1 Ternary operator

Read the following code and, without executing it first, explain what it does.

```
int currentYear = 2012
print "When were you born (year)?"
String str = System.console().readLine();
int birthYear = Integer.parseInt(str);
int ageAprox = currentYear - birthYear;
String result = (ageAprox > 17) ? "" : "not "
println "It seems you are " + result + "an adult."
```

Hint: look at section “The Ternary Operator”.

2 Ende homage

Is there anything wrong with the following piece of code (hint: yes)? What does it do?

```
int i = 10
while (i < 5) {
    i++;
    println i
}
```

3 Yet another loop

What does the following piece of code do?

```
String str = System.console().readLine()
int i = Integer.parseInt(str)
while (i < 10) {
  i++;
  String str = System.console().readLine()
  int j = Integer.parseInt(str)
  if (j == 0) {
      break;
  } else if (j != 1) {
    println j;
  }
}
System.out.println("finished");
```

Hint: the reserved word `break` exits the current loop.

4 Prime numbers

Write a program that asks a number from the user, then says whether the number is prime or not. Remember that a number that is divisible by any number apart from 1 and itself is prime. You can use the modulo operator (if `a % b` is zero, then a is divisible by b).

5 Multiplication

Write a program that requests two numbers from the user and then outputs its product. You cannot use the "*" operator.

6 Division

Write a program that requests two numbers from the user and then outputs the quotient and the remainder, e.g. if the user enters 7 and 3, your program should ouput something like "7 divided by 3 is 2, remainder 1". You cannot use the "/" or "%" operators.

7 Naive sorting

Write a program that reads three numbers and prints them in order, from lowest to highest.

8 Maximising

Write a program that read a (arbitrarily long) sequence of positive numbers. The sequence is ended when the users enters "-1". At that point, the program must output the highest number in the sequence.

9 Going up!

Read an arbitrarily long sequence of positive numbers from the user, until -1 is entered. At that point, print "Yes" if the numbers were consecutive and increasing and "No" otherwise. Sequences "1,2,3,4,-1" and "5,6,7,8,9,10,11,-1" should output "Yes", but "2,3,5,6,7,-1", "10,9,8,7,-1", and "1,1,2,3,4,5,-1" should output "No".

10 You said high, I said low...

Modify your former program so that it outputs "Yes" when the numbers are consecutive, regardless of whether they go up or down. For example, both "2,3,4,5,6,-1" and "10,9,8,7,-1" should now result in "Yes".

11 Poker hands (harder)

Read five cards from the user. For each card, read the rank (1,2,3,4,5,6,7,8,9, 10,J,Q,K) and the suit ("spades", "hearts", "diamonds", "clubs"). Each of the five cards must be valid before accepting the next one. Once the program has the five cards, it should tell the user what is the best hand she has got, as per the following list (from best to worst):

Straigh flush: all cards are of the same suite and their ranks are consecutive. Note that they are probably not ordered as they were entered.

Poker: four of the five cards have the same rank.

Full House: three of a kind plus two of a kind.

Flush: all cards share the same suit, but are not consecutive.

Straight: all cards are consecutive, but not of the same suit.

Three of a kind: three of the five cards have the same rank.

Two pairs: two pairs (see below).

Pair: two of the five cards have the same rank.

Nothing: any other situation.

If you ever launch an online poker business, this could be one (very small) piece of it.