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»

Отчёт по лабораторной работе №3 по курсу «Функциональное и логическое программирование»

Тема Работа интерпретатора Lisp

Студент Богаченко А. Е.

Группа ИУ7-65Б

Оценка (баллы) _____

Преподаватели Строганов Ю. В., Толпинская Н. Б.

Москва — 2022 г.

Задание 1

Написать функцию, которая принимает целое число и возвращает первое четное число, не меньшее аргумента.

Листинг 1 – Задание 1

```
(defun make-first-even (num)
  (if (evenp num)
      num
      (+ num 1)))
```

Задание 2

Написать функцию, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента.

Листинг 2 – Задание 2

```
(defun make-abs-greater-num (num)
  (if (> num 0)
      (+ num 1)
      (- num 1)))
```

Задание 3

Написать функцию, которая принимает два числа и возвращает список из этих чисел, расположенный по возрастанию.

Листинг 3 – Задание 3

```
(defun make-asc-pair (a b)
  (list (min a b) (max a b)))
```

Задание 4

Написать функцию, которая принимает три числа и возвращает T только тогда, когда первое число расположено между вторым и третьим.

Листинг 4 – Задание 4

```
(defun is-between (a b c)
  (if (and (> a b) (< a c))
      T
      Nil))
```

Задание 5

Каков результат вычисления следующих выражений?

Листинг 5 – Задание 5

```
  (and 'fee 'fie 'foe) ; FOE
  (or 'fee 'fie 'foe) ; FEE
  (and (equal 'abc 'abc) 'yes) ; YES
  (or nil 'fie 'foe) ; FIE
  (and nil 'fie 'foe) ; NIL
  (or (equal 'abc 'abc) 'yes) ; T
```

Задание 6

Написать предикат, который принимает два числа аргумента и возвращает T, если первое число не меньше второго.

Листинг 6 – Задание 6

```
(defun gep (a b)
  (if (>= a b)
      T
      Nil))
```

Задание 7

Какой из следующих двух вариантов предиката ошибочен и почему?

Листинг 7 – mystery

```
(defun pred1 (x)
  (and (numberp x) (plusp x))) ; Good

(defun pred2 (x)
  (and (plusp x) (numberp x))) ; Bad
```

Второй предикат ошибочен, потому что в случае, если на вход функции будет подан аргумент, не являющийся числом, к нему будет применена функция `plusp`, определяющая, является ли число положительным. Она определена только для чисел, поэтому в случае нечислового аргумента интерпретатор выдаст ошибку.

Задание 8

Решить задачу 4, используя для ее решения конструкции `IF`, `COND`, `AND/OR`.

Листинг 8 – Задание 8

```
;; IF variant
(defun is-between-if (a b c)
  (if (and (> a b) (< a c))
      T
      Nil))

;; COND variant
(defun is-between-cond (a b c)
  (cond
    ((and (> a b) (< a c)) T)
    (T Nil)))

;; AND/OR variant
(defun is-between-andor (a b c)
  (and (> a b) (< a c)))
```

Задание 9

Переписать функцию `how-alike`, приведенную в лекции и использующую `COND`, используя конструкции `IF, AND/OR`.

Листинг 9 – Задание 9

```
;; COND variant
(defun how-alike-cond (x y)
  (cond
    ((or (= x y) (equal x y)) 'the_same)
    ((and (oddp x) (oddp y)) 'both_odd)
    ((and (evenp x) (evenp y)) 'both_even)
    (T 'difference)))

;; IF variant
(defun how-alike-if (x y)
  (if (if (= x y)
          (equal x y))
      'the_same
      (if (if (oddp x)
              (oddp y))
          'both_odd
          (if (if (evenp x)
                  (evenp y))
              'both_even
              'difference))))

;; AND/OR variant
(defun how-alike-andor (x y)
  (or (and (or (= x y) (equal x y)) 'the_same)
  (and (oddp x) (oddp y) 'both_odd)
  (and (evenp x) (evenp y) 'both_even)
  'difference))
```

Контрольные вопросы

1. Базис языка `Lisp`

Базис языка представлен:

- структурами, атомами;
- функциями:
  `atom, eq, cons, car, cdr,`
  `cond, quote, lambda, eval, label`.

2. Классификация функций языка `Lisp`

Функции в языке `Lisp`:

- чистые (с фиксированным количеством аргументов) – математические функции;
- рекурсивные функции;
- специальные функции – формы (принимают произвольное количество аргументов или по разному обрабатывают аргументы);
- псевдофункции (создающие «эффект» - отображающие на экране процесс обработки данных и т. п.);
- функции с вариативными значениями, выбирающие одно значение;
- функции высших порядков – функционалы (используются для построения синтаксически управляемых программ).

По назначению функции разделяются следующим образом:

1. конструкторы — создают значение (`cons`, например);
2. селекторы — получают доступ по адресу (`car`, `cdr`);
3. предикаты — возвращают `Nil`, `T`.

3. Способы создания функций

С помощью макро определения `defun` или с использованием Лямбда-нотации (функция без имени).

Работа функций `and`, `or`, `if`, `cond`

Функция `and`

Синтаксис: `(and expression-1 expression-2 ... expression-n)`

Функция возвращает первое `expression`, результат вычисления которого = `Nil`. Если все не `Nil`, то возвращается результат вычисления последнего выражения.

Примеры:

```
(and 1 Nil 2) -> Nil
(and 1 2 3) -> 3
```

Функция `or`

Синтаксис: `(or expression-1 expression-2 ... expression-n)`

Функция возвращает первое `expression`, результат вычисления которого не `Nil`. Если все `Nil`, то возвращается `Nil`.

Примеры:

```
(or Nil Nil 2) -> 2
(or 1 2 3) -> 1
```

Функция `if`

Синтаксис: `(if condition t-expression f-expression)`

Если вычисленный предикат не `Nil`, то выполняется `t-expression`, иначе - `f-expression`.

Примеры:

```
(if Nil 2 3) -> 3
(if 0 2 3) -> 2
```

Функция `cond`

Синтаксис:

```
(cond
  (condition-1 expression-1)
  (condition-2 expression-2)
  ...
  (condition-n expression-n))
```

По порядку вычисляются и проверяются на равенство с `Nil` предикаты. Для первого предиката, который не равен `Nil`, вычисляется находящееся с ним в списке выражение и возвращается его значение. Если все предикаты вернут `Nil`, то и `cond` вернет `Nil`.

Примеры:

```
(cond (Nil 1) (2 3)) -> 3
(cond (Nil 1) (Nil 2)) -> Nil
```