МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования «Южно-Уральский государственный университет»

(национальный исследовательский университет)

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

рекурсия

(название практической работы)

ОТЧЕТ

по практической работе № 3

по дисциплине «Функциональное и логическое программирование»

Выполнил:

студент группы КЭ–401

/ Д.А. Иванов /

(подпись)

« » 2024 г.

Проверил:

преподаватель кафедры СП

/ М.А. Ческидова /

(подпись)

« » 2024 г.

# Задание 1

На листинге 1 представлен код для генерации всех целых чисел, начиная с 1 и заканчивая 7.

Листинг 1 ­– Генерации целых чисел.

% Рекурсивный вывод численных значений от 1 до N

print\_numbers(N) :-

N < 8,

write(N), write(' '),

N1 is N + 1,

print\_numbers(N1).

На листинге 2 представлен код для генерации целых чисел он 1 до 7 в порядке убывания.

Листинг 2 ­– Генерации целых чисел в порядке убывания.

% Рекурсивный вывод численных значений от N до 1

print\_numbers\_dc(N) :-

N > 0,

write(N), write(' '),

N1 is N - 1,

print\_numbers\_dc(N1).

На рисунке 1 представлены результаты выполнения этих 2 программ.

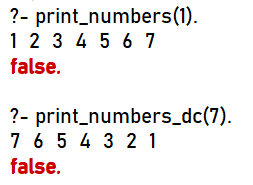


Рисунок 1 – Результат выполнения задания 1

# Задание 2

Найдите НОД по следующему правилу:

1. если X=Y, то NOD=X (ответ);
2. если X<Y, то искать NOD как наибольший общий делитель для X и разности Y-X;
3. если X>Y, то формулировка аналогична правилу 2),если X и Y поменять местами.

На листинге 3 представлен код для поиска НОД.

Листинг 3 – Поиск НОД.

% Рекурсивный поиск НОД

nod(X, X, X) :- !.

nod(X, Y, NOD) :-

X < Y, !,

Z is Y - X,

nod(X, Z, NOD).

nod(X, Y, NOD) :-

X > Y,

Z is X - Y,

nod(Z, Y, NOD).

Результат представлен на рисунке 2.



Рисунок 2 – Результат выполнения задания 2

# Задание 3

На листинге 4 представлен код для вычисления суммы ряда.

Листинг 4 ­– Вычисление суммы ряда.

% Рекурсивный поиск суммы чисел от 1 до N

sum\_series(0, 0).

sum\_series(N, Sum) :-

N > 0,

N1 is N - 1,

sum\_series(N1, Sum1),

Sum is Sum1 + N.

На листинге 5 представлен код для вычисления суммы ряда нечетных чисел.

Листинг 5 ­– Вычисление суммы ряда нечетных чисел.

% Рекурсивный поиск суммы нечётных чисел от 1 до N

sum\_odd\_series(0, 0).

sum\_odd\_series(N, Sum) :-

N > 0,

N1 is N - 1,

sum\_odd\_series(N1, Sum1),

(

N mod 2 =:= 1 ->

Sum is Sum1 + N;

Sum is Sum1

).

На рисунке 3 представлены результаты выполнения этих программ.

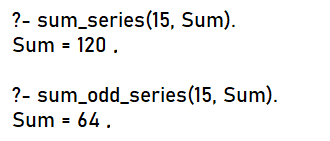


Рисунок 3 – Результат выполнения задания 3

# Задание 4

Определите правило for(N1, N2, X), которое с помощью перебора порождает все целые числа, отвечающие условию N1<=X<=N2.

На листинге 6 представлен код для правила for(N1, N2, X).

Листинг 6 ­– Правило for(N1, N2, X).

% Рекурсивный вывод численных значений от N1 до N2

for(N1, N2, N1) :-

N1 =< N2.

for(N1, N2, X) :-

N1 < N2,

N11 is N1 + 1,

for(N11, N2, X).

На рисунке 4 представлен результат выполнения этой программы.

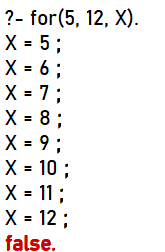


Рисунок 4 – Результат выполнения задания 4

# Задание 5

Имеется несколько населенных пунктов. Некоторые пары пунктов соединены дорогами, причем двигаться по дорогам можно только в одном направлении. Определить, можно ли попасть по этим дорогам из одного пункта в другой. Пусть, к примеру, соединение между дорогами задано схемой, изображенной на рисунке 5.

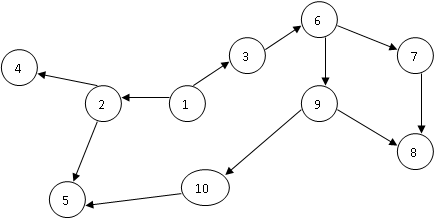


Рисунок 5 – Схема населенных пунктов

На листинге 7 представлен код для определения существования пути из одного пункта в другой.

Листинг 7 ­– Определение существования пути.

% Определяем правила для построения пути между пунктами

way(1, 2).

way(1, 3).

way(2, 4).

way(2, 5).

way(3, 6).

way(6, 7).

way(6, 9).

way(7, 8).

way(9, 8).

way(9, 10).

way(10, 5).

path(X, X, [X]).

% Рекурсивный случай: путь из X в Y через промежуточные точки

path(X, Y, [X|Path]) :-

way(X, Z),

path(Z, Y, Path).

% Проверка достижимости точки Y из точки X

reachable(X, Y) :-

path(X, Y, \_).

На рисунке 6 представлены результаты выполнения этой программы.

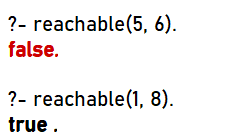


Рисунок 6 – Результат выполнения задания 5

# Задание 6

В соревнованиях по бегу Юра, Гриша и Толя заняли три первых места. Какое место занял каждый ребенок, если Гриша занял не второе и не третье место, а Толя – не третье?

На листинге 8 представлен код программы для решения этой задачи.

Листинг 8 ­– Соревнования по бегу.

% Определяем факты

runner(yura).

runner(grisha).

runner(tolya).

% Определяем возможные места

place(1).

place(2).

place(3).

% Решение задачи

solution(YuraPlace, GrishaPlace, TolyaPlace) :-

% Каждый бегун занимает место

place(YuraPlace),

place(GrishaPlace),

place(TolyaPlace),

% Все места различны

YuraPlace \= GrishaPlace,

YuraPlace \= TolyaPlace,

GrishaPlace \= TolyaPlace,

% Условия задачи

GrishaPlace \= 2,

GrishaPlace \= 3,

TolyaPlace \= 3.

% Вывод результата

print\_result :-

solution(YuraPlace, GrishaPlace, TolyaPlace),

write('Юра занял '), write(YuraPlace), write(' место.'), nl,

write('Гриша занял '), write(GrishaPlace), write(' место.'), nl,

write('Толя занял '), write(TolyaPlace), write(' место.'), nl.

На рисунке 7 представлены результаты выполнения программы.

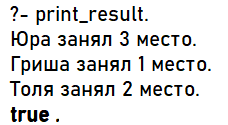


Рисунок 7 – Результат выполнения задания 6

# Задание 7

В бутылке, стакане, кувшине и банке находятся молоко, лимонад, квас и вода. Известно, что вода и молоко не в бутылке, сосуд с лимонадом находится между кувшином и сосудом с квасом, в банке – не лимонад и не вода. Стакан находится около банки и сосуда с молоком. Как распределены эти жидкости по сосудам?

На листинге 10 представлен код для решения этого задания.

Листинг 10 ­– Распределения жидкостей по сосудам.

% Определяем сосуды и жидкости

сосуд(бутылка).

сосуд(стакан).

сосуд(кувшин).

сосуд(банка).

жидкость(молоко).

жидкость(лемонад).

жидкость(квас).

жидкость(вода).

% Решение задачи

solution(Бутылка, Стакан, Кувшин, Банка) :-

% Каждый сосуд содержит одну из жидкостей

жидкость(Бутылка),

жидкость(Стакан),

жидкость(Кувшин),

жидкость(Банка),

% Все жидкости различны

Бутылка \= Стакан,

Бутылка \= Кувшин,

Бутылка \= Банка,

Стакан \= Кувшин,

Стакан \= Банка,

Кувшин \= Банка,

% Условия задачи

Бутылка \= вода,

Бутылка \= молоко,

Кувшин \= лемонад,

Кувшин \= квас,

Банка \= лемонад,

Банка \= вода,

Стакан \= молоко,

Банка \= молоко.

% Вывод результата

print\_result :-

solution(Бутылка, Стакан, Кувшин, Банка),

write('Бутылка содержит '), write(Бутылка), nl,

write('Стакан содержит '), write(Стакан), nl,

write('Кувшин содержит '), write(Кувшин), nl,

write('Банка содержит '), write(Банка), nl.

На рисунке 8 представлены результаты выполнения этой программы.

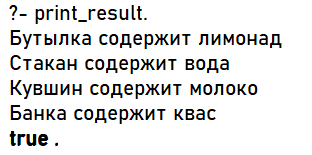


Рисунок 8 – Результат выполнения задания 7

# Задание 8

Три подруги вышли в белом, зеленом и синем платьях и туфлях. Известно, что только у Ани цвета платья и туфлей совпадали. Ни туфли, ни платье Вали не были белыми. Наташа была в зеленых туфлях. Определить цвета платья и туфель на каждой из подруг.

На листинге 11 представлен код для определения цвета платья и туфель каждой девочки.

Листинг 11 ­– Цвет платья и туфель.

% Определяем цвета

цвет(белый).

цвет(зеленый).

цвет(синий).

% Решение задачи

solution(ПлатьеАни, ТуфлиАни, ПлатьеВали, ТуфлиВали, ПлатьеНаташи, ТуфлиНаташи) :-

% Каждый элемент одежды имеет один из цветов

цвет(ПлатьеАни), цвет(ТуфлиАни),

цвет(ПлатьеВали), цвет(ТуфлиВали),

цвет(ПлатьеНаташи), цвет(ТуфлиНаташи),

% Только у Ани цвета платья и туфель совпадают

ПлатьеАни = ТуфлиАни,

ПлатьеВали \= ТуфлиВали,

ПлатьеНаташи \= ТуфлиНаташи,

% Ни туфли, ни платье Вали не были белыми

ПлатьеВали \= белый,

ТуфлиВали \= белый,

% Наташа была в зеленых туфлях

ТуфлиНаташи = зеленый,

% Все платья разного цвета

ПлатьеАни \= ПлатьеВали, ПлатьеАни \= ПлатьеНаташи, ПлатьеВали \= ПлатьеНаташи,

% Все туфли разного цвета

ТуфлиАни \= ТуфлиВали, ТуфлиАни \= ТуфлиНаташи, ТуфлиВали \= ТуфлиНаташи.

% Вывод результата

print\_result :-

solution(ПлатьеАни, ТуфлиАни, ПлатьеВали, ТуфлиВали, ПлатьеНаташи, ТуфлиНаташи),

write('Аня: платье - '), write(ПлатьеАни), write(', туфли - '), write(ТуфлиАни), nl,

write('Валя: платье - '), write(ПлатьеВали), write(', туфли - '), write(ТуфлиВали), nl,

write('Наташа: платье - '), write(ПлатьеНаташи), write(', туфли - '), write(ТуфлиНаташи), nl.

На рисунке 9 представлены результаты выполнения этой программы.

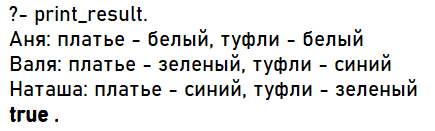


Рисунок 9 – Результат выполнения задания 1

# задание 9

Для решения следующих заданий используется предикат distinct, проверяющий все ли элементы предаваемого списка уникальны. Для этого просто сравнивается длина изначального списка и отсортированного списка. Код этого предиката представлен на листинге 12.

Листинг 12 ­– предикат distinct.

distinct(List) :-

sort(List, Sorted),

length(List, Length),

length(Sorted, Length).

1. Мама, папа и я сидели на скамейке. В каком порядке мы сидели на скамейке, если известно, что я сидел слева от папы, а мама слева от меня.
2. Даны четыре числа X, Y, Z и T. X меньше Y и меньше T; Y больше Z и больше T; Z больше X и меньше T. В каком порядке расположены эти числа?
3. Возле почты растут 6 деревьев: сосна, береза, липа, тополь, ель и клен. Какое из этих деревьев самое высокое и какое самое низкое, если известно, что береза ниже тополя, а липа выше клена, сосна ниже ели, липа ниже березы, сосна выше тополя?

На листинге 13 представлен код программы для решения первой задачи.

Листинг 13 ­– Соревнования по бегу.

family(me).

family(mom).

family(dad).

left(me, dad).

left(mom, me).

range\_a(X, Y, Z) :-

left(X, Y),

left(Y, Z).

print\_a() :-

family(X),

family(Y),

family(Z),

distinct([X, Y, Z]),

range\_a(X, Y, Z),

write(X), write(' '),

write(Y),write(' '),

write(Z).

На рисунке 10 представлены результаты выполнения программы.

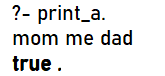


Рисунок 10 – Результат выполнения задания 9.1

На листинге 14 представлен код программы для решения второй задачи.

Листинг 14 ­– Порядок расположения чисел.

num(x).

num(y).

num(z).

num(t).

less(x, y).

less(x, t).

less(z, y).

less(t, y).

less(x, z).

less(z, t).

range\_b(X, Y, Z, T) :-

less(X, Y),

less(Y, Z),

less(Z, T).

print\_b :-

num(X),

num(Y),

num(Z),

num(T),

distinct([X, Y, Z, T]),

range\_b(X, Y, Z, T),

write(X), write(' < '),

write(Y),write(' < '),

write(Z),write(' < '),

write(T).

На рисунке 11 представлены результаты выполнения этой программы.

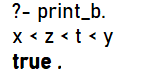


Рисунок 11 – Результат выполнения задания 9.2

На листинге 15 представлен код программы для решения третей задачи.

Листинг 15 ­– Высота деревьев.

tree(ель).

tree(сосна).

tree(тополь).

tree(береза).

tree(липа).

tree(клен).

less(тополь, сосна).

less(липа, береза).

less(береза, тополь).

less(клен, липа).

less(сосна, ель).

range\_c(Д1, Д2, Д3, Д4, Д5, Д6) :-

less(Д1, Д2),

less(Д2, Д3),

less(Д3, Д4),

less(Д4, Д5),

less(Д5, Д6).

print\_c :-

tree(Д1),

tree(Д2),

tree(Д3),

tree(Д4),

tree(Д5),

tree(Д6),

distinct([Д1, Д2, Д3, Д4, Д5, Д6]),

range\_c(Д1, Д2, Д3, Д4, Д5, Д6),

write("Самое большое - "), write(Д6),

write("Самое маленькое - "), write(Д1).

На рисунке 12 представлены результаты выполнения этой программы

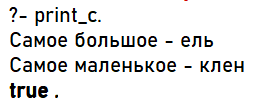


Рисунок 12 – Результат выполнения задания 9.3

# задание 10

Найти наибольшую и наименьшую цифры в записи данного натурального числа.

На листинге 16 представлен код программы для определения наибольшей и наименьшей цифры в записи натурального числа.

Листинг 15 ­– Наибольшая и наименьшая цифра.

% Базовый случай: число однозначное

find\_min\_max(N, N, N) :-

N < 10.

% Рекурсивный случай: число многозначное

find\_min\_max(N, Min, Max) :-

N >= 10,

N1 is N mod 10,

N2 is N // 10,

find\_min\_max(N2, Min1, Max1),

(

N1 < Min1 -> Min = N1, Max = Max1;

N1 > Max1 -> Min = Min1, Max = N1;

N1 =< Max1, N1 >= Min1 -> Min = Min1, Max = Max1

).

На рисунке 12 представлены результаты выполнения этой программы

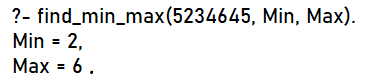


Рисунок 12 – Результат выполнения задания 9.3