Assignment 1

Task 3

a.

Code:

```
local X Y=300 Z=30 in
    X = Y * Z
    {System.showInfo X}
end
```

Output:

```
9000
```

b.

Code:

```
local X Y in
    X = "This is a string"
    thread {System.showInfo Y} end
    Y = X
end
```

Output

```
This is a string
```

Answer to questions

- Y is printed because the code is compiled before executed
- The execution of the thread is halted until variable Y is assigned
- Y = X assigns the value of X to Y

Task 4

a.

Code:

```
fun {Max Number1 Number2}
    if Number1 > Number2 then
        Number1
    else
        Number2
    end
end
```

b.

Code:

```
proc {PrintGreater Number1 Number2}
    {System.showInfo {Max Number1 Number2}}
end

{PrintGreater 10 89}
```

Output:

```
89
```

Task 5

Code:

```
proc {Circle R} A D C Pi = 355.0/113.0 in
    A = Pi * R * R
    D = 2.0 * R
    C = Pi * D

    {System.showInfo "Areal: " #A}
    {System.showInfo "Diameter: " #D}
    {System.showInfo "Circumference: " #C}
end

{Circle 5.0}
```

Output:

```
Areal: 78.54
Diameter: 10.0
Circumference: 31.416
```

Task 6

Code:

```
fun {Factorial N}
    if N == 1 then
        N
    else
        N * {Factorial N-1}
    end
end

{System.showInfo {Factorial 10}}
```

Output:

```
3628800
```

Task 7

List implementation:

```
% a
fun {Length List}
    case List of _|Tail then
        1 + {Length Tail}
    else
        0
    end
end

% b
fun {Take List Count}
    if Count >= {Length List} then
        List
    else
        if Count == 0 then
            nil
        else
```

```
            case List of Head|Tail then
                Head|{Take Tail Count - 1}
            end
        end
    end
end

% c
fun {Drop List Count}
    if Count >= {Length List} then
        nil
    else
        case List of Head|Tail then
            if Count == 0 then
                Head|Tail
            else
                {Drop Tail Count - 1}
            end
        end
    end
end

% d
fun {Append List1 List2}
    if {Length List1} == 0 then
        List2
    else
        case List1 of Head|Tail then
            Head|{Append Tail List2}
        else
            List1
        end
    end
end

% e
fun {Member List Element}
    case List of Head|Tail then
        if Head == Element then
            true
        else
            {Member Tail Element}
        end
    else
        false
    end
end

% f
fun {Position List Element}
    case List of Head|Tail then
        if Head == Element then
            0
        else
```

```
            1 + {Position Tail Element}
        end
    end
end
```

Execution of functions:

```
% Defining the list
List = [2 9 8 4 5]
{System.showInfo "Original list:"}
{System.show List}

% a results
{System.showInfo "Task A results:"}
{System.showInfo {Length List}}

% b results
{System.showInfo "Task B results:"}
{System.show {Take List 2}}

% c results
{System.showInfo "Task C results:"}
{System.show {Drop List 2}}

% d results
{System.showInfo "Task D results:"}
{System.show {Append List List}}

% e results
{System.showInfo "Task E results:"}
{System.show {Member List 9}}
{System.show {Member List 6}}

% f results
{System.showInfo "Task F results:"}
{System.show {Position List 9}}
{System.show {Position List 4}}
```

Output:

```
Original list:
[2 9 8 4 5]
Task A results:
5
Task B results:
[2 9]
Task C results:
[8 4 5]
Task D results:
```

```
2|9|8|4|5|2|9|8|4|...|...
Task E results:
true
false
Task F results:
1
3
```

Task 8

a.

Code:

```
fun {Push List Element}
    Element|List
end

{System.show {Push List 2}}
```

Output:

```
[2 2 9 8 4 5]
```

b.

Code:

```
fun {Peek List}
    case List of Head|_ then
        Head
    end
end

{System.show {Peek List}}
```

Output:

```
2
```

c.

Code:

```
fun {Pop List}
    case List of _|Tail then
        Tail
    end
end

{System.show {Pop List}}
```

Output:

```
[9 8 4 5]
```