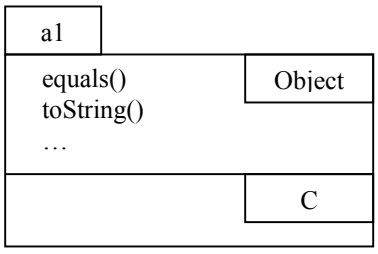
1. k= m+1; j= n;
/* invariant: b[m] = x,
b[m+1..j–1] <= x,
b[k+1..n] >= x */
while (j <= k) {
if (b[j] <= x)
{ j= j+1; }
else { Swap b[j] and b[k]; k= k – 1; }
}
// b[m] = x, b[m+1..k] <= x, b[k+1..n] >= x
Swap b[m] and b[k];

2. Note: such algorithms are actually used, when really large integers are to be maintained. The "base" would not be 10 but some other number like the largest int-1. If you are interested, look at Java API classes BigInteger and BigDecimal.

```
ds= 0; d[ds]= 0;
// invariant: as in question on final
while (ds != Math.max(bs,cs)) {
    if (ds < bs) { d[ds]= d[ds] + b[ds]; }
    if (ds < cs) { d[ds]= d[ds] + c[ds]; }
    d[ds+1]= d[ds] / 10;
    d[ds]= d[ds] % 10;
    ds= ds+1;
}
if (d[ds] > 0) ds= ds+1;
```

3. /** = "array b is triangular", i.e. each row i has 2*i+1 elements */
public static boolean isTriangular(**int**[][] b) {
// invariant: rows 0..k-1 are triangular
for (**int** k= 0; k != b.length; k= k+1) {
if (b[k].length != 2*k + 1)
return false;
}
return true;
}

4a.



4.b. d: 384
e: 64
f: 384
d = e: false
a: true
b: false

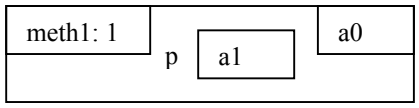
5. % series(n) is the cum sum of first n terms of :
% – 1/5 + 2/10 – 3/15 + 4/20 – 5/25 + 6/30 – …

```
function ans= series(n)
  numerator= 1:n;
  % set alt to (–1 + 1 – 1 + 1 – … )
  alt= cumprod( (–1) .* ones(n));
  denominator= alt .* (5 .* ones(n));
  ans= cumsum (numerator ./ denominator);
```

6a. A parameter is a variable that is declared within the parentheses of a method header. An argument is an expression that appears within the parentheses of a method call.

6b. Make a class abstract so that it cannot be instantiated (but its subclasses can be).

6c.



Argument of first call: a0
Argument of second call: a1

7.

```
/** An instance is a person and their assigned
    license plate number */
public class License {
  /** License plates have the form "NYS ddddd"
      where d denotes a digit. Assigned so far are
      "NYS 00001"..lastPlate */
  private static String lastPlate= "NYS 00000";

  /** Vector of all instances of this class that
      have been created */
  private static Vector licenses= new Vector();

  private String person; // The person
  private String plate;  // plate assigned to person

  /** Constructor: an instance with name p and
      plate n */
  private License(String p, String n) { }

 /** assign a new license plate number x (say) to
     person p and return a License new License
     (p, x). Maintain defs of fields of this class. */
  public static License assign(String p) { }

  /** = an instance for person p (if none, null) */
  public License getLicense(String p) { }

  /** = the name of the person with this license */
  public String getName() { }

 /** = the license number of this license */
  public String getNumber() { }
}
```

8a. this.name= name; **this**.ad= ad;

8b. super(p, ad); date= **new** Date();

8c. The folder has two partitions. The top one is named Student. It has two fields: name and ad. It has three methods: Student(String,Address), getName(), and getAddress(). The bottom one is named StudentAdmitted. It has one field, date, and two methods: StudentAdmitted(String, Address) and equals().

8d. return date.equals(sa.date) &&
getName().equals(sa.getName()) &&
getAddress().equals(sa.getAddress());