

# Facit och kommentarer till tentamen 2026-04-15 i DA4003

## Del 1: flervalsfrågor (1p per fråga)

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

## Del 2: kodfrågor

9. (a) Möjlig lösning:

```
int* divisors(int n) {  
  
    int c = 0;  
  
    // First calculate how many divisors there are  
    // Obs: start from 1 to avoid division by 0  
    for (int i = 1; i < n; i++)  
        if (n % i == 0)  
            c++;  
  
    // Then allocate c + 1 numbers  
    int *out = malloc((c + 1) * sizeof(int));  
  
    // Write the first part of the output  
    *out = c;  
  
    // Start counting from second index for the rest of the output  
    int j = 1;  
  
    for (int i = 1; i < n; i++)  
        if (n % i == 0) {  
            *(out + j) = i;  
            j++;  
        }  
  
    return out;  
}
```

- (b) Möjlig lösning:

```
int x = 0;  
  
printf("Write a number: ");  
scanf("%d",&x);  
  
int* divs = divisors(x);  
  
int sum = 0;
```

```

for (int i = 0; i < *divs; i++)
    sum += *(divs + i + 1);

if (sum == x)
    printf("The number is perfect!\n");
else
    printf("The number is not perfect!\n");

```

10. (a) Möjlig lösning:

```

abstract class List {

    abstract Integer length();

}

```

(b) Möjlig lösning:

```

class Empty extends List {

    public Empty() { }

    public Integer length() {
        return 0;
    }

}

```

(c) Möjlig lösning:

```

class Cons extends List {

    public Integer head;
    public List tail;

    public Cons(Integer x, List xs) {
        head = x;
        tail = xs;
    }

    public Integer length() {
        return (1 + tail.length());
    }

}

```

11. (a) Möjlig lösning:

```

sumFst :: [(Int,a)] -> Int
sumFst xs = sum (map fst xs)

```

(b) Möjlig lösning:

```

prefix :: String -> String -> Bool
prefix s1 s2 = take (length s1) s2 == s1

```

(c) Möjlig lösning:

```

data Letter = Vowel | Consonant
    deriving (Eq, Show)

typeOfLetter :: Char -> Letter
typeOfLetter c | elem c "aeiouy" = Vowel
                | otherwise = Consonant

```

12. (a) Möjlig lösning:

```
split(XS,N,YS,ZS) :-  
    length(YS,N),  
    append(YS,ZS,XS).
```

(b) Möjlig lösning:

```
evenOdd([],[],[]) :- !.  
evenOdd([X],[X],[]) :- !.  
evenOdd([X,Y],[X],[Y]) :- !.  
evenOdd([X,Y|XS],[X|E],[Y|O]) :- evenOdd(XS,E,O).
```

(c) Möjlig lösning:

```
length_list(empty,0).  
length_list(cons(_,XS),N) :-  
    length_list(XS,M),  
    N is 1 + M.
```