|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт информационных технологий

Кафедра вычислительной техники

|  |  |
| --- | --- |
| **КУРСОВАЯ РАБОТА** | |
| **по дисциплине** | |
| **«Математическая логика и теория алгоритмов»** | |
| **Тема курсовой работы**  **«**Программы на языке программирования ПРОЛОГ**»**  **Задачи варианта № 21** | |
| Студент группы ИВБО-03-17 | Ковалев Даниил Дмитриевич |
|  |  |

|  |  |  |
| --- | --- | --- |
| Работа представлена к защите | «\_\_»\_\_\_\_\_\_\_201\_\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Допущен к защите» | «\_\_»\_\_\_\_\_\_\_201\_\_\_ г. | *(подпись руководителя)* |

Москва 2018

**Задача №12.1**

**Условие задачи:**

Вычислить сумму элементов списка целых чисел, находящихся на чётных позициях.

**Код программы:**

domains

i = integer % целое число

list = i\* % список целых чисел

predicates

nondeterm go % главное меню программы

nondeterm do(char) % запускает выполнение полученного на вход задания

nondeterm readList(list) % вводит список от пользователя

% задание 12.2 - Вычислить сумму элементов списка, находящихся на нечётных позициях.

nondeterm task\_12(list, i, i)

clauses

go :-

write("=== SELECT TASK ===\n"),

write("Press 1 - task 12.1\n"),

write("Press 0 - to exit\n\n"),

write("Task: "), readchar(A),

write(A), nl, do(A),

go.

do('1') :- % задание 12.1

write("LIST: "), nl, readList(L),

write("LIST: ", L), nl,

task\_12(L, 1, Summ),

write("Summ of elements on even positions: ", Summ), nl.

do('0') :-

write("Good bye!"), nl,

exit.

% ввод списка от пользователя

readList([H|T]) :-

write("Add element: "),

readint(H), % читаем число и помещаем его в голову

readList(T). % запускаем себя для хвоста

% если введено не число – прекращаем рекурсию, возвращаем пустой список

readList([]).

% задание 12.1 - Вычислить сумму элементов списка, находящихся на чётных позициях.

% На основе счётчика (N), который на каждом элементе увеличивается на 1, проверяем является

% ли текущая позиция нечётной (остаток от деления счётчика на 2 равен 0), и если да, то

% сумма будет равна текущему элементу + сумме элементов на нечётных позициях хвоста

task\_12([], \_, 0).

task\_12([H|T], N, S) :-

N mod 2 = 0,

NewN = N + 1,

task\_12(T, NewN, TailS),

S = TailS + H.

task\_12([\_|T], N, S) :-

NewN = N + 1,

task\_12(T, NewN, S).

goal go.

**Задача №18.**

**Условие задачи:**

Удалить из списка все отрицательные элементы.

**Код программы:**

domains

i = integer % целое число

list = i\* % список целых чисел

predicates

nondeterm go % главное меню программы

nondeterm do(char) % запускает выполнение полученного на вход задания

nondeterm readList(list) % вводит список от пользователя

% задание 18 - Удалить отрицательные элементы списка

nondeterm task\_18(list, list)

clauses

go :-

write("=== SELECT TASK ===\n"),

write("Press 1 - task 18\n"),

write("Press 0 - to exit\n\n"),

write("Task: "), readchar(A),

write(A), nl, do(A),

go.

do('1') :- % задание 18

write("LIST: "), nl, readList(LIST), nl,

write("LIST: ", LIST), nl,

task\_18(LIST, NewList),

write("RESULT LIST: ", NewList), nl.

do('0') :-

write("Good bye!"), nl,

exit.

% ввод списка от пользователя

readList([H|T]) :-

write("Add element: "),

readint(H), % читаем число и помещаем его в голову

readList(T). % запускаем себя для хвоста

% если введено не число – прекращаем рекурсию, возвращаем пустой список

readList([]).

% задание 18 - Удалить отрицательные элементы списка

% Если элемент списка >= 0, заносим его в список-результат

% Если нет - пропускаем. И повторяем для хвостов списков.

% Для пустого списка результат - пуской список

task\_18([H|T], [H|T2]) :- H >= 0, task\_18(T, T2).

task\_18([\_|T], L2) :- task\_18(T, L2).

task\_18([],[]).

goal go.

**Задача №19.**

**Условие задачи:**

Создать список из N начальных чисел ряда Фибоначчи.

**Код программы:**

domains

i = integer % целое число

list = i\* % список целых чисел

predicates

nondeterm go % главное меню программы

nondeterm do(char) % запускает выполнение полученного на вход задания

nondeterm readList(list) % вводит список от пользователя

% задание 19 - Создать список из N начальных чисел ряда Фибоначчи

nondeterm task\_19(i, i, i, list)

clauses

go :-

write("=== SELECT TASK ===\n"),

write("Press 1 - task 19\n"),

write("Press 0 - to exit\n\n"),

write("Task: "), readchar(A),

write(A), nl, do(A),

go.

do('1') :- % задание 19

write("N = "), readint(N),

task\_19(N, 0, 0, R),

write(R), nl.

do('0') :-

write("Good bye!"), nl,

exit.

% ввод списка от пользователя

readList([H|T]) :-

write("Add element: "),

readint(H), % читаем число и помещаем его в голову

readList(T). % запускаем себя для хвоста

% если введено не число – прекращаем рекурсию, возвращаем пустой список

readList([]).

% задание 19 - Создать список из N начальных чисел ряда Фибоначчи

% если предыдущие числа равны 0 и 0, то мы находимся в начале ряда

% устанавливаем первое число в 1

% если предыдущие числа равны 0 и 1, то мы на втором элементе ряда

% а он тоже будет единицей

% для остальных случаев новое число равно сумме двух предыдущих

% при этом в рекурсию мы передаём уже второе предыдущее и текущее число

% N - счётчик, который уменьшается на 1 с каждым числом, и процесс остановится

% когда он будет равен 0 (вернётся пустой список)

task\_19(0, \_, \_, []).

task\_19(N, 0, 0, [1|T]) :- N2 = N - 1, task\_19(N2, 1, 0, T).

task\_19(N, 1, 0, [1|T]) :- N2 = N - 1, task\_19(N2, 1, 1, T).

task\_19(N, A, B, [C|T]) :-

C = A + B,

N2 = N - 1,

task\_19(N2, B, C, T).

goal go.

**Задача №26.**

**Условие задачи:**

Заменить в исходном списке N-ый элемент заданной константой.

**Код программы:**

domains

i = integer % целое число

list = i\* % список целых чисел

predicates

nondeterm go % главное меню программы

nondeterm do(char) % запускает выполнение полученного на вход задания

nondeterm readList(list) % вводит список от пользователя

% задача 26 - Заменить в исходном списке N-ый элемент заданной константой.

nondeterm task\_26(list, i, i, list)

clauses

go :-

write("=== SELECT TASK ===\n"),

write("Press 1 - task 26\n"),

write("Press 0 - to exit\n\n"),

write("Task: "), readchar(A),

write(A), nl, do(A),

go.

do('1') :- % задание 26

write("List: "), nl, readList(L),

write("N = "), readint(N),

write("C = "), readint(C),

task\_26(L, N, C, R),

write("Result: ", R), nl.

do('0') :-

write("Good bye!"), nl,

exit.

% ввод списка от пользователя

readList([H|T]) :-

write("Add element: "),

readint(H), % читаем число и помещаем его в голову

readList(T). % запускаем себя для хвоста

% если введено не число – прекращаем рекурсию, возвращаем пустой список

readList([]).

% задача 26 - Заменить в исходном списке N-ый элемент заданной константой.

% идём по списку и уменьшаем N на 1 на каждом элементе, и копируем элементы в конечный список

% и когда N = 0, значит, мы дошли до нужного элемента, помещаем в результат константу вместо исходного элемента

task\_26([], \_, \_, []).

task\_26([\_|T], 1, C, [C|RT]) :- task\_26(T, 0, C, RT).

task\_26([H|T], N, C, [H|R]) :- N2 = N - 1, task\_26(T, N2, C, R).

goal go.

**Задача №31.**

**Условие задачи:**

По списку элементов сформировать новый список элементов согласно заданному списка порядковых номеров.

**Код программы:**

domains

i = integer % целое число

list = i\* % список целых чисел

predicates

nondeterm go % главное меню программы

nondeterm do(char) % запускает выполнение полученного на вход задания

nondeterm readList(list) % вводит список от пользователя

% задание 31 – По списку элементов сформировать новый список элементов согласно списка порядковых номеров

nondeterm task\_31(list, list, list, i, list)

clauses

go :-

write("=== SELECT TASK ===\n"),

write("Press 1 - task 31\n"),

write("Press 0 - to exit\n\n"),

write("Task: "), readchar(A),

write(A), nl, do(A),

go.

do('1') :- % задание 31

write("LIST OF ELEMENTS: "), nl, readList(LIST), nl,

write("LIST OF NUMBERS: "), nl, readList(NUMS), nl,

write("LIST OF ELEMENTS: ", LIST), nl,

write("LIST OF NUMBERS: ", NUMS), nl,

task\_31(LIST, LIST, NUMS, 1, NewList),

write("RESULT LIST: ", NewList), nl.

do('0') :-

write("Good bye!"), nl,

exit.

% ввод списка от пользователя

readList([H|T]) :-

write("Add element: "),

readint(H), % читаем число и помещаем его в голову

readList(T). % запускаем себя для хвоста

% если введено не число – прекращаем рекурсию, возвращаем пустой список

readList([]).

% задание 31 – По списку элементов сформировать новый список элементов согласно списка порядковых номеров

% идём по списку, сравнивая текущий элемент списка позиций с текущей позицией (POS)

% Если они совпадают - то вставляем текущий элемент списка в результат, сбрасываем счётчик позиций,

% и повторяем процесс заново для всего списка. Если же элемент списка позиций с текущей не совпал,

% то двигаемся по списку дальше, пока не найдём нужную позицию и соответствующий ей элемент

task\_31(\_, \_, [], \_, []).

task\_31(FullList, [H|\_], [Pos|NT], Pos, [H|RT]) :-

task\_31(FullList, FullList, NT, 1, RT).

task\_31(FullList, [\_|T], PosList, Pos, Res) :-

NewPos = Pos + 1,

task\_31(FullList, T, PosList, NewPos, Res).

goal go.

**Задача №32.2.**

**Условие задачи:**

Объединить два списка одинаковой длины в третий так,

чтобы вначале списка были элементы с н/ч позиций первого

списка, а в конце с н/ч - из второго. н - нечётные, ч - чётные. 2(нч).

**Код программы:**

**Задача №37.**

**Условие задачи:**

Число, представленное списком двоичных разрядов напечатать в виде десятичного числа.

**Код программы:**

**Задача №42.**

**Условие задачи:**

Число, представленное списком двоичных разрядов напечатать в виде десятичного числа.

**Код программы:**

**Задача №43.**

**Условие задачи:**

Число, представленное списком двоичных разрядов напечатать в виде десятичного числа.

**Код программы:**

**Задача №53.**

**Условие задачи:**

Число, представленное списком двоичных разрядов напечатать в виде десятичного числа.

**Код программы:**

**Задача №63.**

**Условие задачи:**

Число, представленное списком двоичных разрядов напечатать в виде десятичного числа.

**Код программы:**

**Задача №71.**

**Условие задачи:**

Число, представленное списком двоичных разрядов напечатать в виде десятичного числа.

**Код программы:**

**Задача №84.**

**Условие задачи:**

Число, представленное списком двоичных разрядов напечатать в виде десятичного числа.

**Код программы:**