Exam in Advanced Programming in Python (DAT515)

Chalmers University of Technology
14 January 2023, 14:00 to 18:00, SB Multisal, Johanneberg
Examiner: Aarne Ranta aarne@chalmers.se
Tel. 1082, mobile 0729 74 47 80

Write your answers directly below the questions. You can of course use separate sheets of paper to draft and experiment, but only the question papers will be graded. This reflects the fact that the answers can and should be short.

You will need 15 points out of 30 in questions 1-5 of this exam to get accepted with the grade that your lab work allows.

If you have done extra labs (colouring or clustering) you will also need to get half of the points for the corresponding extra questions, in order to justify the extra points. If you have not done extra labs, your answers to those questions will not be graded.

The final result of this exam will be reported as 3, 4, 5, or rejected. As specified in the course plan, you will get

- grade 5 if you have at least 50 points from the labs, at least half of the points of the bonus questions corresponding to your labs, and at least 15 points from questions 1-5.
- grade 4 if you have at least 40 points from the labs, at least half of the points of the bonus questions corresponding to your labs, and at least 15 points from questions 1-5.
- grade 3 if you have at least 30 points from the labs and at least 15 points from questions 1-5.
- grade U otherwise

For your reference: the syntax of (the relevant parts of) Python

```
<stm> ::= <decorator>* class <name> (<name>,*)?: <block>
      |   <decorator>* def <name> (<arg>,*): <block>
      |   import <name> <asname>?
      |   from <name> import <imports>
      |   <exp>,* = <exp>,*
      |   <exp> <assignop> <exp>
      |   for <name> in <exp>: <block>
      |   <exp>
      |   return <exp>,*
      |   yield <exp>,*
      |   if <exp>: <block> <elses>?
      |   while <exp>: <block>
      |   pass
      |   break
      |   continue
      |   try: <block> <except>* <elses> <finally>?
      |   assert <exp> ,<exp>?
      |   raise <name>
      |   with <exp> as <name>: <block>
<decorator> ::= @ <exp>
<asname>    ::= as <name>
imports     ::= * | <name>,*
<elses>     ::= <elif>* else: <block>
<elif>      ::= elif exp: <block>
<except>    ::= except <name>: <block>
<finally>   ::= finally: <block>
<block>     ::= <stm> <stm>*
<exp> ::= <exp> <op> <exp>
      |   <name>.?<name>(<arg>,*)
      |   <literal>
      |   <name>
      |   ( <exp>,* )
      |   [ <exp>,* ]
      |   { <exp>,* }
      |   <exp>[exp]
      |   <exp>[<slice>,*]
      |   lambda <name>*: <exp>
      |   { <keyvalue>,* }
      |   ( <exp> for <name> in <exp> <cond>? )
      |   [ <exp> for <name> in <exp> <cond>? ]
      |   { <exp> for <name> in <exp> <cond>? }
      |   { <exp>: <exp> for <name> in <exp> <cond>? }
      |   - <exp>
      |   not <exp>
<keyvalue> ::= <exp>: <exp>
<arg>      ::= <name>
           |   <name> = <exp>
           |   *<name>
           |   **<name>
<cond> ::= if <exp>
<op>   ::= + | - | * | ** | / | // | % | @
       |   == | > | >= | < | <= | != | in | not in | and | or
<assignop> ::= += | -= | *=
<slice> ::= <exp>? :<exp>? <step>?
<step>  ::= :<exp>?
```

Question 1 (12 p). What is the *value* (or possibly error) of the following expressions? Remember that **None** is also a value!

- `'madam'.reverse()`
  Answer:

- `list(range(1, 10, 2))`
  Answer:

- `{{n} for n in range(5)}`
  Answer:

- `{n: {n} for n in range(5)}`
  Answer:

- `len({print(n) for n in range(5)})`
  Answer:

- `{1, 2, 1} != {1, 2}`
  Answer:

- `[] is []`
  Answer:

- `{} == set()`
  Answer:

Question 2 (3 p). Write a lambda expression for a function that takes two lists and returns a set that contains all elements in the two lists. For example, if this expression is bound to the variable **e**, then we should have

```
e([1, 2, 3], [2, 3, 5, 6]) == {1, 2, 3, 5, 6}
```

Question 3 (6 p). Consider a dictionary of the following form (assumed to contain all countries of the world, with the names of the countries as keys):

```
countries = {
 'Afghanistan': {'capital': 'Kabul', 'area': 652230, 'population': 36643815,
'continent': 'Asia', 'currency': 'afghani'},
 'Albania': {'capital': 'Tirana', 'area': 28748, 'population': 3020209,
'continent': 'Europe', 'currency': 'lek'},
 'Algeria': {'capital': 'Algiers', 'area': 2381741, 'population': 41318142,
'continent': 'Africa', 'currency': 'dinar'},
# etc
}
```

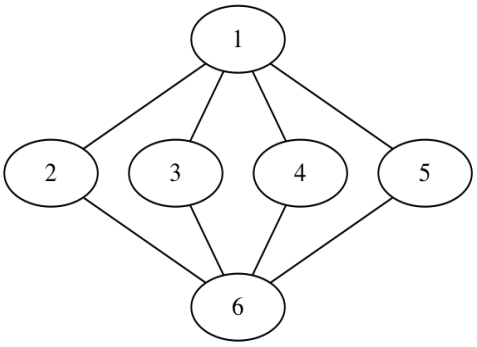
Write a Python expression whose value is the set of the names of all countries outside Europe.

Answer:

Also write an expression that answers the query *How many countries outside Europe use the euro as their currency?* as an integer. The euro is identified with the string `'euro'`.

Answer:

Question 4 (3 p). Write the adjacency list of the following graph as a Python dictionary.



Question 5 (6 p). The following class defines rectangles in terms of their length and width:

```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def area(self):
        return self.length * self.width

    def perimeter(self):
        return 2 * self.length + 2 * self.width
```

Now, squares are a special case of rectangles, where the length is the same as the width. Write a definition of class Square as a subclass of Rectangle, so that it makes a maximal use of inheritance - that is, does not override anything that ca be inherited. It should also enable the creation of a Square with just one argument, the length of the side.

In addition, use these classes to write Python expressions for

- a rectangle R with length 3 and height 2
  answer:

  R =

  NOTE: height -> width, corrected on campus

- the area of R
  answer:

- a square S whose side is 4
  answer:

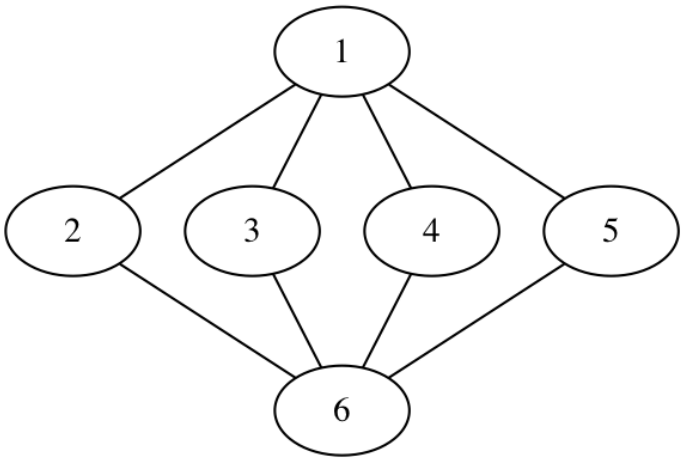
  S =

- the area of S
  answer:

Bonus question on graph colouring (6 p). Answer this question if and only if you have submitted the extra lab on graph colouring (be it one or two parts - the question is the same in both cases).

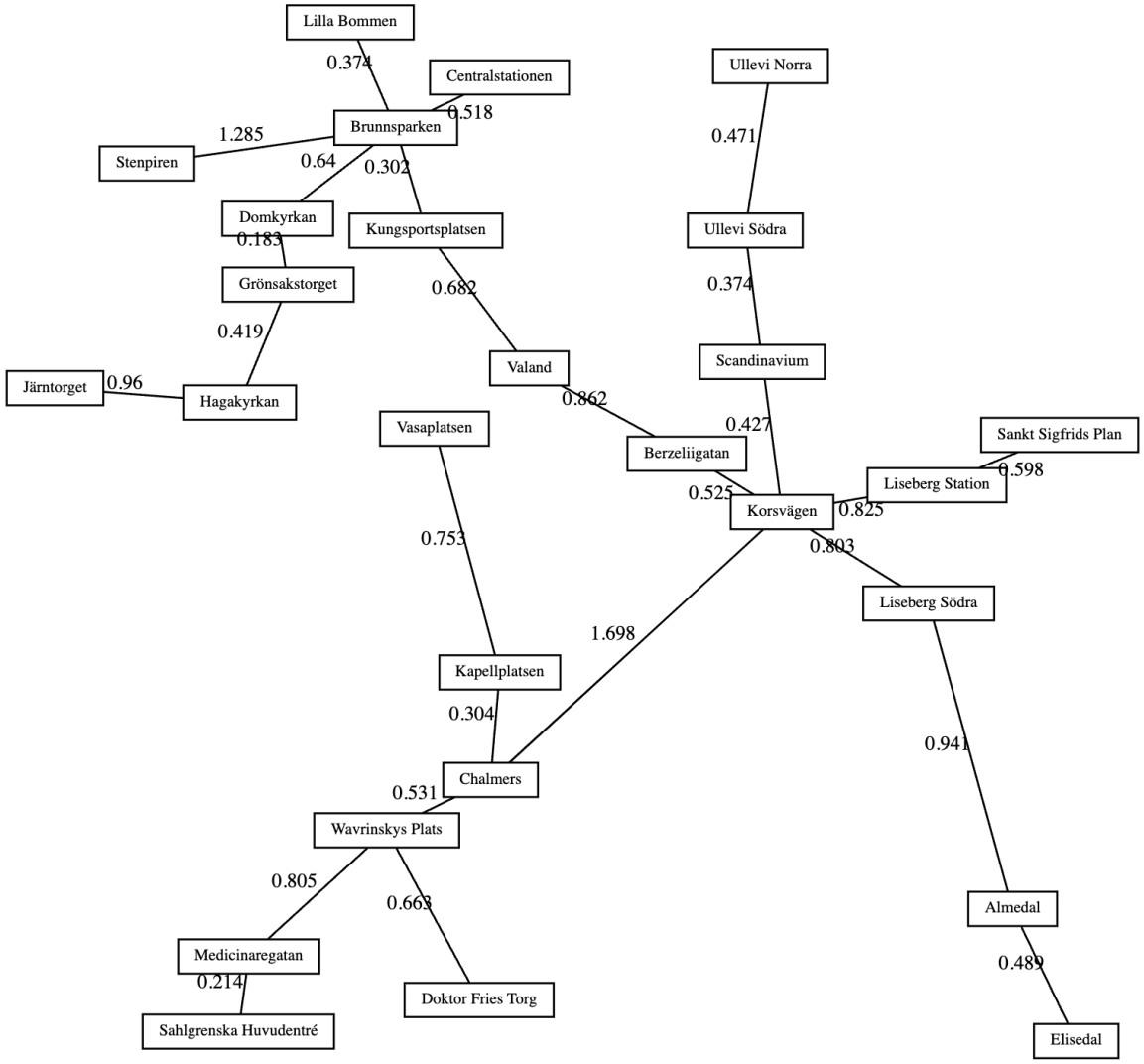
Use the simplify-select algorithm to colour the following graph with just three colours. Show

- an order in which the vertices can be removed in the simplify phase,
- how colours are then selected for each vertex at a time.



Bonus question on clustering (6 p). Answer this question if and only if you have submitted the extra lab on clustering.

The following graph shows a subset of the Gothenburg tram network. Each edge is marked by the geographical distance between the stops that it connects. Using the distances as weights, show the k-spanning tree clusters with k = 2, 3, 4. It is enough to state which edges are removed in each case. The aim is to remove the heaviest edges first



.