

Міністерство освіти і науки України
Національний технічний університет України „КПІ”
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки
інформації та управління

ЗВІТ

до лабораторної роботи № 9
з предмету:

„МУЛЬТИПАРАДИГМЕННЕ ПРОГРАМУВАННЯ”

**Виконав
студент**

*ІІІ-61 Кушка Михайло
Олександрович, 3-й курс, ІІІ-6116*

(№ групи, прізвище, ім'я, по батькові, курс, номер
залікової книжки)

Прийняв

Очеретяний О. К.

(посада, прізвище, ім'я, по батькові)

Київ 2018

ЗМІСТ

1. ПОСТАНОВКА ЗАДАЧІ	3
2. РЕЗУЛЬТАТИ РОБОТИ.....	4
3. ВИСНОВОК.....	5
4. КОД ПРОГРАМИ	6

1. ПОСТАНОВКА ЗАДАЧІ

Завдання:

1. Підсумувати цілі позитивні числа, які йдуть одне за другим з кроком **d**, закінчуються числом **n**. Значення **d** та **n** вводяться за запитом з екрану монітора (наприклад, **d=3** та **n=11**, підсумок **11+8+5+2=26**). У випадку **d>=n** підсумок дорівнює **n**.

2. Звести число **a** у цілий ступінь **m** (**m** може бути позитивним, нульовим, негативним). Скласти два варіанти програми:

а) використовуючи рекурсивний вираз $a^m = a^{(m-1)} * a$;

б) використовуючи можливість обчислень за формулою $a^{(2*m)} = a^m * a^m$ для парного ступеня.

Визначити максимальний ступінь, у який можна звести число **a=2** по першому й другому варіантах програми. Результати й причини їхнього розходження відобразити у висновках звіту.

3. Знайти підсумок цілих послідовних чисел від 0 до **N**.

4. Обчислити значення наступних функцій, використовуючи розкладання В ряд (в ітеративному варіанті програми задавати точність обчислення функції):

$$\pi^2/6 = 1 + 1/2^2 + 1/3^2 + \dots + 1/k^2 + \dots$$

2. РЕЗУЛЬТАТИ РОБОТИ

```
| ?- sum_up(11, 3).
```

The sum is 26

true ? ;

no

```
| ?- pow(2, 4, X).
```

X = 16 ? ;

no

```
| ?- pow(2, -4, X).
```

X = 0.0625 ? '

Action (; for next solution, a for all solutions, RET to stop) ?

yes

```
| ?- pow(2, 0, X).
```

X = 1 ? ;

no

```
| ?- sum_line(5).
```

The sum is 15

true ? ;

no

```
| ?- calc(9999).
```

The result is 1.6448340518460649

true ? ;

(5 ms) no

3. ВИСНОВОК

В ході даної лабораторної роботи я навчився писати більш складні програми на Prolog. Обчислюючи зведення до степеню розкладаючи парні степені за формулою $a^{(2*m)} = a^m * a^m$ працює швидше, ніж звичайне обчислення рекурсивно ($a^m = a^{(m-1)} * a$), оскільки в разі розкладання ми обчислюємо a^m лише один раз, замість того, щоб обчислювати a^{2m} .

4. КОД ПРОГРАМИ

```
%%%%%%%%%%  
% 1 %  
%%%%%%%%%%
```

```
sum_up(N, D) :-  
    Sum is 0,  
    ( D >= N ->  
        format('The result is ~w ~n', [D]) ;  
        sum(N, D, Sum)  
    ).
```

```
sum(N, D, Sum) :-  
    N <= 0,  
    format('The sum is ~w ~n', [Sum]).
```

```
sum(N, D, Sum) :-  
    N > 0,  
    Res is N + Sum,  
    New is N - D,  
    sum(New, D, Res).
```

```
%%%%%%%%%%  
% 2 %  
%%%%%%%%%%
```

```
pow(A, N, X) :-  
    ( N >= 0 ->  
        power(A, N, 1, X) ;  
        negative_power(A, N, 1, X)  
    ).
```

```
power(A, N, Res, X) :-  
    N = 0,  
    X is Res.
```

```
power(A, N, Res, X) :-  
    N > 0,  
    Tmp is A * Res,  
    New is N - 1,  
    power(A, New, Tmp, X).
```

```
negative_power(A, N, Res, X) :-  
    N = 0,  
    X is Res.
```

```
negative_power(A, N, Res, X) :-  
    N < 0,  
    Tmp is 1 / A * Res,
```

```

New is N + 1,
negative_power(A, New, Tmp, X).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

is_even(X) :-
    Y is X // 2,
    X == Y * 2.

pow2(A, N, X) :-
    ( is_even(N) ->
        pow_even(A, N, 1, 1, X) ;
        pow_odd(A, N, 1, 1, X)
    ).

pow_even(A, N, Rese, Reso, X) :-
    N = 0,
    X is Rese.

pow_even(A, N, Rese, Reso, X) :-
    N > 0,
    New is N // 2,
    pow_odd(A, New, Reso, XX),
    X is XX * XX.

pow_odd(A, N, Rese, Reso, X) :-
    N = 1,
    X is Reso.

pow_odd(A, N, Rese, Reso, X) :-
    N > 1,
    Tmp is A * Reso,
    New is N - 1,
    format('~w ~n', [Tmp]),
    ( is_even(New) ->
        pow_even(A, New, Rese, Tmp, X) ;
        pow_odd(A, New, Rese, Tmp, X)
    ).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sum_line(N) :-
    s_line(N, 0).

s_line(0, Sum) :-
    format('The sum is ~w', [Sum]).

s_line(N, Sum) :-

```

```

N > 0,
Res is N + Sum,
New is N - 1,
s_line(New, Res).

%%%%%%%%%%
% 4 %
%%%%%%%%%%
% pi ** 2 / 6 == 1.6449340668482264
% Calculate Pi^2 / 6 with N iterations
calc(N) :-
    calculate(N, 1, 0).

calculate(N, N, Res) :-
    format('The result is ~w ~n', [Res]).

calculate(N, K, Res) :-
    K <= N,
    Tmp is 1 / K ** 2 + Res,
    New is K + 1,
    calculate(N, New, Tmp).

```