

№	Название процедуры	Описание	Тест 1	Тест 2	Тест 3	Тест 4
1	<code>_def_InitializationNewList(var inp:char)</code>	Процедура, которая создаёт новый список, с одним элементом. Ничего не выводит, используется в других процедурах.	Вход Любое натуральное число (от 0 до 9) Выход Создаётся новый список с одним(первым) элементом из принятого числа	Любой символ (не пробел) Создаётся новый список с одним(первым) элементом из принятого символа	Символ, состоящий из пробела Процедура не создаёт новый список, и ничего не выводит	- -
2	<code>_def_PrintHeadElementList(ln1_h:list)</code>	Процедура, которая выводит первый(единственный) элемент списка, если он существует. Не используется в других процедурах.	Вход - Выход -	- -	- -	- -
3	<code>_def_PrintTailElementList(ln1_t:list)</code>	Процедура, которая последний(единственный) первый элемент списка, если он существует. Не используется в других процедурах.	Вход - Выход -	- -	- -	- -
4	<code>_AddElementForCurrentList_afterLastElement(var ln1_h, ln1_t:list; inp:char)</code>	Вставление элемента в конец списка. Если списка нет, то вызывается процедура №1, и в неё передаётся элемент, который ввёл пользователь. Процедура завершается, когда пользователь вводит пробел	Вход пробел Выход Список не изменяется. Если он был пуст, то он пустым и остаётся. Завершение работы процедуры.	п количество символов в строку В конец списка добавляется первый введённый символ	п количество символов, через enter, последний символ - пробел В конец списка последовательно вводятся все п символов. В итоге, список увеличивается на п элементов. Если были введены не символы, а наборы символов, то в список добавляется первый введённый символ, в каждой строке.	п количество символов, через enter, случайное количество пробелов, в случайных местах ввода Всё тоже самое, что и в предыдущем тесте, только ввод заканчивается, при введении первого пробела. Последующие символы не обрабатываются этой процедурой.
5	<code>_def_AddElementForCurrentList_afterLastElement(var ln1_h, ln1_t:list)</code>	Процедура, с выводом данных на экран пользователю. После вывода сообщения пользователю, передаёт вводимый элемент процедуре № 4	Вход пробел Выход Завершение процедуры	Любой символ (не пробел) Переход к процедуре №4, и передача ей этого символа.	- -	- -
6	<code>_AddElementForCurrentList_beforeFirstElement(var ln1_h, ln1_t:list; inp:char)</code>	Вставление элемента в начало списка. Если списка нет, то вызывается процедура №1, и в неё передаётся элемент, который ввёл пользователь. Процедура завершается, когда пользователь вводит пробел	Вход пробел Выход Список не изменяется. Если он был пуст, то он пустым и остаётся. Завершение работы процедуры.	п количество символов в строку В начало списка добавляется первый введённый символ	п количество символов, через enter, последний символ - пробел В начало списка последовательно вводятся все п символов. Каждый новый символ вставляется перед всеми остальными, т.е. становится первым. В итоге, список увеличивается на п элементов. Если были введены не символы, а наборы символов, то в список добавляется первый введённый символ, в каждой строке.	п количество символов, через enter, случайное количество пробелов, в случайных местах ввода Всё тоже самое, что и в предыдущем тесте, только ввод заканчивается, при введении первого пробела. Последующие символы не обрабатываются этой процедурой.
7	<code>_def_AddElementForCurrentList_beforeFirstElement(var ln1_h, ln1_t:list)</code>	Процедура, с выводом данных на экран пользователю. После вывода сообщения пользователю, передаёт вводимый элемент процедуре № 6	Вход пробел Выход Завершение процедуры	Любой символ (не пробел) Переход к процедуре №6, и передача ей этого символа.	- -	- -
8	<code>_StartProgram()</code>	Процедура, которая вызывается в начале работы программы. Не несёт в себе смысловой нагрузки, создавалась исключительно для улучшения визуального представления программы. Не несёт полезной вычислительной нагрузки.	Вход - Выход -	- -	- -	- -
9	<code>_def_PrintAllList(var ln1_h:list)</code>	Процедура вывода всех значений текущего списка. Создавалась исключительно для упрощения работы пользователя с программой. Не несёт полезной вычислительной нагрузки.	Вход Нет списка Выход Вывод: "[Список пуст]"	Список с 1 элементом Вывод единственного элемента списка, в квадратных скобках.	Список с п элементами Вывод всех п элементов списка в квадратных скобках, через запятую.	- -

10	<code>_DelElement_MultiPodbor(var ln1_h, ln1_t:list; mod:integer):char</code>	Функция, которая получает на вход переменную mod (0 - удалить начальный элемент списка, 1 удалить конечный элемент списка), которая также анализирует весь список, и если он состоит из одного элемента, также корректно завершает работу, после чего передаёт в качестве выходного аргумента, удалённый элемент из массива.	Вход Выход	Пустой список Пустой список не передаётся в эту функцию, так как такое состояние массива обрабатывается в других процедурах. При получении пустого списка, функция породит неисключаемую ошибку, и остановит работу программы.	Список из 1 элемента Вне зависимости от получаемого значения mod даёт этот единственный элемент, и возвращает удалённый элемент. Список становится пустым.	Список из n элементов В зависимости от получаемого значения mod, если mod = 0, удаляет первый элемент списка, если mod = 1, удаляет последний элемент списка. Возвращает удалённый элемент.	- -
11	<code>_def_DelFirstElement (var ln1_h, ln1_t:list)</code>	Процедура, которая удаляет первый элемент.	Вход Выход	Список, состоящий из от 1 до n элементов Выводит в консоль сообщение об удалении элемента. Вызывает функцию № 10 с mod = 0, которая удаляет первый (единственный) элемент. Выводит сообщение об удалённом элементе.	Пустой список Выводит в консоль сообщение, об невозможности удаления элемента, из пустого списка. Завершает процедуру	- -	- -
12	<code>_def_DelEndElement (var ln1_h, ln1_t:list)</code>	Процедура, которая удаляет последний элемент.	Вход Выход	Список, состоящий из от 1 до n элементов Выводит в консоль сообщение об удалении элемента. Вызывает функцию № 10 с mod = 1, которая удаляет последний(единственный) элемент. Выводит сообщение об удалённом элементе.	Пустой список Выводит в консоль сообщение, об невозможности удаления элемента, из пустого списка. Завершает процедуру	- -	- -
13	<code>_DelAllList(var ln1_h, ln1_t:list)</code>	Процедура, которая удаляет все элементы списка.	Вход Выход	Список, состоящий из от 1 до n элементов Выводит предупреждение об удалении всего списка, после получения утвердительной команды, последовательно удаляет все элементы списка, с конца, вызывая функцию №10 с mod = 1.	Пустой список Без ошибок обрабатывает исключение из цикла на первом шаге.	- -	- -
14	<code>_PrintVolumeElement (ln1_h, ln1_t:list):integer</code>	Функция, которая часто используется в последующих процедурах. На выходе выдаёт число элементов в списке.	Вход Выход	Список, состоящий из от 1 до n элементов Целое число элементов списка.	Пустой список Возвращает 0	- -	- -
15	<code>_DelElement_ForCount(var ln1_h, ln1_t:list; count:integer):char</code>	Функция удаления элемента, по номеру в списке. Возвращает удалённый элемент. Для определения длины массива используется функция № 14	Вход Выход	Пустой список, любое n значение на входе Не обрабатывается этой функцией. Такие значения не подаются на вход этой функции в коде.	Список из n элементов, значение m на входе, n<=m Корректное удаление элемента. Если m = 0, используется функция №10 mod = 0, если n = m, используется функция №10 mod = 1, в остальных случаях, если удаляемый элемент находится в середине массива, функция самостоятельно проводит его удаление.	Список из n элементов, значение m на входе, n>m Функция ничего не удаляет, и ничего не возвращает.	- -
16	<code>_def_DelRandElement (var ln1_h, ln1_t:list)</code>	Процедура удаления случайного элемента из списка	Вход Выход	Пустой список Выводится сообщение, что нельзя удалить ни одного элемента из пустого списка.	Список из 1 элемента Используется функция № 14 для получения длины массива. Далее генерируется случайное число из длины массива (В нашем случае - 0), далее применяется функция № 15, в аргумент которой посылаётся сгенерированный номер, которая выводит значение удалённого элемента.	Список из n элементов Используется функция № 14 для получения длины массива. Далее генерируется случайное число из длины массива, далее применяется функция № 15, в аргумент которой посылаётся сгенерированный номер, которая выводит значение удалённого элемента.	- -

17	<code>_def_DeSelectedElement (var ln1_h, ln1_t:list)</code>	Процедура удаления по номеру. Полностью использует функцию № 15, и также осуществляет вывод сообщений и значений в консоль.	Вход	Пустой список	Список, состоящий из от 1 до n элементов	-	-
			Выход	Выводит сообщение, что нельзя удалить элементы из пустого списка	Выводит удалённый элемент, если введённый номер <= длине массива. В противном случае, выводит ошибку, и прерывает операцию.	-	-
18	<code>_def_SearchSelectednElement (var ln1_h, ln1_t:list)</code>	Процедура, которая ищет элемент, по заданному номеру	Вход	Пустой список, любое n значение на входе	Список из n элементов, значение m на входе, n<=m	Список из n элементов, значение m на входе, n>m	-
			Выход	Выводит предупреждение, что поиск невозможен в пустом массиве, и завершает процедуру.	Выводит значение элемента с номером m	Выводит предупреждение, говоря, что введён неправильный номер, и завершает процедуру.	-
19	<code>_def_SearchMinAndMaxElement (var ln1_h, ln1_t:list)</code>	Процедура, которая выводит минимальный и максимальный элементы списка.	Вход	Список пуст	В списке n одинаковых элементов	В списке n различных или равных элементов	-
			Выход	Выводит предупреждение, что в пустом списке нет наибольших и наименьших элементов	Выводит единственный элемент, как минимальный, и как максимальный	Путём сравнения элементов друг с другом, выводит максимальный и минимальные элементы.	-
20	<code>_def_SearchSelectedElement (var ln1_h, ln1_t:list)</code>	Процедура поиска элемента по значению	Вход	Список пуст	В списке нет нужных нам элементов	В списке от 1 до n нужных нам элементов	-
			Выход	Выводится предупреждение, что в пустом списке мы не сможем найти ни одного элемента	Выводится сообщение о том, что не удалось найти ни одного подходящего элемента	Выводится количество найденных в списке нужных нам элементов	-
21	<code>_def_AddElementBeforeEachEven (var ln1_h, ln1_t:list)</code>	Процедура, которая добавляет вводимый пользователем элемент, после каждого чётного элемента списка. Введённые элементы не влияют на определение чётности списка, во время выполнения процедуры.	Вход	Пустой список	Список из 2x элеметнов	Список из n>2 элементов	-
			Выход	Ничего не выводит, процедура завершается	Просит ввести значение один раз, и вставляет его перед 2m элементов, но выходе получается список из 3 элементов.	Просит ввести новое значение каждый раз, когда бегунок доходит до чётного элемента списка. Вставляет вводимый элемент списка, перед текущим чётным, и идёт дальше, пока не встретит конец списка.	-
22	<code>_def_DeElementBeforeEachEven (var ln1_h, ln1_t:list)</code>	Процедура, которая удаляет элемент, перед каждым чётным элементом. Удаляемые элементы не влияют на определение чётности элемента, во время выполнения процедуры.	Вход	Пустой список	Список из 2x элеметнов	Список из n>2 элементов	-
			Выход	Ничего не выводит, процедура завершается	Удаляет 1 элемент, используя функцию № 15, выводит удалённый элемент, и его номер по списку.	Удаляет каждый элемент, перед чётным, используя функцию № 15. Процедура заканчивается, когда заканчивается список.	-
23	<code>_def_FillElement (var ln1_h, ln1_t:list)</code>	Процедура, которая создаёт новый список, и заполняет его вводимое количество раз, вводимым элементом	Вход	a = пробел	i <= 0	-	-
			Выход	Выводится предупреждение о том, что нельзя заполнить массив пробелами.	Выводится предупреждение о том, что нужно ввести i > 0	-	-
24	<code>_def_EndOfProgramm()</code>	Процедура завершения программы. Выводит на экран завершающие сообщение, и выводит основную процедуру из цикла, завершая её.	Вход	-	-	-	-
			Выход	-	-	-	-
25	<code>MainP()</code>	Основная процедура, которая при помощи вывода и ввода данных пользователем, тем или иным образом использует все процедуры программы, позволяя пользователю производить указанные в процедуре операции, над динамическим массивом.	Вход	Число от 0 до 17	Число не от 0 до 17	Символ	-
			Выход	Выполнение заданной процедуры	Вывод ошибки, и посторонний вывод списка всех команд	Ошибка. Эта процедура не принимает на вход символов. Программа остановится, без возможности продолжения.	-