```
# DPP Assessed exercises 9

import re
import numpy.random as npr
import random
import string

# Q1 Suppose I want to generate a password of length n using a random combination of
# of the letters (a-z). Write a function that takes n and a seed value s as inputs an
# a string containing the password. Now all you have to remember for your password is
# seed value you used to create it.
def exercise1(n,s):
    random.seed(s)
    # suppose the password is lowercase
    letters = string.ascii_lowercase
    # randomly create a password from letters
    password = ''.join(random.choice(letters) for i in range(n))
    return password

# Suggested tests
exercise1(12,12)
# Should return 'lgrcddmqwruf' or 'pivqvlemalpi' (if using the random package rather
exercise1(20,999)
# Should return 'afbibztqnviiqfcllvxe' or 'zvcssrppezukuzudugei' (for random package)

# Q2 Suppose I have a list of phone number and I wish to extract the area codes of ea
# number. Write a function that takes a list of phone numbers as input and extracts t
# area code (assuming that the area code is enclosed in parentheses, e.g. the area co
# for (08) 03 49 98, would be 08).
def exercise2(phones):
    # set a re rule to match any item in the "( )"
    rule = re.compile(r'[(](.*?)[)]')
    area_code = re.findall(rule,phones)
    for i in range(len(area_code)):
        area_code[i] = "("+ area_code[i] +")"
    return area_code
# Suggested tests
ph_num1 = '(01) 12 05 25, (04) 25 23 11, (08) 03 49 98'
exercise2(ph_num1)
# Should return ['(01)', '(04)', '(08)']
ph_num2 = '(05) 73 43 12, (01) 11 34 67, (07) 91 62 46, (08) 04 23 81'
exercise2(ph_num2)
# Should return ['(05)', '(01)', '(07)', '(08)']

# Q3 I have a list of strings consisting of email addresses and I want to find the do
# (the part after the @). Write a function that extracts the characters after the @ s
# for each email address and returns them as a list.
def exercise3(emails):
    # set a re rule to match anything after "@"
    rule = re.compile(r'[@](.*)')
    e_address = list()
    for i in range(len(emails)):
        e_address.append(re.findall(rule,emails[i])[0])
    return e_address

# Suggested tests
exercise3(('myemail@ucd.ie','youremail@gmail.com'))
# Should return ['ucd.ie', 'gmail.com']
exercise3(('test1@ucd.ie','test2@gmail.com','test2@hotmail.com'))
# Should return ['ucd.ie', 'gmail.com','hotmail.com']


# Q4 I have a list of strings, each of which contains an email address. Write a funct
# that finds and returns all of the email addresses in a given list of strings. You m
# that all email addresses consist of a set of characters (from a-z) and digits (from
```

```python
# followed by an @ symbol, followed by another set of characters, followed by a full
# and finally a third set of characters (none of the email addresses will have specia
# characters, such as ? and !).
def exercise4(liststrings):
    # user name could contain letters "a-z", numbers "0-9"
    # domain name could consist of "@" and letters "a-z"
    # dot-something contain "." and letters "a-z"
    rule = re.compile("[a-z0-9]+@[a-z]+\.[a-z]+")
    e_address = list()
    for i in range(len(liststrings)):
        e_address.append(re.findall(rule,liststrings[i]))
    return e_address

# Suggested tests
junk_mails = ['John Koftaram <test@capahq.org> would like to connect on LinkedIn. How
exercise4(junk_mails)
# Should return [['test@capahq.org'], ['admin@handwheel.com'], ['manuelmedina@aol.com
contacts = ['Anne Bannon, email: annebannon72@gmail.com','Conor Darcy, phone: (01) 12
exercise4(contacts)
# Should return [['annebannon72@gmail.com'], ['conordarcy@icloud.com'], ['eamonnfriel
```