/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Создайте предикат, находящий в исходной строке слово максимальной (минимальной) длины.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

predicates

length(string,integer)

getWord(string,string,string,string)

notlonger(string,string)

smallest(string,string)

biggest(string,string)

clauses

length("",0) :- !.

length(S,N) :-

frontchar(S, \_, S1),

length(S1, N1),

N = N1 + 1.

getWord(S, W, E, T) :-

frontchar(S, C, S1),

C <> ' ',

!,

str\_char(SC, C),

concat(W, SC, W1),

getWord(S1, W1, E, T).

getWord(S, W, W, R) :-

frontchar(S, ' ', R),

!.

getWord(S, W, W, S).

notlonger(A, B) :-

length(A, LenA),

length(B, LenB),

LenA>0,

LenA<=LenB.

smallest("", "") :- !.

smallest(S, W) :-

getWord(S, "", WW, SS),

smallest(SS, W),

notlonger(W, WW),

!.

smallest(S,W) :-

getWord(S, "", W, \_).

biggest("", "") :- !.

biggest(S, W) :-

getWord(S, "", WW, SS),

biggest(SS, W),

notlonger(WW, W),

!.

biggest(S,W) :-

getWord(S, "", W, \_).

goal

write("Введите предложение: "),

readln(String),

smallest(String, Smallest),

biggest(String, Biggest).

