

UITWERKINGEN

Opgave 1

```
a)  wim, 3
    xyz
    abcdef, 6
    abcdef
    abcdef, 6
    xyz
    wim, 3
    XXX
    abcdef, 6
    XXX

b)  double benaderLnXPlus1 (double x, int aantalTermen) {
    double macht = 1.0,
        resultaat = 0.0;
    int    teken = -1;

    for (int i = 1; i <= aantalTermen; i++) {
        teken = -teken;
        macht = macht * x;
        resultaat += teken * macht / i;
    }

    return resultaat;
}
```

Het is toegestaan om een methode `macht()` te maken die x^n uitrekent.

```
c)  static final int AANTAL_RIJEN    = 15,
    AANTAL_KOLOMMEN = 35;

    int[][] matrix = new int [AANTAL_RIJEN][AANTAL_KOLOMMEN];

    boolean gebalanceerdeMatrix (int[][] m) {
        int somOnder = 0,
            somBoven = 0;
        for (int i = 0; i < m.length; i++) {
            for (int j = 0; j < m[0].length; j++) {
                if (i < j) {
                    somBoven += m[i][j];
                } else
                if (i > j) {
                    somOnder += m[i][j];
                }
            }
        }
    }
```

```
    return somOnder == somBoven;
}
```

d) -6
15
30
45
1
15
21
1

Opgave 2.

a) in de class Faculteit:

```
Faculteit () {
    studentenArray = new Student[MAX_AANTAL_STUDENTEN];
    aantalStudenten = 0;
}
```

```
void voegToe (Student student) {
    studentenArray[aantalStudenten] = student;
    aantalStudenten += 1;
}
```

b) in de class Student:

```
static final int GRENS_GOED_ECTS_PER_JAAR = 54; // 54.0 is ook goed
```

```
boolean goed () {
    return ectsPerJaar > GRENS_GOED_ECTS_PER_JAAR;
}
```

in de class Faculteit:

```
Faculteit aantalGoedeStudenten () {
    Faculteit resultaat = new Faculteit();

    for (int i = 0; i < aantalStudenten; i++) {
        if (studentenArray[i].goed()) {
            resultaat.voegToe(studentenArray[i]);
        }
    }

    return resultaat;
}
```

Bij retourneren van een array-object: -3

c) in de class Student:

```

static final int GRENS_TOP_INSCHRIJFDUUR = 3; // jaar

boolean topInschrijfduur () {
    return aantalJaarIngeschreven >= GRENS_TOP_INSCHRIJFDUUR;
}

in de class Faculteit:

Faculteit topStudenten () {
    Faculteit resultaat = new Faculteit();

    Faculteit goedeStudenten = goedeStudenten();
    for (int i = 0; i < goedeStudenten.aantalStudenten; i++) {
        if (goedeStudenten.studentenArray[i].topInschrijfduur()) {
            resultaat.voegToe(goedeStudenten.studentenArray[i]);
        }
    }

    return resultaat;
}

```

Bij geen gebruik van de methode goedeStudenten(): -4

Opgave 3.

- a)


```

int cijferSom (int n) {
    if (n < 10) {
        return n;
    }

    return cijferSom(n/10) + n%10;
}

```
- b)


```

boolean negenProef (int n) {
    if (n < 10) {
        return n == 9;
    }

    return n%9 == 0 && negenProef(cijferSom(n));
}

```
- c)


```

String voegHaakjesToe (String s, int i) {
    if (i == s.length()) {
        return "";
    }

    char c = s.charAt(i);
    int n = c - '0';

    return aantalKeer("(" , n)      +
           c                        +
           voegHaakjesToe(s, i+1) +

```

```
        aantalKeer(" ", n)      ;  
    }
```