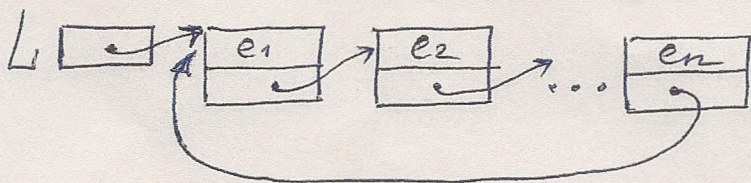


Кольцевой список.



Это односторонний список, в последнем звене которого вместо nil хранится ссылка на 1-е звено.

Прорешайте к экзамену все задачи из №16.41
Вот некоторые из них. задачи: (на стр. 163)

type список = ↑ звено; звено = record элем: integer; след: список end

1 задача Описать процедуру del(L), которая удаляет (с освобождением памяти) первый элемент из L, где L — непустой односторонний список.

Кольцевой

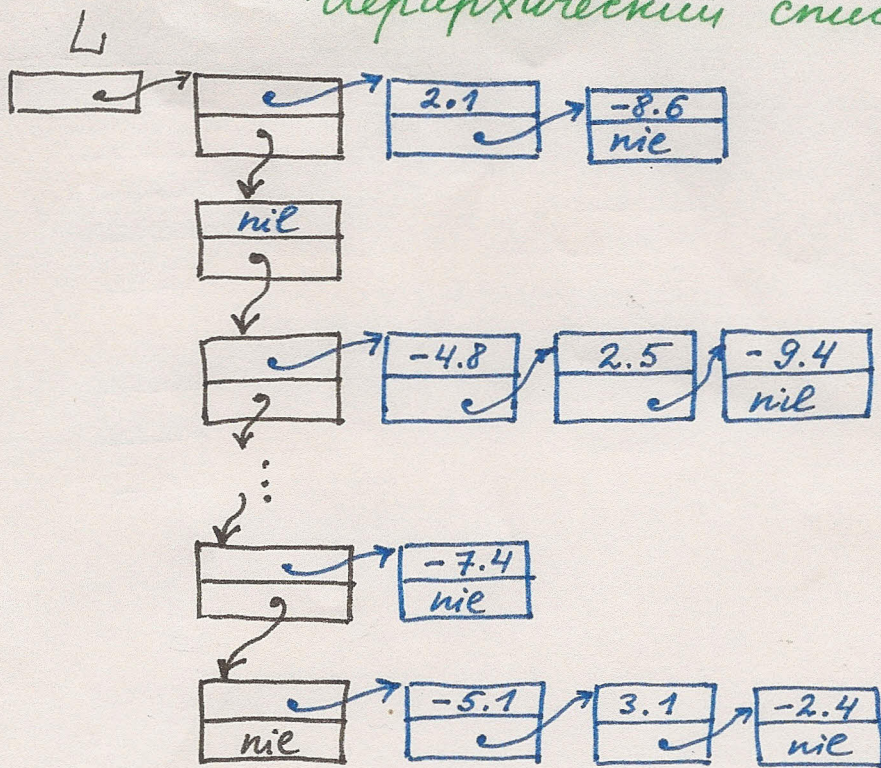
```
procedure del(var L: список);
var p, q: список;
begin
  if L↑.след = L then { в списке только одно звено }
  begin
    dispose(L); L := nil end
  else { в списке ≥ 2-х звеньев }
  begin
    { поиск последнего звена: }
    p := L; while p↑.след <> L do p := p↑.след;
    { p — указывает на последнее звено }
    q := L; { адрес текущего звена }
    L := L↑.след; dispose(q); { удал. 1-го звена }
    p↑.след := L { ссылка на новое 1-е звено }
  end;
end;
```

2 задача Описать проц. ins(L), которая вставляет новый элемент φ в начало L (одностороннего кольц. списка).

```
procedure ins(var L: список);
var p, q: список;
begin
  if L = nil then begin new(L); L↑.элем := 0; L↑.след := L end
  else
  begin
    { поиск последнего звена: }
    p := L; while p↑.след <> L do p := p↑.след;
    { p — указывает на последнее звено }
    new(q); q↑.элем := φ; q↑.след := L; L := q;
    p↑.след := L { ссылка на новое 1-е звено }
  end;
end;
```

Двусторонний список. Что это такое, узнаете самостоятельно из №16.43 (стр. 164). Прорешайте все задачи из этого упр-я, чтобы "набить руку" к экзамену, мало ли... **Надеемся, справитесь.**

Иерархический список (маловероятно!) НО возможно!



```

type
  {тип подсписка:}
  list1 = ↑ node1;
  node1 = record
    elem: real;
    next: list1;
  end;
  {тип основного списка:}
  list = ↑ node;
  node = record
    elem: list1;
    next: list;
  end;
  
```

т.е. каждый элемент

списка типа list - это ссылка на список (типа list1) вещественных чисел.

Пара задач с экзамена:

1 задача Не используя глобальные переменные и операторы цикла и перехода, описать функцию num(L) где num(L) количество всех вещ. чисел, входящих в состав списка L типа list.

```

function num(L: list): real;
begin
  function num1(L1: list1): real;
  {количество чисел элементов списка L1 типа list1}
  begin if L1 = nil then num1 := 0 else
        num1 := 1 + num1(L1↑.next)
      end;
  begin {просмотр всех элементов списка L типа list:}
    if L = nil then num := 0 else num := num1(L↑.elem) +
      num(L↑.next)
    end;
  end;
end;
  
```

2 задача ... описать функцию sum(L) где возвращение суммы всех вещ. чисел, входящих в состав списка L

```

function sum(L: list): real;
begin
  function sum1(L1: list1): real;
  {сумма всех элементов списка L1 типа list1}
  begin if L1 = nil then sum1 := 0 else
        sum1 := L1↑.elem + sum1(L1↑.next)
      end;
  begin {просмотр всех элементов списка L типа list:}
    if L = nil then sum := 0 else sum := sum1(L↑.elem) + sum(L↑.next)
    end;
  end;
end;
  
```