = (cardA[1]==cardB[1]+1) or (cardA[1]==cardB[1]-1)
royal_flush = flush and straight and ((cardA[1]==14) or (cardB[1]==14))
```

This answer uses a set of strings for the valid suits but you could use the operator *in* with a list or tuple also.

Two cards are a *flush* if they are the same suit (index 0 of the tuple).
Two cards are a *straight* if the numbers are consecutive (index 1 of the tuple).
Two cards are a *royal flush* if they are numbers Ace and King (index 1) from the same suit (index 0). *Here we can reuse flush and straight if we wish.*

**Question 6.**

```
import random
def count_heads(N):
    """Count number of heads from N flips of a fair coin"""
    count_heads = 0
    for i in range(N):
        flip = random.randint(0,1)
        if (flip==0):
            print('Tails')
        else:
            print('Heads')
            count_heads = count_heads+1
    return(count_heads)

print(count_heads(10))
```

Output ($N = 10$): *One possible run, will be similar up to random variation.*

```
Heads
Heads
Heads
Heads
Heads
Tails
Tails
Tails
Tails
Heads
6
```

Visual check is that there really are 6 Heads printed out, so we get some confirmation.

**Question 7.** Docstring:

```
    """Count number of indices where the arguments A and B
    have the same value."""
```

Call with different types (needs to involve two of string, list, tuple), e.g.

```
mystery("abc",["a","b","c"])
mystery("abc",("a","b","c"))
mystery([1,2,3],(1,2,3)])
```

**Question 8.**

(a) List of tuples, where each tuple has two elements, both strings.

*Remember list is square brackets [], tuple is round brackets () and string is quotes.*

(b) Birthdays

```
for b in birthdays:
    name = b[0]
    dob = b[1]
    day,month,year = dob.split()
    # or ... day,month,year = str.split(dob)
    print(name,'was born in',month)
# or
# for i in range(len(birthdays)):
#     b = birthdays[i]
#     ... etc
```

(c) Issue: gives only one Guido (the one born 1570)
Reason: keys are unique in a dictionary
Possible solution: dictionary with key being name and values being a list of birthdays (other possibilities)

*Just for interest, the characters here are Boris Johnson, Prince Harry, Donald Trump, Guido van Rossum (creator of Python) and Guido Fawkes (better known as Guy Fawkes of fireworks fame), all with their real birthdays.*