Facit och kommentarer till tentamen 2022-04-21 i DA4003

Del 1: flervalsfrågor (1p per fråga)

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

Del 2: kodfrågor

9. (a) Möjlig lösning:

```
int length(char* s) {

  int c;

  for (c = 0;; c++)
    if (s[c] == '\0')
      break;

  return c;
}
```

(b) Möjlig lösning:

```
int length(char* s) {

  int c = 0;

 start_loop:
  if (s[c] != '\0') {
    c++;
    goto start_loop;
  }

  return c;
}
```

(c) Möjlig lösning:

```
void reverse(char* s) {

  int l = length(s);

  for (int i = 0; i < l/2; i++) {
    char temp = *(s + i);
    *(s + i) = *(s + l - i - 1);
    *(s + l - i - 1) = temp;
  }
}
```

10. Möjlig lösning:

```
class Wordle {

    private String secret;

    public Wordle(String s) {
        secret = s;
    }

    public String guess_word(String guess) {

        String out = "";

        for (int i = 0; i < 5; i++) {
            char c = guess.charAt(i);

            if (c == secret.charAt(i)) {
                out += "G";
                continue;
            }

            String temp = secret.substring(0,i) + secret.substring(i+1);

            if (temp.contains("" + guess.charAt(i))) {
                out += "Y";
            } else {
                out += "X";
            }

            // Alternative version which doesn't use substring and contains:
            // boolean changed = false;
            //
            // for (int j = 0; j < 5; j++) {
            //      if (i != j && c == secret.charAt(j)) {
            //          out += "Y";
            //          changed = true;
            //          break;
            //      }
            //  }
            //
            // if (!changed)
            //     out += "X";
        }

        return out;

    }
}
```

11. (a) Möjlig lösning:

```
data Rank = Num Int
          | Jack
          | Queen
          | King
          | Ace
  deriving (Show,Eq)
```

(b) Möjlig lösning:

```
isValidRank :: Rank -> Bool
isValidRank (Num n) = 2 <= n && n <= 10
isValidRank _ = True
```

(c) Möjlig lösning:

```
fullDeck :: [(Suit,Rank)]
fullDeck = [ (s,r) | s <- [Hearts,Diamonds,Clubs,Spades], r <- allRanks ]
  where
    allRanks :: [Rank]
    allRanks = map Num [2..10] ++ [Jack,Queen,King,Ace]
```

12. (a) Möjlig lösning:

```
inc_tree(leaf(X),leaf(Y)) :- Y is X + 1.
inc_tree(branch(X,L,R),branch(Y,L2,R2)) :-
    Y is X + 1,
    inc_tree(L,L2),
    inc_tree(R,R2).
```

(b) Möjlig lösning:

```
list_delete(_, [], []).
list_delete(X,[X|L1], L2) :- list_delete(X,L1,L2).
list_delete(X, [Y|L1], [Y|L2]) :- list_delete(X,L1,L2).
```