

Facit och kommentarer till tentamen 2025-04-23 i DA4003

Del 1: flervalsfrågor (1p per fråga)

1. C
2. B, E
3. A
4. E
5. B
6. E
7. **Fel i uppgiften och inget alternativ var rätt.** Rätt svar skulle ha varit [1, 3, 7], vilket var tänkt att vara alternativ D. Jag tog bort uppgiften helt när jag rättade och drog bort 1 från alla betygsgränser.
8. E

Del 2: kodfrågor

9. (a) Möjlig lösning:

```
void swap(int* p, int* q) {  
    int temp = *p;  
    *p = *q;  
    *q = temp;  
}
```

- (b) Möjlig lösning:¹

```
int x = 42;  
int y = 58;  
  
int *p;  
p = &x;  
  
int *q;  
q = &y;  
  
swap(p, q);  
  
printf("%d %d\n", *p, *q);
```

- (c) Möjlig lösning:

```
void swap_many(int* p, int* q, int n) {  
  
    for (int i = 0; i < n; i++) {  
        swap(p+i, q+i);  
    }  
}
```

10. (a) Möjlig lösning:

¹För full pott måste man lösa det så att p och q innehåller vettiga adresser som pekar på något, så t.ex. `int *p = 42` och `int *q = 58` ger bara 1 poäng då det inte funkar.

```
// public framför är OK
abstract class LinkedList {

    abstract Integer max();

}
```

(b) Möjlig lösning:

```
class End extends LinkedList {

    public Integer value;

    public End(Integer v) {
        value = v;
    }

    public Integer max() {
        return value;
    }

}
```

(c) Möjlig lösning:

```
class Node extends LinkedList {

    public Integer value;
    public LinkedList next;

    public Node(Integer v,LinkedList n) {
        value = v;
        next = n;
    }

    public Integer max() {
        Integer temp = next.max();

        if (temp > value) {
            return temp;
        } else {
            return value;
        }
    }

}
```

11. (a) Möjlig lösning:

```
data Nucleotide = A | C | G | T
    deriving (Show)
```

(b) Möjlig lösning:

```
complement :: Nucleotide -> Nucleotide
complement A = T
complement T = A
complement C = G
complement G = C
```

(c) Möjlig lösning:

```
type DNA = [Nucleotide]    -- Behövs inte för poäng

reverseComplement :: DNA -> DNA
reverseComplement = reverse . map complement
```

(d) Möjlig lösning:

```
kmers :: Int -> DNA -> [DNA]
kmers n xs | length xs == n = [xs]
           | otherwise = take n xs : kmers n (drop 1 xs)
```

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

```
split(XS,N,Start,Middle,End) :-
    length(Start,N),
    length(End,N),
    append(Start,Middle,YS),
    append(YS,End,XS),
    !.
```

(b) Möjlig lösning:

```
permute_middle(XS,N,Out) :-
    split(XS,N,Start,Middle,End),
    permutation(Middle,P),
    append(Start,P,AlmostThere),
    append(AlmostThere,End,Out).
```