

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра прикладной математики и кибернетики

Лабораторная работа № 1
по дисциплине «Функциональное и логическое программирование»

Бригада № 5

Выполнили:

студенты группы ИП-811

Макаров Д.А

Разумов Д.Б

Работу проверил: Галкина М.Ю

Новосибирск 2020 г.

Содержание

Задание	3
Листинг программы	4
Скриншоты работы программы.....	5

Задание

1. Напишите сложную функцию, используя композиции функций CAR и CDR, которая возвращает атом * при применении к следующему списку:
(1 (2 (3 4) *) 5)

2. Объясните работу функций и определите результат обращения:
(cons '(a (b c)) nil)

3. Из атомов 1, 2, 3, nil создайте указанные списки двумя способами:
а) с помощью композиций функций CONS;
б) с помощью композиций функций LIST.
((1)(2 3))

4. С помощью DEFUN определите функцию, которая возвращает измененный список по заданию (в теле функции разрешается использовать только следующие встроенные функции: CAR, CDR, CONS, APPEND, LIST, LAST, BUTLAST с одним аргументом). Проверьте её работу, организовав обращение к функции со списками разной длины.

Функция меняет местами третий и последний элементы списка

Листинг программы

Задание 1

```
(caddr(cadr'(1 (2 (3 4) *) 5)))
```

Задание 2

```
(cons '(a (b c)) nil)
```

Задание 3 а

```
(cons '(1) (cons '(2 3) nil))
```

Задание 3 б

```
(list '(1) '(2 3))
```

Задание 4

```
(defun repl (lis)
```

```
  (append (cons(car lis) nil) (cons (cadr lis) nil) (last lis)
```

```
    (butlast (cdddr lis)) (cons(caddr lis) nil))
```

```
)
```

```
(set 'li '(1 2 3 4 5 6 7 (8 9 10)))
```

```
(repl li)
```

Скриншоты работы программы

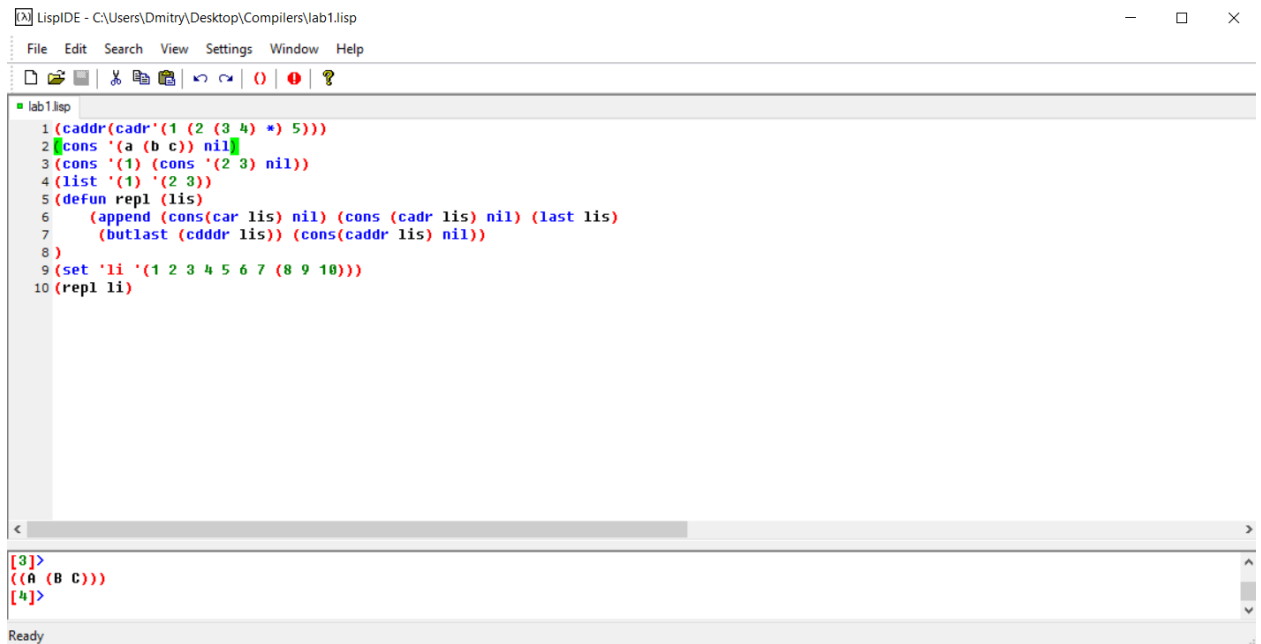
Рис.1 – выполнение задания 1



```
LispIDE - C:\Users\Dmitry\Desktop\Compilers\lab1.lisp
File Edit Search View Settings Window Help
lab1.lisp
1 (caddr(cadr'(1 (2 (3 4) *) 5)))
2 (cons '(a (b c)) nil)
3 (cons '(1) (cons '(2 3) nil))
4 (list '(1) '(2 3))
5 (defun repl (lis)
6   (append (cons(car lis) nil) (cons (cadr lis) nil) (last lis)
7   (butlast (caddr lis)) (cons(caddr lis) nil))
8 )
9 (set 'li '(1 2 3 4 5 6 7 (8 9 10)))
10 (repl li)

[2]>
*
[3]>
Ready
```

Рис.2 – выполнение задания 2



```
LispIDE - C:\Users\Dmitry\Desktop\Compilers\lab1.lisp
File Edit Search View Settings Window Help
lab1.lisp
1 (caddr(cadr'(1 (2 (3 4) *) 5)))
2 (cons '(a (b c)) nil)
3 (cons '(1) (cons '(2 3) nil))
4 (list '(1) '(2 3))
5 (defun repl (lis)
6   (append (cons(car lis) nil) (cons (cadr lis) nil) (last lis)
7   (butlast (caddr lis)) (cons(caddr lis) nil))
8 )
9 (set 'li '(1 2 3 4 5 6 7 (8 9 10)))
10 (repl li)

[3]>
((A (B C)))
[4]>
Ready
```

Объяснение работы функции: функция **cons** берет элемент, записанный в качестве первого аргумента и список, записанный в качестве второго аргумента, и возвращает список с добавленным элементом в начало списка.

В данном случае в пустой список (**nil**) добавляется элемент (a (b c))

Рис. 3 – выполнение задания 3 (а)




```
LispIDE - C:\Users\Dmitry\Desktop\Compilers\lab1.lisp
File Edit Search View Settings Window Help

lab1.lisp
1 (caddr(cadr'(1 (2 (3 4) *) 5)))
2 (cons '(a (b c)) nil)
3 (cons '(1) (cons '(2 3) nil))
4 (list '(1) '(2 3))
5 (defun repl (lis)
6   (append (cons(car lis) nil) (cons (cadr lis) nil) (last lis)
7   (butlast (caddr lis)) (cons(caddr lis) nil))
8 )
9 (set 'li '(1 2 3 4 5 6 7 (8 9 10)))
10 (repl li)

[4]>
((1) (2 3))
[5]>

Ready
```

Рис. 4 – выполнение задания 3 (б)



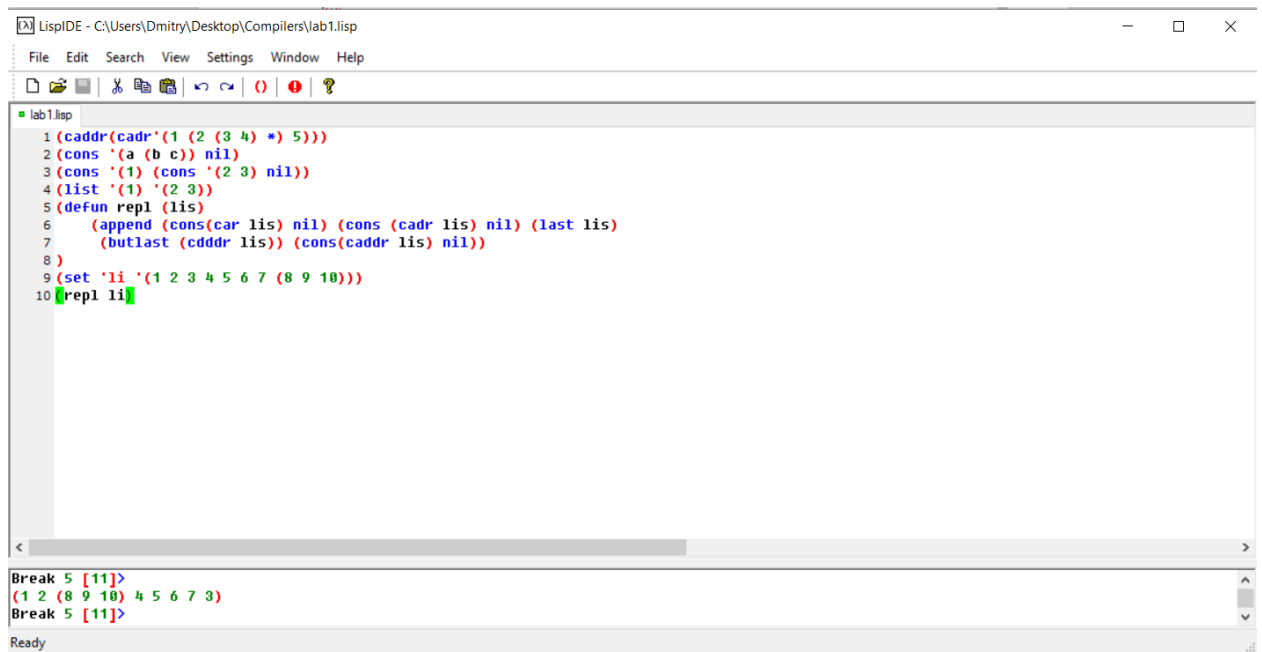
```
LispIDE - C:\Users\Dmitry\Desktop\Compilers\lab1.lisp
File Edit Search View Settings Window Help

lab1.lisp
1 (caddr(cadr'(1 (2 (3 4) *) 5)))
2 (cons '(a (b c)) nil)
3 (cons '(1) (cons '(2 3) nil))
4 (list '(1) '(2 3))
5 (defun repl (lis)
6   (append (cons(car lis) nil) (cons (cadr lis) nil) (last lis)
7   (butlast (caddr lis)) (cons(caddr lis) nil))
8 )
9 (set 'li '(1 2 3 4 5 6 7 (8 9 10)))
10 (repl li)

[5]>
((1) (2 3))
[6]>

Ready
```

Рис. 5 – выполнение задания 4



The screenshot shows the LispIDE window with the following content:

File Edit Search View Settings Window Help

lab1.lisp

```
1 (caddr(cadr'(1 (2 (3 4) *) 5)))  
2 (cons '(a (b c)) nil)  
3 (cons '(1) (cons '(2 3) nil))  
4 (list '(1) '(2 3))  
5 (defun repl (lis)  
6   (append (cons(car lis) nil) (cons (cadr lis) nil) (last lis)  
7   (butlast (caddr lis)) (cons(caddr lis) nil))  
8 )  
9 (set 'li '(1 2 3 4 5 6 7 (8 9 10)))  
10 (repl li)
```

Break 5 [11]>
(1 2 (8 9 10) 4 5 6 7 3)
Break 5 [11]>

Ready