Lösungsvorschlag Aufgabe 1 (10 + 5 = 15 Punkte):

a) Die Ausgabe sieht wie folgt aus:

l	m	r
0	4	8
0	1	3
2	2	3
3	3	3

b) Ist der zu suchende Wert nicht im Array enthalten, so wird *SearchException* geworfen: ist $l + 1 = r$, so ist $m = l$. Nun ist $a[m] \neq key$, wegen $a[m] = a[l] \leq key$ folgt damit $a[m] < key$. Im nächsten rekursiven Aufruf ist also $l = r = m$ und damit $a[m] > key$. Somit ist im darauffolgenden Aufruf $l - 1 = r$ und die Exception wird geworfen.

Lösungsvorschlag Aufgabe 2 (18 Punkte):

```
double setPairProd(double a[ ]) throws NoSetException {
 double res = 0.0;
 for (int i = 0; i < a.length; i++) {
   for (int j = i + 1; j < a.length; j++) {
     if (a[i] == a[j]) throw new NoSetException();
     res += a[i] * a[j];
   }
 }
 return res;
}
```

Lösungsvorschlag Aufgabe 3 (16 + 11 = 27 Punkte):

```
public class Element {
  ...
  void insert(T v, Comparator⟨T⟩ comp) {
    int c = comp.compare(v, this.val);
    if (c < 0) {
      if (this.left == null) this.left = new Element⟨T⟩(v);
      else this.left.insert(v, comp);
    } else if (c > 0) {
      if (this.right == null) this.right = new Element⟨T⟩(v);
      else this.right.insert(v, comp);
    }
  }

  void accept(Visitor⟨T⟩ vis) {
    vis.visit(this);
    if (this.left != null) this.left.accept(vis);
    if (this.right != null) this.right.accept(vis);
  }
...
}

class CountingVisitor implements Visitor⟨Integer⟩ {
  int count = 0;

  public void visit(Element⟨Integer⟩ e) {
    if (e.getVal() > 0) count++;
  }

  public int getRes() {
    return this.count;
  }
}
```

Lösungsvorschlag Aufgabe 4 (15 + 5 = 20 Punkte):

a)
$\langle \mathtt{n} \geq 0 \rangle$
$\langle \mathtt{n} \geq 0 \wedge 0 = 0 \wedge 0 = 0 \rangle$

```
i = 0;
```

$\langle \mathtt{n} \geq 0 \wedge \mathtt{i} = 0 \wedge 0 = 0 \rangle$

```
res = 0;
```

$\langle \mathtt{n} \geq 0 \wedge \mathtt{i} = 0 \wedge \mathtt{res} = 0 \rangle$
$\langle \mathtt{res} = \mathtt{i}^2 \wedge \mathtt{i} \leq \mathtt{n} \rangle$

```
while (i < n){
```

$\langle \mathtt{res} = \mathtt{i}^2 \wedge \mathtt{i} \leq \mathtt{n} \wedge \mathtt{i} < \mathtt{n} \rangle$
$\langle \mathtt{res} + 2 * (\mathtt{i} + 1) - 1 = (\mathtt{i} + 1)^2 \wedge \mathtt{i} + 1 \leq \mathtt{n} \rangle$

```
   i = i + 1;
```

$\langle \mathtt{res} + 2 * \mathtt{i} - 1 = \mathtt{i}^2 \wedge \mathtt{i} \leq \mathtt{n} \rangle$

```
   res = res + 2 * i - 1;
```

$\langle \mathtt{res} = \mathtt{i}^2 \wedge \mathtt{i} \leq \mathtt{n} \rangle$

```
}
```

$\langle \mathtt{res} = \mathtt{i}^2 \wedge \mathtt{i} \leq \mathtt{n} \wedge \mathtt{i} \not< \mathtt{n} \rangle$
$\langle \mathtt{res} = \mathtt{n}^2 \rangle$

b) Eine gültige Variante für die Terminierung ist $V = \mathtt{n} - \mathtt{i}$, denn die Schleifenbedingung $B = \mathtt{i} < \mathtt{n}$ impliziert $\mathtt{n} - \mathtt{i} \geq 0$ und es gilt:

$\langle \mathtt{n} - \mathtt{i} = m \wedge \mathtt{i} < \mathtt{n} \rangle$
$\langle \mathtt{n} - (\mathtt{i} + 1) < m \rangle$

```
   i = i + 1;
```

$\langle \mathtt{n} - \mathtt{i} < m \rangle$

```
   res = res + 2 * i - 1;
```

$\langle \mathtt{n} - \mathtt{i} < m \rangle$

Damit ist die Terminierung der einzigen Schleife in P gezeigt.

Lösungsvorschlag Aufgabe 5 (6 + 6 + 8 = 20 Punkte):

a) $substractAll :: Int \rightarrow [Int] \rightarrow Int$
$substractAll\ res\ [\,] = res$
$substractAll\ res\ (x : xs) = substractAll\ (res - x)\ xs$

b) $[6, 5, 4, 3, 2, 1]$

c) $mapMult :: (a \rightarrow b) \rightarrow MultTree\ a \rightarrow MultTree\ b$
$mapMult\ f\ (Node\ x\ xs) = Node\ (f\ x)\ (map\ (mapMult\ f)\ xs)$

Lösungsvorschlag Aufgabe 6 (4 + 6 + 10 = 20 Punkte):

a) i) $f(g(X), Y, Y), f(Z, Z, g(Z))$: occur failure X in $g(X)$
ii) $f(X, Z, b), f(Y, a, Y)$: mgu $X/b, Y/b, Z/a$
iii) $f(X, b, X), f(a, Y, Y)$: clash failure a/b
iv) $f(h(X, Y), Z), f(Z, h(b, a))$: mgu $X/b, Y/a, Z/h(b, a)$

b) `A / [1,3,2]`

c)
```
add(X,[],[X]).
add(X,[Y|XS],[X,Y|XS]) :- max(X,Y,Y).
add(X,[Y|XS],[Y|YS]) :- max(X,Y,X), add(X,XS,YS).
```