Homework 2 (Control flow, compound types)

Select and solve **5** tasks from this list.
Include all the solutions in **one** R script file, see `homework2_template.R` for a template.
When you're done, send them via courses.ipipan.edu.pl.
All the scripts will be examined by plagiarism detection software.
Do not use any facilities from external CRAN packages – implement all the algorithms from scratch.

Exercise 02.01. Write a function `logiderle()` with the following parameters:

- an integer vector `i` of length n (for some n),
- an integer vector `j` of the same length as `i`,
- a single natural number `n`.

If for all possible l it does not hold $1 \le \mathtt{i}_l \le \mathtt{j}_l \le \mathtt{n}$ and $\mathtt{i}_l > \mathtt{j}_{l-1}$, throw an error immediately.

The function should return a logical vector `w` of length `n` such that $\mathtt{w}_l$ == `TRUE` iff $(\exists p)\ l \in [\mathtt{i}_p; \mathtt{j}_p]$. For example, if $\mathtt{n} = 7$, `i` == `c`$(1,4)$, `j` == `c`$(1,6)$, then the result is (`TRUE`, `FALSE`, `FALSE`, `TRUE`, `TRUE`, `TRUE`, `FALSE`).

Exercise 02.02. Write a function `golden_ratio()` to determine a local minimum of a continuous real function $\mathtt{f} : [\mathtt{a}, \mathtt{b}] \to \mathbb{R}$. Parameters:

- a vectorized R function `f()`, i.e. an object such that for all n and $\mathtt{x} \in [\mathtt{a}, \mathtt{b}]^n$ it holds $\mathtt{f}(\mathtt{x}) \in \mathbb{R}^n$,
- a single numeric value `a`,
- a single numeric value $\mathtt{b} > \mathtt{a}$,
- a positive numeric value `eps` (by default equal to 10^{-16}),
- a natural number `maxiter` (by default equal to 100).

Algorithm: for $i = 1, 2, \ldots,$ `maxiter` do:

1. Let $x_L := b - \varphi(b-a)$ and $x_P := a + \varphi(b-a)$, where $\varphi = \frac{\sqrt{5}-1}{2}$ (the golden ratio).
2. If $f(x_L) > f(x_P)$, then $a := x_L$. Otherwise set $b := x_P$.
3. If $b - a <$ `eps`, then you're done.

If the method didn't converge in `maxiter` iterations, generate a warning with the `warning()` function.

The function should return a named list with the following components (cf. `?optim`):

1. `par` – the approximate location of a local minimum: $(a+b)/2$,
2. `value` – value of `f()` at the minimum,
3. `counts` – number of iterations considered, i,
4. `convergence` – `0` if the method converged in no more than `maxiter` iterations and `1` otherwise,
5. `message` – always equal to `NULL`.

Exercise 02.03. Write a function `gendiscrete()` with the following parameters:

- a single natural number `n`;
- a numeric vector `x` of length k (for some k) with unique elements,
- a numeric vector `p` of length k such that $p_i \ge 0$ and $\sum_{i=1}^{k} p_i = 1$. If the elements in `p` don't sum up to 1, then generate a warning and normalize it yourself.

The function should generate a sequence consisting of randomly chosen elements in `x` (with replacement) such that $\Pr(\mathrm{SELECT}_j = \mathtt{x}_i) = \mathtt{p}_i$ $(\forall i = 1, \ldots, k, j = 1, \ldots, \mathtt{n})$.

One of the simplest algorithms to generate a single observation from the above distribution is as follows:

1. Generate a random observation u from the uniform distribution on $(0,1)$, see `?runif`.
2. Find $m \in \{1, \ldots, k\}$ such that $u \in (\sum_{j=1}^{m-1} \mathtt{p}_j, \sum_{j=1}^{m} \mathtt{p}_j]$, under the assumption that $\sum_{j=1}^{0} \cdot = 0$.
3. Return $\mathtt{x}_m$ as result.

HUMAN CAPITAL
NATIONAL COHESION STRATEGY
EUROPEAN UNION
EUROPEAN SOCIAL FUND
Project co-financed by the European Union under the European Social Fund

Exercise 02.04. Write a function `unwind()` to transform a given $n \times m$ matrix (with the `dimnames` attribute set) into a data frame with nm rows and three columns. The column names of the resulting data frame are given via the 2nd function's parameter.

All the elements of the input matrix should be included in the first data frame's column. The 2nd and 3rd column should consist in a proper combination of the input matrix's dim names.

For example, let's consider the built-in `WorldPhones` matrix. This is the expected result of a call to `unwind(WorldPhones, c("count", "where", "when"))`:

```
   count    where  when
...
2  60423   N.Amer  1956
3  64721   N.Amer  1957
...
9  29990   Europe  1956
10 32510   Europe  1957
...
```

Exercise 02.05. We say that a directed graph is *immoral*, if there exist 2 vertices not joined (directly) with any edge but having a common child. Write a function `inquisition()` to determine if a given graph (represented by a square 0-1 matrix) is immoral.

Exercise 02.06.** Write a function declared as `comb(k, file="")`, `k` $\in \mathbb{N}$, that asks a user to input a set of positive numeric values (call `scan(file, what=double(), n=1, quiet=TRUE)` to get a single value). You don't know in advance what is the number of elements the user wishes to input (assume that a value ≤ 0 denotes end of input).

Return a random sequence (without replacement) consisting of only $\min\{n, \mathtt{k}\}$ of the provided elements, where n is the total number of elements typed in by the user. Use only $O(\mathtt{k})$ space (e.g. 1-3 auxiliary vectors of size `k` and some vectors of length 1; in other words, you may not refer to all the n values provided at the same time, the selection process must be performed iteratively). Of course, by "random" we mean a situation in which all the resulting sequences appear with the same probability (a uniform distribution). Example:

```r
print(comb(2)) # file="" => default => keyboard input
## 1: 3 <USER INPUT>
## 1: 4 <USER INPUT>
## 1: 1 <USER INPUT>
## 1: 2 <USER INPUT>
## 1: 0 <USER INPUT => THIS IS THE END>
## [1] 3 1
```

For testing purposes, you may read input values from a `textConnection`:

```r
test <- c("1", "2", "3", "4", "5", "6", "7", "0")
test_connection <- textConnection(test)
print(comb(2, test_connection))
## [1] 5 2
close(test_connection)
```

HUMAN CAPITAL
NATIONAL COHESION STRATEGY
EUROPEAN UNION
EUROPEAN SOCIAL FUND
Project co-financed by the European Union under the European Social Fund