

Problem 1.

a) Let the classes A, B and C be

```
class A {
    int g;

    A (int g) {
        this.g = g;
    }

    void add (int x) {
        g += x;
    }
}

class B {
    int g;
    A a;

    B (int g) {
        this.g = g;
        this.a = new A(11);
    }

    void add (int x) {
        g += x;
    }
}

class C {
    int g;
    B b;

    C (int g) {
        this.g = g;
        b = new B(13);
    }
}
```

Further we have a program with the following statements/declarations

```
A a = new A(1);
B b = new B(2);
C c = new C(3);
PrintStream out = new PrintStream(System.out);

void println (A x, B y) {
    out.printf("%d %d\n", x.g, y.g);
}

void doSomething () {
    println(a, b);
    b.a.add(4);
    println(a, b);
    c.b.add(5);
    println(c.b.a, b);
    b = c.b;
    a = b.a;
    println(a, b);
    c.b.a.add(6);
    println(a, c.b);
    b = c.b;
    println(c.b.a, b);
}
```

What will be printed when the method doSomething() is called?

- b) The following heading of a method calculate() is given

```
double calculate (double x, int n)
```

This method calculates the first n terms of the series

$$1!/2x^3 - 2!/3x^4 + 3!/4x^5 - 4!/5x^6 + 5!/6x^7 - \dots$$

x^n is the notation for "x to the power n" and n! is the notation for "the factorial of n". The number of terms that should be calculated is given by numberOfTerms. Implement this method. Do this without using any methods from the class Math. Assume: numberOfTerms ≥ 1.

- c) Give the declaration of a variable "matrix" with as type a 2-dimensional array of char's with 8 rows and 4 columns. Use constants when necessary.

Program a method numberOfTops() that will be able, for any 2-dimensional array of int's, to count how many tops there are in the array. An int x in the array is a top if x is bigger than each of the 4 numbers above, below, to the left and to the right of x

examples

1 3 2 5	the only top is 9	1 9 1 1	no tops here
7 9 4 1	(9 is a top; it is bigger	9 1 1 1	
3 5 6 9	than 3, 5, 7 and 4)	1 1 9 1	

```
d) class Problem_1d {
    PrintStream out;
    int a, b;

    Problem_1d() {
        out = new PrintStream(System.out);
        a = 5;
        b = 6;
    }

    void print (int x, int y) {
        out.printf("%d %d\n", x, y);
    }

    int m1 (int c) {
        a = c + 1;
        b = a + 2;
        c = a + b;
        print(c, a);
        return c;
    }

    int m2 (int b, int c) {
        a = b - c;
        b = m1(a);
        print(c, b);
        c += 1;
        return b+c;
    }

    void start() {
        print (a, b);
        int c = m1(b);
        print(a, b);
        b = m2(c, a);
        print(a, b);
    }

    public static void main(String argv[]) {
        new Problem_1d().start();
    }
}
```

What will be printed when this program is executed?

Problem 2.

For every subproblem of problem 2, program sub problems in separate methods in the correct class. Use constants when necessary.

- a) Given is the following class:

```
class Coin {
    String country,
           grade;    // mint state, fine, good, fair or poor
    int printedValue,
        yearMinted;
}
```

The constructor and the methods are omitted, as they are not necessary for this problem.

Make a class CoinCollection. This class should be able to store a maximum of 2500 coins. Further the class should have a default constructor and a method add(). The default constructor should initialize the CoinCollection-object to an empty coin collection. The method add() should make it possible to add 1 coin to the coin collection.

- b) Program in the class CoinCollection a method

```
CoinCollection rareCoins ()
```

which, in a new CoinCollection-object, returns all the coins that are rare. For this problem it is defined that a coin is rare if its grade is "mint state" and its printed value is more than 10.

- c) Program in the class CoinCollection a method

```
void removeModernCoins ()
```

which removes all modern coins from the Coin Collection. For this problem it is defined that a coin is modern if the year it was minted is after 1945.

- d) Given is that the class CoinCollection contains a method

```
int coinsFrom (String origin)
```

that returns the total number of coins in the coin collection from the country origin. You can use this method without having to program it.

Now add to the class CoinCollection a method

```
int rareOldCoinsFrom (String origin)
```

which returns the total number of rare coins in the coin collection that are not modern coins and that are from the country origin. Program this method without using a for-, while or do-while statement.

Problem 3.

a) Write a recursive method

```
int count (int[] r, int i, int a) // 0 <= i <= r.length
```

that counts how many elements in r, starting from index position i, equal a

examples: assume the array r contains {1, 2, 3, 1, 2, 3, 1, 2, 1}

```
count(r, 0, 1) gives 4
count(r, 2, 1) gives 3
count(r, 4, 1) gives 2
count(r, 9, 1) gives 0
```

b) Given is that the class String contains a method

```
String substring (int start) // 0<=start<=length string
```

that returns the substring from the character on index position start till the last character (inclusive). The class String also contains a method

```
String substring (int start, int stop)
// 0<=start<=length string, stop <= length string
```

examples:

```
"abcdef".substring(3)    returns "def"
"abcdef".substring(3, 5) returns "de"
"abcdef".substring(0)    returns "abcdef"
"abcdef".substring(0, 3) returns "abc"
"abcdef".substring(6)    returns ""
"abcdef".substring(3, 0) returns ""
```

Write a recursive solution for the method

```
int count (String s, String pattern) // pattern length > 0
```

that counts the number of occurrences of the pattern in s.

examples:

```
count("abcabcabc", "ab") gives 3
count("aaaa", "a")       gives 4
count("aaaa", "aa")      gives 3
count("aaaa", "aaaa")    gives 1
count("abcd", "ba")      gives 0
```

grade:

	Problem a	b	c	d	total
1.	5	5	5	5	20
2.	2	6	6	3	17
3.	4	4			8
					-- +
					45

The grade E follows from the points P with the formula: $E = P / 5 + 1$