Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №7 по курсу "Функциональное и логическое программирование"

Тема Использование управляющих структур, модификация списков

Студент Пересторонин П.Г.

Группа ИУ7-63Б

Оценка _____

Преподаватель Толпинская Н. Б.

Москва — 2021 г.

Оглавление

1 Задания

1.1 Написать функцию, которая по своему аргументу-списку `lst` определяет, является ли он полиндромом (то есть равны ли `lst` и `(reverse lst)`)

```
(defun polyndromp (lst)
  (equal lst (reverse lst)))
```

1.2 Написать предикат `set-equal`, который возвращает `t`, если два его множества-аргумента содержат одни и те же элементы, порядок которых не имеет значения

```
(defun set-equal (lst1 lst2)
  (and (subsetp lst2 lst1) (subsetp lst1 lst2)))
```

1.3 Напишите необходимые функции, которые обрабатывают таблицу из точечных пар: `(страна . столица)`, и возвращают по стране столицу, а по столице — страну

```
(defun get-cptl (cntry cntry-cptl)
  (let ((pair (assoc cntry cntry-cptl)))
```

```
    (and pair (cdr pair))))

(defun get-cntry (cptl cntry-cptl)
  (let ((pair (rassoc cptl cntry-cptl)))
    (and pair (car pair))))
```

1.4 Напишите функцию `swap-first-last`, которая переставляет в списке аргументе первый и последний элементы

1.4.1 Разрушающая структуру

```
(defun nswap-first-last (lst)
  (let ((el1 (car lst))
        (last-el (last lst)))
    (setf (car lst) (car last-el))
    (setf (car last-el) el1)
    lst))
```

1.4.2 Не разрушающая структуру

```
(defun swap-first-last (lst)
  (let ((el1 (car lst))
        (last-el (car (last lst))))
    (reverse (cons el1
          (cdr
            (reverse
              (cons last-el (cdr lst))))))))
```

1.5 Напишите функцию `swap-two-ellement`, которая переставляет в списке-аргументе два указанных своими порядковыми номерами элемента в этом списке

1.6 Разрушающая структуру

```
(defun nswap-two-ellement (n1 n2 lst)
  (let ((len (length lst)))
    (and (< n1 len) (< n2 len)
         (let ((el1 (nth n1 lst))
               (el2 (nth n2 lst)))
           (setf (nth n1 lst) el2)
           (setf (nth n2 lst) el1)
           lst))))
```

1.7 Не разрушающая структуру

```
(defun swap-two-ellement (n1 n2 lst)
  (let ((len (length lst))
        (lst-copy (copy-list lst)))
    (and (< n1 len) (< n2 len)
         (let ((el1 (nth n1 lst))
               (el2 (nth n2 lst)))
           (setf (nth n1 lst-copy) el2)
           (setf (nth n2 lst-copy) el1)
           lst-copy))))
```

1.8 Напишите две функции, `swap-to-left` и `swap-to-right`, которые производят круговую перестановку в списке-аргументе влево и вправо, соответственно

```
(defun swap-to-left (lst)
  (and lst
       (let ((tail (cdr lst))
             (head (car lst)))
         (reverse (cons head (reverse tail))))))

(defun swap-to-right (lst)
  (and lst
       (let ((last-el (car (last lst))))
         (reverse (cdr (reverse (cons last-el lst)))))))
```

2 Ответы на вопросы к лабораторной работе

2.1 Способы определения функций

2.2 Варианты и методы модификации списков