Facit och kommentarer till tenta 2023-10-03 i DA2005 och DA2004

Del A: flervalsfrågor

1. *C*
2. *C*
3. *D*
4. *A*
5. *A*
6. *A*
7. *C, D, E*
8. *A, B, C, D*
9. *D*
10. *E*

Del B: kodfrågor

11. Raden `assert x > 0` kräver att villkoret $x > 0$ är uppfyllt när man kommer till raden. Om det är uppfyllt händer ingenting, men om det inte är uppfyllt avbryts programmet med ett felmeddelande om var och varför.

12. Ett fel är att `left_len` lämnats oinitierat. Variabeln bör nollställas innan första loopen. Ett annat fel är att beräkningen av `n_spaces` är fel: det blir inte en rak vänsterkant på andra kolumnen om man inte gör korrigeringen `n_spaces = left_len - len(key)`. Fullständig kod:

```
def tabulate_dict(d):
    '''
    Tabulate the dictionary d, assumed to contain strings as keys and values.
    The right column is properly left indented.
    '''
    left_len = 0   # Rättelse
    for key, val in d.items():
        if len(key) > left_len:
            left_len = len(key)

    for key, val in d.items():
        n_spaces = left_len - len(key)    # Rättelse
        spacer = ' ' * n_spaces
        print(key, spacer, val)
```

13. Här är en möjlig lösning. Notera att jag använder en hjälpfunktion för att göra koden mer lättläslig.

```
def count_swedish_alphabet(s):
    swe_chars = 'åäöÅÄÖ'
    count = 0
    for c in s:
        if is_swedish_char(c):
            count += 1
    return count

def is_swedish_char(c):
    if ord(c) >= ord('a') and ord(c) <= ord('z'):
        return True
    elif ord(c) >= ord('A') and ord(c) <= ord('Z'):
```

```
        return True
    elif c in 'åäöÅÄÖ':
        return True
    else:
        return False
```

14.

A.
```
def count_empty_lines(filename):
    n_empty = 0
    with open(filename) as h:
        for line in h:
            if len(line) < 2:
                n_empty += 1
    return n_empty
```

B.
```
def count_empty_lines(filename):
    n_empty = 0
    try:
        with open(filename) as h:
            for line in h:
                if len(line) < 2:
                    n_empty += 1
    except:
        print(f'Warning: could not read file {filename}')
    return n_empty
```

15.
```
def compose(f, g):
    return lambda x: f(g(x))
```

Funktionen tar funktionerna f och g som kombineras ihop i en ny funktion, skapad med ett lambda-uttryck.

16.
```
class Polynomial:
    def __init__(self, coeffs):
        self._coeffs = coeffs

    def multiply(self, factor):
        self._coeffs = [factor * c for c in self._coeffs]

    def __str__(self):
        return poly_to_string(self._coeffs)
```