Question 1.

```
int h= –1; int p= –1;
while (p != k) {
    p= p + 1;
    // if b[0..h] does not contain a dup for b[p],
    // then move b[p] into the first segment
    if  (h < 0 || b[h] != b[p]) {
        h= h+1;
        b[h]= b[p];
    }
}
```

Question 2. This answer makes use of conditional expressions to give a simple solution.

```
public static int[][] reduceArray
                    (int [][] b, int r, int c) {
  int res[][]= new int[b.length-1][b[0].length-1];

  /** copy elements of b to res */
  for (int i= 0; i != res.length; i= i+1) {
     for (int j= 0; j != res[i].length; j= j+1) {
        res[i][j]= b[i < r? i: i+1][j < c ?j : j+1];
     }
  }
  return res;
}
```

Question 3a. mean(v(v > 0))

3b.

```
% Store the powers of x in powerx
powerx = (x*x) .* cumprod(x + zeros(1, n));

% store in s the signs of the terms
s = -1 * cumprod(-1 + zeros(1,n));

% Store the denominators in denom
fact = cumprod(1:(2*n – 1));
denom = fact(1:2:(2*n – 1));

% Store the solution in sol
sol = cumsum(signs .* powerx ./ denom);
```

Question 4.

a. 22	b. 11
c. 11	d. false
e. 99	f. 99
g. true	h. ERR

Question 5.

```
/** = number of occurrences of c in s */
public static int occ(char c, String s) {
   if (s.length() == 0) {
       return 0;
   }
   return (s.charAt(0) == c ? 1 : 0)  +
           occ(c, s.substring(1));
}
```

```
/** = s but with duplicate characters removed */
public static String remDup(String s) {
   if (s.length() < 2)
       return s;
   if (occ(s.charAt(0), s.substring(1)) > 0)
       return remDup(s.substring(1));
   return s.charAt(0) + remDup(s.substring(1));
}
```

Question 6

```
import java.util.*; // not needed on the test
public class Person {
  private String p; //Person's name
  private String town; // hometown
  private Vector friends; //friends

  /**Constructor: person p from town t with
                    friends f*/
  public Person(String p, String t,
                    Vector f) {
    this.p= p;
    town= t;
    friends= f;
  }

  /**Constructor: person "John Doe" from
          "Middle USA" with no friends */
  public Person() {
    this("John Doe", "Middle USA",
         new Vector());
  }

  /** = this person's name */
  public String getPerson() {
      return p;
  }

  /** = this person's home town */
  public String getTown() {
      return town;
  }

  /** = this person's friends */
  public Vector getFriends() {
      return friends;
  }

  /** Add p as a friend of this person */
  public void addFriend(Person p) {
      friends.add(p);
    }
}
```

```
import java.util.*; // not needed on the test
public class Band extends Person {
  private Vector members;

  /**Constructor: A Band called p from town t
        with friends f and members m */
  public Band(String p, String t, Vector f,
                    Vector m) {
      super(p, t, f);
      members= m;
    }

  /** = this band's members */
    public Vector getMembers() {
      return members;
  }

  /** Add p to this Band's members */
  public void addMember(Person p) {
      members.add(p);
  }

  /** Remove p from this Band's members */
  public void removeMember(Person p) {
      members.remove(p);
  }
}
```

Question 7.

7. (a) A `JFrame` uses a `BorderLayout` manager. A `JPanel` uses a `FlowLayout` manager. A `Box` uses a `BoxLayout` manager.

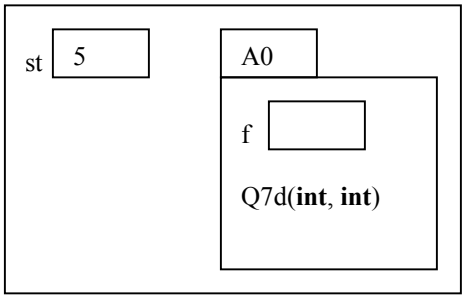
7(b) A type is a set of values together with the primitive or basic operations on them.

7(c) A static (or class) variable is declared in a class definition. A non-static (or instance) variable is declared in a class definition. A parameter is declared within the parentheses of the header of a method. A local variable is declared in the body of a method.

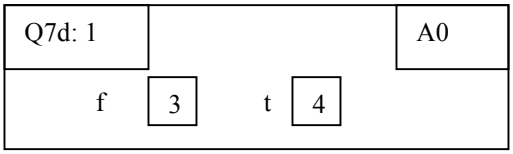
7(d)

Q7d's file drawer.

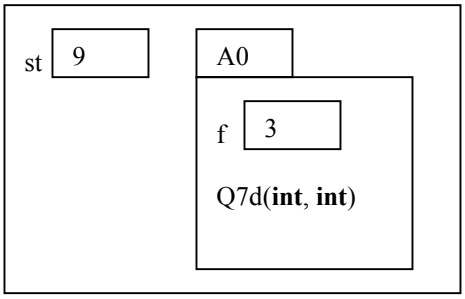
Step 1. Draw a new object of class `Q7d` (it goes in `Q7d`'s file drawer):



Step 2. Execute the constructor call `Q7d(3, 4)`. The frame for the call just after it is drawn is this:



Executing the call changes object A0 and static variable st so that everything looks like this:



Step 3. Yield as value of the new-expression the name of the newly created object, in this case, A0.

Question 8 (This is one of several possibilities).

```
/** Partition algorithm
   Pre: b[h] contains some value x
   Post: b[h..j-1] ≤ x = b[j]  and  b[j] ≤ b[j+1..k]*/
public static void partition(
  int[] b, int h, int k) {
  int j= h;
  int i= k;
  // inv: b[h..j-1] <= x = b[j]
  //      b[j+1..i] is unknown
  //      b[i+1..k] >= x
  while (j != i) {
    if (b[j+1] <= b[j]) {
        Swap b[j+1] and b[j];   j= j+1;
    }
    else {Swap b[j+1]and b[i];   i= i-1;
    }
  }
  return j;
}
```