

[Личный кабинет](#) / [Мои курсы](#) / [ВвФП](#) / [Итоговая контрольная работа \(дд/IV/2025\)](#) / [Демовариант к/р весна 2025 г.](#)**Тест начат** воскресенье, 13 апреля 2025, 20:23**Состояние** Завершены**Завершен** воскресенье, 13 апреля 2025, 23:13**Прошло  
времени** 2 час. 50 мин.**Оценка** Еще не оцененоВопрос **1**

Выполнен

Баллов: 0,00 из 0,50

Положим, что пользователь определил функцию (**define (quintduplicate x) (+ x (+ x x) (+ x x))**), которая находит упятерённое значение своего параметра **x** суммированием. Укажите, сколько вызовов функции **+** потребуется выполнить при нормальном порядке вычислений (точнее, том порядке, который назван "нормальным" в учебнике), чтобы узнать значение выражения (**quintduplicate (quintduplicate 2)**)? Вызов функции **+** учитывается однократно независимо от количества аргументов-слагаемых.

- ☐ a. 6
- ☐ b. 21
- ☐ c. 18
- ☐ d. количество суммирований будет кратно 3
- ☐ e. 19
- ☐ f. больше 21
- ☐ g. 20
- ☐ h. количество суммирований будет чётным
- ☐ i. 3
- ☐ j. 9
- ☒ k. 12
- ☐ l. 15



Вопрос **2**

Выполнен

Баллов: 0,50 из 0,50

Укажите всё то, что верно описывает вычисление выражения `(if (or) (delay (/ 1 0)) (delay (quotient 7 2)))`, где `delay` -- спецформа, описанная в лекции 8.

- ☐ a. Результатом будет `#<promise>` -- поделить на 0. Это так, поскольку результат `(or)` -- `#t` и сработает первая ветка `if`.
- ☒ b. Результатом будет `#<promise>` -- обещание найти частное при делении нацело 7 на 2. Это так, поскольку результат `(or)` -- `#f` и сработает вторая ветка `if`.
- ☐ c. Результатом будет ошибка. Спецформу `delay` нельзя использовать, записывая аргументы вызова спецформы `if`.
- ☐ d. Результатом будет ошибка из-за деления на 0. `if` вынудит вычислить все свои аргументы. При вычислении второго аргумента (если нумеровать с 1) произойдёт деление на 0.

Вопрос **3**

Выполнен

Баллов: 0,50 из 0,50

Какое подвыражение вычисляется первым при вычислении выражения `(f a b)` при выполнении *согласно стандарту языка*?

- ☐ a. `f`
- ☐ b. `b`
- ☐ c. `a`
- ☒ d. неизвестно, так как по стандарту порядок вычисления подвыражений не фиксирован и определяется в реализации интерпретатора

Вопрос **4**

Выполнен

Баллов: 0,50 из 0,50

Укажите среди перечисленного то и только то, что справедливо для рекурсивных функций.

- ☒ a. вызов рекурсивной функции может породить рекурсивный процесс
- ☐ b. вызов рекурсивной функции не может породить итеративный процесс
- ☐ c. вызовы рекурсивных функций порождают только рекурсивные процессы
- ☐ d. отложенные вычисления возникают при вычислении вызовов рекурсивных функций только в случае хвостовой рекурсии
- ☐ e. вызов рекурсивной функции не может породить рекурсивный процесс
- ☐ f. отложенные вычисления возникают всегда при вычислении вызовов рекурсивных функций
- ☒ g. вызов рекурсивной функции может породить итеративный процесс



Вопрос **5**

Выполнен

Баллов: 0,00 из 0,50

Укажите среди перечисленного то и только то, что справедливо для специальной формы **cond**.

- ☐ a. в **cond** всегда должны быть либо две альтернативы, либо более чем две альтернативы
- ☐ b. в **cond** предикат из очередной альтернативы не вычисляется только тогда, когда хотя бы один из предшествующих ему был равен **#t**
- ☒ c. в **cond** предикат из очередной альтернативы вычисляется только тогда, когда либо он первый, либо все предшествующие ему были ложны
- ☒ d. в **cond** всегда должна быть либо одна альтернатива, либо более чем одна альтернатива
- ☐ e. если в **cond** истинны предикаты в нескольких альтернативах, то выполняемая альтернатива выбирается из них случайно
- ☐ f. в **cond** всегда должна быть **else** альтернатива



Вопрос **6**

Выполнен

Балл: 10,00

Составьте код решений задач по своему выбору и отправьте как TXT-файл. Так как проверка осуществляется вручную, то даётся лишь одна попытка. Удачное решение принесёт до 10 баллов (максимум по 2 балла за каждую задачу). Эффективность решений по времени счёта и памяти учитывается. Баллы за неэффективные решения снижаются.

**I.** Дополните предложенные описания функции **filter**, фильтрующей список. **foldl** и **foldr** – лево- и право- ассоциативные свёртки. Дополняя описания, нельзя пользоваться спецформами **define**, **let**, а также функциями **reverse**, **lambda**, **foldr**, **foldl**, **map** и т. п. функциями высших порядков обработки списков.

**(define (filter1 f lst)**

**(reverse (foldl (lambda (x y)**

**null lst)))**

**(define (filter2 f lst)**

**(foldr (lambda (x y)**

**null lst))**

**II.** Квадродеревом называется следующий способ представления растровых черно-белых изображений:

1) Если изображение целиком белое, то оно представляется квадродеревом из одной «белой» вершины. Значение такого квадродерева: 0 [ноль].

2) Если изображение целиком чёрное, то оно представляется квадродеревом из одной «чёрной» вершины. Значение такого квадродерева: 1[единица].

3) Если на изображении есть и чёрные и белые участки, то оно делится на 4 равные части (верхнюю левую, верхнюю правую, нижнюю левую, нижнюю правую) и представляется квадродеревом, состоящим из корневой вершины и четырёх поддеревьев, которые описывают части изображения. Обозначим значения поддеревьев: <верхлевд>, <верхправд>, <нижнлевд>, <нижнправд>; тогда значением всего дерева будет вектор из четырёх элементов, т. е. результат вычисления **(vector <верхлевд> <верхправд> <нижнлевд> <нижнправд>)**.

Пример квадродерева: **#(1 0 0 #(1 1 1 0))**

Составьте функцию **(taskII t s)**, которая находит суммарную площадь чёрных участков изображения, считая, что площадь всего участка, описываемого квадродеревом **t**, равна **s**. Результат для дерева из примера равен 7, если площадь квадрата равна 16.

**III.** Реализуйте функцию **(taskIII-cps t s cc)**, являющуюся версией **(taskII t s)**, которая составлена в стиле передачи продолжений.

**IV.** Опишите бесконечный поток **nonfactorials**, составленный из всех натуральных чисел, не являющихся факториалами. Числа в потоке упорядочены по возрастанию. Начало **nonfactorials**: 3 4 5 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 ...

**Указание:** Принимается только решение через порождающую функцию. Решения через неявное определение потока не будут оценены на полный балл.

**V.** Опишите мемоизированную функцию **(taskV n)**, находящую n-ое число в последовательности простых чисел Мерсенна: 3, 7, 31, 127, 8191, 131071, 524287, 2147483647, 2305843009213693951, 618970019642690137449562111, 162259276829213363391578010288127, ... . Простое число Мерсенна -- это простое число, представимое как  $2^k - 1$ , т. е. какая-то степень двойки ( $k$  -- натуральное), уменьшенная на 1. В решении можете использовать сведения со страницы <https://oeis.org/A000668>. В комментарии к коду сообщите своё мнение о том, при каких условиях мемоизация будет уместна.

 [solution\\_6.txt](#)



## Вопрос 7

Выполнен

Баллов: 0,50 из 0,50

Укажите среди предложенного те и только те выражения, которые являются *литералами*.

- ☐ a. `(/ 2 3)`
- ☐ b. `abc`
- ☐ c. `(a b c)`
- ☒ d. `'(/ 2 3)`
- ☐ e. `('a 'b 'c)`
- ☒ f. `'abc`
- ☒ g. `(quote (/ 2 3))`
- ☒ h. `"abc"`
- ☒ i. `2/3`
- ☐ j. `(a 'b 'c)`
- ☐ k. `+`
- ☒ l. `#t`
- ☐ m. `(+ 1 (+ 2 3))`

## Вопрос 8

Выполнен

Баллов: 0,00 из 0,50

Укажите те и только те утверждения, которые верны.

- ☒ a. Функция `add1` является "синтаксическим сахаром", так как может быть заменена использованием функции `+`.
- ☐ b. Структура данных вектор является "синтаксическим сахаром", так как может быть заменена использованием списков. Программы со списками могут быть не такими эффективными, как их версии с использованием векторов, но это не имеет значения при решении вопроса о том, являются ли вектора "синтаксическим сахаром".
- ☐ c. Спецформа `case` является "синтаксическим сахаром", так как может быть заменена функцией, реализующей точно такое же вычисление, что и `case`.
- ☒ d. Структура данных вектор не является "синтаксическим сахаром", так как синтаксические конструкции, введённые для её поддержки (в частности, запись внешнего представления векторов), уникальны.



Вопрос **9**

Выполнен

Балл: 5,00

Составьте на листе тексты ответов и нарисуйте стрелочную диаграмму. Сфотографируйте и загрузите имидж или pdf-версию. Удачное решение принесёт до 5 баллов. Составляемые Вами примеры не должны совпадать или мало отличаться от примеров из книг, со слайдов лекций. Так как проверка производится вручную, то предоставляется лишь одна попытка.

**I. (до 2 баллов)** Расскажите об остаточных вычислениях. Приведите пример выражения и остаточных вычислений в этом выражении относительно выбранного Вами подвыражения. Расскажите об описании функций в стиле передачи остаточных вычислений (cps -- continuation-passing style). Приведите осмысленный пример cps-описания нетривиальной функции. Какие возможности предоставляют cps-описания функций по сравнению с обычными? (Примером может быть код из Ваших решений задач демоварианта, если Вы там использовали cps.)

**II. (до 1 балла)** Пусть `mcons`, `mcar`, `mcdrr`, `set-mcdr!`, `set-mcdr!` – конструкторы, селекторы и мутаторы мутируемых списков и мутируемых пар. Пусть в программе вычислено:

```
> (define a (mcons 1 (mcons 2 (mcons 3))))  
> (define b (mcons 4 '(a) a))  
> (define c (mcdrr a))  
> (set-mcdr! c (mcdrr b))
```

Нарисуйте стрелочную диаграмму, изображающую `a`, `b` и `c` после выполненных вычислений.

**III. (до 2 баллов)** Расскажите о мемоизации, в чём заключается этот программистский приём? В каких случаях оправдано использование мемоизации? В каких случаях использование мемоизации не оправдано? Ответьте и обоснуйте свой ответ на вопрос указанием осмысленных примеров кода. (Это могут быть примеры кода из Ваших решений задач демоварианта, если Вы там использовали или не использовали мемоизацию.)

[solution\\_9.txt](#)Вопрос **10**

Выполнен

Баллов: 0,50 из 0,50

Укажите те и только те утверждения о модели вычислений с окружениями, которые верны.

- ☒ a. Для вычисления вызовов некоторых видов функций в модели вычислений с окружениями не требуется создавать новое окружение, предназначенное для вычисления вызова.
- ☐ b. При вычислении определения функции создаётся пара со ссылкой на её тело и список параметров и ссылкой на кадр глобального окружения.
- ☐ c. При вычислении вызова функции, определённой программистом в коде, создаётся пара со ссылкой на её тело и список параметров и ссылкой на кадр окружения, в котором происходит вычисление.
- ☒ d. При вычислении определения функции создаётся пара со ссылкой на её тело и список параметров и ссылкой на кадр окружения, в котором произошло вычисление определения функции.
- ☒ e. При вычислении вызова функции, определённой программистом в коде, создаётся новый кадр окружения, хранящий связывания всех параметров функции и используемый при вычислении тела функции.



Вопрос **11**

Выполнен

Баллов: 0,00 из 0,50

Пусть даны определения:

```
(define f symbol?)  
(define a (list 1 2 f))  
(define b '(1 2 f))  
(define c (caddr a))  
(define d (caddr b))
```

Укажите те и только те утверждения, которые верны. Указание: При составлении ответа Вам может помочь стрелочная диаграмма.

- ☒ a. все элементы списка **b** являются числами или символами.
- ☐ b. все элементы списка **a** являются числами или символами.
- ☐ c. **c** и **d** совпадают (являются одним и тем же объектом).
- ☐ d. **c** и **d** одинаково выглядят, но не совпадают (не являются одним и тем же объектом).
- ☒ e. `(eq? (c 'c) (c d))` даёт **#t**, так как совпадают результаты вычисления выражений '**c** и **d**.
- ☐ f. **a** и **b** совпадают (являются одним и тем же объектом).
- ☒ g. `(eq? (c d) (c 'd))` даёт **#t** из-за того, что значение **d** является символом, **c** имеет то же значение, что и **f**, а функция **f** определена как синоним функции **symbol?**.
- ☐ h. **a** и **b** одинаково выглядят, но не совпадают (не являются одним и тем же объектом).

Вопрос **12**

Выполнен

Баллов: 0,50 из 0,50

Реализованное на языке Scheme эффективное итеративное вычисление списка, составленного из факториалов первых **N** натуральных чисел, идущих по убыванию, требует... (укажите лучшую оценку из подходящих)

- ☐ a. квадратичной относительно **N** памяти
- ☐ b. экспоненциальной относительно **N** памяти
- ☐ c. экспоненциального относительно **N** времени счёта
- ☐ d. константного времени счёта, не зависящего от **N**
- ☒ e. линейной относительно **N** памяти
- ☐ f. константной памяти, не зависящей от **N**
- ☐ g. квадратичного относительно **N** времени счёта
- ☐ h. логарифмического относительно **N** времени счёта
- ☒ i. линейного относительно **N** времени счёта

[◀ Видео по восьмой теме \(2021\)](#)[Демовариант ноябрь-декабрь 2024 ▶](#)