

```

% этот скрипт вытаскивает из таблицы "InputSIM2023.xlsx" значения
% настроечных коэффициентов.
% К этому скрипту обращается скрипт direct_taskSIM, если пользователь
% выбрал опцию 1 (считаем изотопный состав по известным входным данным)

T = readtable("InputSIM2023.xlsx"); % обращаемся к таблице с входными данными
moisture_source = T{1:13,2}; % выбираем массив данных с условиями в источнике влаги
teta_s = moisture_source(1,1); % температура поверхности океана (градусы C)
freemem = moisture_source(12,1);
gradHs_T = moisture_source(13,1);
Humid_quest = input(['нажми 1, если влажность - независимый параметр, или 2, ' ...
    'если она - функция температуры ']);
if Humid_quest == 1
    H_s0 = moisture_source(2,1); % влажность (в долях единицы) берется из таблицы
elseif Humid_quest == 2
    H_s0 = teta_s*gradHs_T+freemem; % влажность считается по температуре в источнике
else
    disp('нужно ввести 1 или 2. Попробуй еще раз');
    return;
end
k_180 = moisture_source(3,1); % кинетические коэффициенты
k_D = moisture_source(4,1);
k_170 = moisture_source(5,1);
delta18_m = moisture_source(6,1); % изотопный состав воды
deltaD_m = moisture_source(7,1);
delta17_m = moisture_source(8,1);
Lambda180 = moisture_source(9,1); % параметр учитывающий рециркуляцию влаги
LambdaD = moisture_source(10,1);
Lambda170 = moisture_source(11,1);

% коэффициенты уравнения для расчета коэффициентов фракционирования пар-жидкость:
fract_coef = T{18:32,2};
A18_v_l = fract_coef(1,1); % пар - жидкость кислород 18
B18_v_l = fract_coef(2,1);
C18_v_l = fract_coef(3,1);
AD_v_l = fract_coef(4,1); % пар - жидкость дейтерий
BD_v_l = fract_coef(5,1);
CD_v_l = fract_coef(6,1);
AD_v_scl = fract_coef(8,1); % пар - переохлажденная жидкость дейтерий
CD_v_scl = fract_coef(9,1);
A18_v_i = fract_coef(10,1); % пар - лёд кислород 18
B18_v_i = fract_coef(11,1);
C18_v_i = fract_coef(12,1);
AD_v_i = fract_coef(13,1); % пар - лёд дейтерий
BD_v_i = fract_coef(14,1);
CD_v_i = fract_coef(15,1);

% степень, в которую нужно возвести alfa_18, чтобы получить alfa_17
n = fract_coef(7,1);

traject = T{36:41,2}; % параметры, описывающие траекторию
S = traject(1,1); % длина в км
E_d = traject(2,1); % высота воздушной массы в конце траектории в м

```

```

gam = traject(3,1);           % кривизна траектории
beta_E = traject(4,1);       % вертикальный градиент температуры
T_d = traject(5,1);          % температура конденсации в конце траектории, *C
p_sl = traject(6,1);         % давление на уровне моря в начале траектории, МПа
grav = 9.8;                  % ускорение силы тяжести
Ma = 0.0289;                 % молярная масса воздуха, кг/моль
Rg = 8.314;                  % газовая постоянная

cloud_physics = T{45:62,2};
L0 = cloud_physics(1,1);
Nu = cloud_physics(2,1);
sigma0 = cloud_physics(3,1);
T_w = cloud_physics(4,1);
T_i = cloud_physics(5,1);
Dif_180 = cloud_physics(16,1);
Dif_D = cloud_physics(17,1);
Dif_170 = cloud_physics(18,1);

Magnus = T{53:56,2};
A_w = Magnus(1,1);
B_w = Magnus(2,1);
A_i = Magnus(3,1);
B_i = Magnus(4,1);

```