Question 1:

```
def disjoint(m1, m2):
    # either of two dictionaries is empty will return true
    if m1 == {} or m2 == {}:
        return True
    else:
        # extracts all items from two dictionaries
        items_m1 = m1.items()
        items_m2 = m2.items()
        # compares them and find items in common
        # if there is no common items, return true
        if len(items_m1 & items_m2) == 0:
            return True
        else:
            return False
```

Question 2:

```
class Rectangle:
    # constructor body
    # with the given x, y, w, h
    def __init__(self, x, y, w, h):
        self._x = x
        self._y = y
        self._width = w
        self._height = h

    # return the field values
    def getx(self):
        return self._x

    def gety(self):
        return self._y

    def get_width(self):
        return self._width

    def get_height(self):
        return self._height

    def __str__(self):
        return '{(%d, %d), %dx%d}' % (self.getx(), self.gety(), self.get_width(), self.get_height())

    # define a merge function
    def merge(self, rect):
        x = min(self.getx(), rect.getx())  # find the x coordinate
        y = max(self.gety(), rect.gety())  # find the y coordinate
        # calculate the width and the height
        to_x = max(self.getx() + self.get_width(), rect.getx() + rect.get_width())
        to_y = min(self.gety() - self.get_height(), rect.gety() - rect.get_height())
        w = to_x - x
        h = y - to_y
        return Rectangle(x, y, w, h)
```

Question 3:

```
def find_matches(name, keywords):
    line_cnt = 0
    # creat an empty list
    res = []
    # extract every line
    for line in open(name).readlines():
        line_cnt += 1
        # split them
        sp = line.split()
        found = False
        for word in sp:
            if word in keywords:
                found = True
        if found:
            res.append(line_cnt)
    # return -1 when the list is empty
    if len(res) == 0:
        return [-1]
    return res
```

Question 4:

```
from critter import Critter
import random


class Quoll(Critter):

    DAY = 0
    NIGHT = 1

    # constructor body
    def __init__(self):
        self._eaten = 0
        self._steps = 0
        self._direction = 0
        self._trip_length = random.randint(1, 2) * 10

    def eat(self):
        if self.get_time_of_day() == 1:
            self._eaten += 1
            return True
        else:
            return False

    def get_time_of_day(self):
        return random.randint(0, 1)

    def move(self):
        self._steps += random.randint(1, 2) * 10
        if self._direction == 0:
            self._trip_length = random.randint(1, 2) * 10
        self._direction = (self._direction - 1 + 4) % 4

    def __str__(self):
```

```
    return 'I am a Critter. Type: Quoll, eaten: %d, current direction: %d, trip length: %d, steps: %d' % \
        (self._eaten, self._direction, self._trip_length, self._steps)
```

Note: When I copy the code from the Pycharm to the Word, there is something wrong with the indentation. I have attached all my python files (include test part) as well. If you have any queries, please check my source code. Thank you!