Facit och kommentarer till tentamen 2025-03-10 i DA4003

Del 1: flervalsfrågor (1p per fråga)

1. C
2. B, C
3. A[1]
4. A, C, E
5. E
6. D
7. C
8. A, B, E

Del 2: kodfrågor

9. (a) Möjlig lösning:

```
int* divisors(int n) {

  int* out = malloc(sizeof(int) * (n - 1));

  for (int d = 1; d < n; d++) {
    if (n % d == 0)
      *(out + (d - 1)) = d;
    else
      *(out + (d - 1)) = 0;
  }

  return out;
}
```

(b) Möjlig lösning:

```
int* divisors_goto(int n) {

  int* out = malloc(sizeof(int) * (n - 1));

  int d = 1;
 start_loop:
  if (n % d == 0)
    *(out + (d - 1)) = d;
  else
    *(out + (d - 1)) = 0;
  d++;
  if (d < n)
    goto start_loop;

  return out;
}
```

(c) Möjlig lösning:

[1] Tanken var att det skulle stå `q += 4` i koden på uppgiften, men i praktiken spelar det ingen roll och resultatet blir `0` oavsett (iaf med gcc).

```
int is_abundant(int n) {

  int s = 0;
  int *divs = divisors(n);

  for (int i = 0; i < n - 1; i++)
    s += *(divs + i);

  return (s > n);
}
```

10. (a) Möjlig lösning:

```
class Length {

    public Integer min;
    public Integer sec;

    public Length(Integer m,Integer s) {
        min = m;
        sec = s;
    }
}
```

(b) Möjlig lösning:

```
    public void add(Integer m,Integer s) {
        Integer temp_s = this.sec + s;

        this.min += m + temp_s / 60;
        this.sec = temp_s % 60;
    }
```

(c) Möjlig lösning:

```
class Song {
    public String artist;
    public String name;
    public Length length;

    public Song(String a,String n,Length l) {
        artist = a;
        name = n;
        length = l;
    }

    public String toString() {
        return (artist + " - " + name + " (" +
                length.min + "," + length.sec +")");
    }
}
```

11. (a) Möjlig lösning:

```
conjunctify :: [String] -> String
conjunctify [] = ""
conjunctify [s] = s
conjunctify [s1,s2] = s1 ++ " and " ++ s2
conjunctify (s:ss) = s ++ ", " ++ conjunctify ss
```

(b) Möjlig lösning:

```
iterate :: (a -> a) -> a -> Int -> [a]
iterate f x 0 = []
iterate f x n = x : iterate f (f x) (n - 1)
```

12. (a) Möjlig lösning (snitt är ett krav för full pott):

```
suffix(Suff,S) :-
    string_to_list(Suff,ListSuff),
    string_to_list(S,ListS),
    append(_,ListSuff,ListS),
    !.
```

(b) Möjlig lösning:

```
max3(X,Y,Z,X) :- X >= Y, X >= Z.
max3(X,Y,Z,Y) :- Y >= X, Y >= Z.
max3(X,Y,Z,Z) :- Z >= X, Z >= Y.

max_tree(leaf(X),X).
max_tree(branch(X,Left,Right),M) :-
    max_tree(Left,MaxLeft),
    max_tree(Right,MaxRight),
    max3(MaxLeft,MaxRight,X,M).
```