Practical 1

1 Question 1

Consider the following code:

```
Input: integer L, integer H
Output: ?
    integer x = L
    integer p = 1
    while x ⩽ H do
        if x is odd then
            p = p × x
        end if
        x = x + 1
    end while
    print "result is", p
```

(a) What does this algorithm do, that is, what is being computed in the variable p?
p is the multiplication of all odd numbers between , and including L and H

(b) Rewrite this code to use a for loop instead of a while loop.

```
Input: integer L, integer H
integer x=L
integer p=1
if x is even then
    x=x+1
end if
for i=L to H; i+=2 do
    p=p×i
end for
print "result is", p
```

2 Question 2

Design an algorithm in pseudocode that, when given some number n and numbers $a_0, a_1, ..., a_{n-1}$ as input, computes and outputs the second smallest of those numbers. In other words, the input/output specification looks something like
Input: integer n, integers $a_0, a_1, ..., a_{n-1}$
Output: second smallest of $a_0, a_1, ...a_{n-1}$
Finding the smallest is a problem given as an exercise for you to do during the lectures. So look back at (or now construct) your solution and think about how to modify it?
(Note you should not include a line in your solution that says something like sort $a_0, a_1, ..., a_{n-1}$ so they are in order from smallest to largest This would not be efficient, as sorting the integers is a more difficult problem than the one you are asked to solve.)

```
positive integer n
integers $a_0, a_1, a_2 ... a_{n-1}$
integer smallest = $a_0$
integer second = $a_0$
for i=1 to n do
    if $a_i \leqslant$ smallest then
        second=smallest
        smallest=$a_i$
    else
        if $a_i \leqslant$ second then
            second=$a_i$
        end if
    end if
print second
```

3 Question 3

Recall that a natural number k is called prime if it can be divided without remainder only by one and itself. Write a simple algorithm in pseudocode that tests whether a given number, i.e., a number that is given as input, is prime. You may use a condition such as "a divides b" and you can also use a command exit that will cause the algorithm to terminate (i.e. once you have found that k is not a prime you can stop).

```
Input: integer k
for i=2 to sqrt(K) do
    if k divides i and k≠i then
        exit
    end if
end for
print "The number is prime"
```

4 Question 4

Write a simple algorithm in pseudocode that takes as input the date (given as three integers for year, month and day) and a further integer k. The output should be the date k days after the input. You can, if you wish, make the simplying assumption that there are no leap years and, moreover, that every month has the same number of days (so a year contains 360 days divided into 12 30-day months, say). Also assume that for two integers a and b, a/b gives an integer and a % b gives the remainder. (For example 800/360 = 2 and 800 % 360 = 80 so 800 days from now is 2 years and 80 days away).

```
Input: integer Year, integer Month, integer Day, integer k
Year = Year + Floor(k/360)
Month = Month + Floor((k%360)/30)
Day = Day + (k%360)%30
Print "The Date is:", Year, "-", Month, "-", Day
```

4.1 Question 4 without assumptions

```
Input: integer Year, integer Month, integer Day, integer k
list month= [0,31,28,31,30,31,30,31,31,30,31,30,31]
list lmonth= [0,31,29,31,30,31,30,31,31,30,31,30,31]
Day=Day+k
while (days ⩾ month[Month] and Year%4≠0) or (days ⩾ lmonth[Month] and Year%4==0) then
    if Year%4==0 then
        Day=Day-lmonth[Month]
    else
        Day=Day-month[Month]
    end if

    if Month == 12 then
        Month=1
        Year=+1
    else
        Month+=1
    end if
end while
print Year,"-",Month,"-",Day
```