

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

Задание.

Написать функцию (или набор функций), которая получает на вход ориентированный граф в виде списка смежных вершин и выводит все имеющиеся в нем циклы без повторов. Реализовать в clisp.

Решение.

```
(defun lastElement (lis)
  "Последний элемент списка"
  (cond ((cdr lis) (lastElement (cdr lis)))
        (t (car lis))))

(defun appendList (lis element)
  "Добавление элемента в конец списка"
  (cond (lis (cons (car lis) (appendList (cdr lis) element)))
        (t (cons element nil))))

(defun shiftCycle (cycle)
  " Для цикла (a1 a2 ... an) преобразование в
  (a2 a3 ... an a1)"
  (appendList (cdr cycle) (car cycle)))

(defun cycleEq0 (cycle1 cycle2 len)
  " Проверка эквивалентности
  цикла cycle1 циклу cycle2 с len
  возможными преобразованиями shiftCycle для
  цикла cycle1"
  (cond ((eql len 0) nil)
        ((equal cycle1 cycle2) t)
```

©МатБюро – Консультации по математике, программированию, экономике, праву, естественным наукам

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

```
(t (cycleEq0 (shiftCycle cycle1) cycle2 (- len 1))))

(defun cycleEq ( cycle1 cycle2 )
  " Проверка эквивалентности циклов
  (a1 ... an) и (b1 ... bm) "
  (cycleEq0 cycle1 cycle2 (length cycle1)))

(defun isMember (lis element)
  " Встречается ли element в списке lis "
  (cond ((equal lis nil) nil)
        ((equal (car lis) element) t)
        (t (isMember (cdr lis) element))))

(defun isCycleMember (lis cycle)
  " Встречается ли эквивалентный циклу cycle
  цикл в списке циклов lis"
  (cond ((equal lis nil) nil)
        ((cycleEq (car lis) cycle) t)
        (t (isCycleMember (cdr lis) cycle))))

(defun mergeLists (lis1 lis2)
  " Объединение двух списков
  (a1 ... an) (b1 ... bm) в (a1 ... an b1 ... bm) "
  (cond (lis1 (cons (car lis1) (mergeLists (cdr lis1) lis2)))
        (t lis2)))

(defun getPairs (element lis)
```

©МатБюро – Консультации по математике, программированию, экономике, праву, естественным наукам

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

```
" Для элемента a и списка (b1 ... bn)
  получение списка ( (a b1) (a b2) ... (a bn) ) "
(cond (lis (cons (list element (car lis))
                (getPairs element (cdr lis))))
      (t nil)))
```

```
(defun edgesByAdj0 (edges adj)
  "Рекурсивный преобразователь списка смежности орграфа
  в список его ориентированных ребер"
  (cond (adj (edgesByAdj0 (mergeLists edges
                                     (getPairs (car (car adj))
                                             (car (cdr (car adj)))))
                          (cdr adj)))
        (t edges)))
```

```
(defun edgesByAdj( adj )
  " По списку смежности ориентированного графа вида
  ( (1 (2 3)) (2 (4 5)) (3 (4 1)) )
  получение списка ориентированных ребер графа
  ((1 2) (1 3) (2 4) (2 5) (3 4) (3 1)) "
  (edgesByAdj0 nil adj))
```

```
(defun cycle2SimpCycle (lis)
  " Преобразование цикла как списка его ориентированных
  ребер в список его вершин в порядке обхода
  (упрощенная запись цикла)
  ( (1 2) (2 3) (3 1) ) -> ( 1 2 3) "
```

©МатБюро – Консультации по математике, программированию, экономике, праву, естественным наукам

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

```
(let ((len (length lis))
      (cond ((= len 1) (list (car (car lis))))
            (t (cons (car (car lis)) (cycle2SimpCycle (cdr lis)))))))

(defun cycles2SimpCycles (lis)
  " Преобразование списка циклов в список
  упрощенных записей циклов"
  (cond (lis (cons (cycle2SimpCycle (car lis))
                  (cycles2SimpCycles (cdr lis))))
        (t nil)))

(defun isCycle (lis)
  " Является ли список ребер циклом,
  т. е. совпадают ли последняя вершина ребра
  с первой вершиной следующего за ней ребра и
  совпадает ли первая вершина первого ребра
  с последней вершиной последнего ребра"
  (cond ((equal (car (car lis)) (car (cdr (lastElement lis)))) t)
        (t nil)))

(defun canAppendEdge (sprout edge)
  " Возможно ли к списку ребер sprout добавить ребро edge
  (если ребра edge еще нет в списке ребер sprout и последняя вершина
  последнего ребра sprout совпадает с первой вершиной ребра edge) "
  (cond ((isMember sprout edge) nil)
        ((= (car (cdr (lastElement sprout))) (car edge)) t)
        (t nil)))
```

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

```
(defun appendEdges (sprout edges)

  "К списку ребер sprout добавляют (если это возможно)
  одно ребро из списка edges. Возврат списка полученных
  списков ребер"

  (cond ((equal edges nil) nil)

        ((canAppendEdge sprout (car edges))

         (cons (appendList sprout (car edges))

               (appendEdges sprout (cdr edges))))

        (t (appendEdges sprout (cdr edges)))))

(defun addCycles (cycles candidates)

  "Добавление к списку циклов тех элементов
  списка списков ребер candidates, которые
  являются циклами и их эквивалентов еще нет
  среди cycles. Возврат списка cycles с
  добавленными к нему элементами"

  (cond ((equal candidates nil) cycles)

        ((and (isCycle (car candidates))

               (not (isCycleMember cycles (car candidates))))

         (addCycles (cons (car candidates) cycles) (cdr candidates)))

        (t (addCycles cycles (cdr candidates))))

)

)

(defun edgesToSprouts (edges)

  "Преобразование списка ориентированных ребер графа
```

©МатБюро – Консультации по математике, программированию, экономике, праву, естественным наукам

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

В список ростков циклов:

```
( (1 2) (1 3) (2 5) ) -> ( ((1 2)) ((1 3)) ((2 5)) )"
```

```
(cond (edges (cons (list (car edges))
                    (edgesToSprouts (cdr edges))))
      (t nil)))
```

```
(defun filterCycles (sprouts)
```

```
"Отсев из списка ростков циклов sprouts всех
готовых циклов"
```

```
(cond ((equal sprouts nil) nil)
      ((isCycle (car sprouts)) (filterCycles (cdr sprouts)))
      (t (cons (car sprouts)
                (filterCycles (cdr sprouts))))
      (t nil)))
```

```
(defun growSprouts (cycles edges sprouts)
```

```
" Рекурсивная процедура роста ростков циклов sprouts в
орграфе с списком ребер edges с накоплением
полученных циклов в cycles"
```

```
(cond (sprouts
      (let ((newSprouts (appendEdges (car sprouts) edges)))
        (growSprouts (addCycles cycles newSprouts)
                     edges
                     (mergeLists (filterCycles newSprouts) (cdr
sprouts)))))
      (t cycles)))
```

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

```
(defun getSimpCycles (adj)
  "Получить по списку смежности вершин орграфа adj
  (например ((1 (2 3)) (2 (3 4)) (4 (1 2))) )
  список его циклов (без повторения ребер в каждом цикле)
  в упрощенной записи (список вершин цикла в порядке их обхода)
  (например ((1 2 3) (4 5)) )"
  (let ((edges (edgesByAdj adj)))
    (cycles2SimpCycles (growSprouts nil edges (edgesToSprouts edges)
      ) )))
```

Примеры использования.

```
(getSimpCycles nil)
(getSimpCycles '((1 (2 3))) )
(getSimpCycles '((1 (2 3)) (2 (4 3))) )
(getSimpCycles '((1 (2 3)) (2 (1 3))) )
(getSimpCycles '((1 (2)) (2 (3)) (3 (1 4)) (4 (2))) )
(getSimpCycles '((1 (2)) (2 (3)) (3 (1 4 6)) (4 (5)) (5 (3)) (6 (7))
(7 (3))) )
```