

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Кафедра вычислительной техники

## **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**

Перевод чисел между системами счисления

Преподаватель

18.05.2016  
подпись, дата

Пушкарев К.В.  
инициалы, фамилия

Студент КИ15-08Б

18.05.2016  
подпись, дата

Войченко В.В.  
инициалы, фамилия

Красноярск 2016

## Цели работы

- I. Изучение позиционных систем счисления.
- II. Изучение алгоритма перевода чисел между позиционными системами счисления.
- III. Получение навыков работы со строками в MATLAB.

## Задания

- I. Написать функцию **num2dig(n)**, преобразующую целое число  $n$  от 0 до 35 в соответствующий по порядку символ из последовательности 0, 1, ..., 9, A, B, C, ..., Z.
- II. Написать функцию **dig2num(c)**, обратную к функции **num2dig()**.
- III. Написать функцию, представляющую действительное число в позиционной системе счисления (СС) с заданным основанием:  
**s = print\_num(n, radix, prec)**,  
где  $n$  -- представляемое число;  $radix$  -- основание СС;  $prec$  - максимальное количество знаков дробной части результата;  $s$  -- строка, представляющая число  $n$ .  
Указание: для записи цифр использовать знаки 0, 1, ..., 9, A, B, C, ..., Z, в качестве разделителя целой и дробной части использовать точку.
- IV. Написать функцию, обратную к функции **print\_num()**: **n = read\_num(s, radix)**.
- V. Сгенерировать в MATLAB 10 случайных действительных чисел в интервале от -1000 до 1000. Преобразовать их с помощью функции **print\_num()** в двоичную, троичную и шестнадцатеричную СС с 4 знаками после запятой. Получившиеся строки преобразовать обратно с помощью функции **read\_num()** и найти разность между результатом и исходным числом.

**Указание:** сгенерированные числа привести в отчёте в десятичной СС минимум с 6 знаками после запятой.

## Листинг функций

### num2dig()

```
function[char] = num2dig(num)
    symb_arr = [ 0 : 9 A : Z ];
    char = symb_arr(num + 1);

end
```

### dig2num()

```
function[num] = dig2num(char)
    symb_arr = [ 0 : 9 A : Z ];
    num = strfind(symb_arr, char) - 1;

end
```

### print\_num()

```
function[s_str] = print_num(n, radix, prec)
    if n < 0                                     % Если число отрицательное, то мы запоминаем знак -
        sign = - ;
        n = abs(n);
    else
        sign = ;
    end
    n_int = fix(n);
    n_div = n - n_int;                           % Смотрим, если ли целая часть у числа
    if n_int == 0
        s(1) = 0;
    else
        s = [];
    end
    while n_int > 0                               % Переводим целую часть
        s = [mod(n_int, radix), s];
        n_int = fix(n_int / radix);
    end
    if n_div > 0                                  % Переводим дробную часть
        s(end + 1) = -1;
        for i = 1:(prec + 1)
            n_div = n_div * radix;
            s = [s fix(n_div)];
            n_div = n_div - fix(n_div);
        end
    end
    s = num_round(s, radix);                     % Округляем, используя функцию num_round()
    s_str = ;                                     % Преобразование в строку
    for i = 1:numel(s)
        if s(i) == -1
            s_str(end + 1) = . ;
        else
            s_str(end + 1) = num2dig(s(i));
        end
    end
    s_str = [sign, s_str];

end
```

## num\_round()

```
function[s] = num_round(s, radix)
if s(end) >= fix(radix / 2) % Если последняя цифра в введенном числе больше половины СЧ, то
                           % увеличиваем предыдущий разряд на единицу
    i = numel(s);
    if s(end - 1) == -1
        s(end - 2) = s(end - 2) + 1;
        i = i - 2;
    else
        s(end - 1) = s(end - 1) + 1;
        i = i - 1;
    end
    while (s(i) == radix) % Если разряд числа максимальный, увеличиваем на единицу предыдущий
        if i == 1
            s = [0 s];
            i = 2;
        else
            s(i) = 0;
            if s(i - 1) == -1
                s(i - 2) = s(i - 2) + 1;
                i = i - 2;
            else
                s(i - 1) = s(i - 1) + 1;
                i = i - 1;
            end
        end
    end
end
end
s = s(1:end - 1);
end
```

### read\_num()

```
function[n] = read_num(s, radix)
    if s(1) == - % Если число отрицательное, то запоминаем его знак
        sign = -1;
        s = s(2:end);
    else
        sign = 1;
    end
    dp = strfind(s, '.'); % Поиск начальной дробной части
    if ~isempty(dp)
        s_int = s(1:(dp - 1));
        s_div = s((dp + 1):end);
    else
        s_int = s;
        s_div = '';
    end
    n_int = double(0); % Перевод целой части в систему счисления
    for i = 1:numel(s_int)
        n_int = n_int + dig2num(s_int(i)) * radix.^(numel(s_int) - i);
    end
    n_div = double(0); % Перевод дробной части в систему счисления
    for i = 1:numel(s_div)
        n_div = n_div + dig2num(s_div(i)) * radix.^(-i);
    end
    n = (n_int + n_div) * sign; % На выходе число, с сохранением знака исходного числа
end
```

### Input.m (скрипт, содержащий код лабораторной работы)

```
%% TASK 1
n = 28;
char = num2dig(n);
%% TASK 2
char = 'S';
n = dig2num(char);
%% TASK 3
n = 1506.2841;
s = print_num(n, 16, 4);
%% TASK 4
s = '1000100011';
n = read_num(s, 2);
%% TASK 5
P = 1000 * (2 * rand(1, 10) - 1);
arr_s = cell(3, 10);
for i = 1:10
    arr_s{1, i} = print_num(P(i), 2, 4);
    arr_s{2, i} = print_num(P(i), 3, 4);
    arr_s{3, i} = print_num(P(i), 16, 4);
end
arr_num = zeros(3, 10);

arr_error = zeros(3, 10);
for i = 1:10
    arr_num(1, i) = read_num(arr_s{1, i}, 2);
    arr_num(2, i) = read_num(arr_s{2, i}, 3);
    arr_num(3, i) = read_num(arr_s{3, i}, 16);
    arr_error(1, i) = arr_num(1, i) - P(i);
    arr_error(2, i) = arr_num(2, i) - P(i);
    arr_error(3, i) = arr_num(3, i) - P(i);
end
```

### Результаты работы программного кода Input.m:

| Вызов функции       | Имя, значение переменной |
|---------------------|--------------------------|
| char = num2dig (28) | char = 'S'               |
| n = dig2num(S)      | n = 28                   |

### Содержимое arr\_s:

| Двоичная СЧ      | Троичная СЧ   | Шестнадцатеричная СЧ |
|------------------|---------------|----------------------|
| 1001110101.0111  | 212022.1100   | 275.7287             |
| 1100101011.1001  | 1010001.1202  | 32B.9579             |
| -1011101010.0000 | -1000122.0002 | -2EA.06C0            |
| 1100111010.1100  | 1010121.2021  | 33A.C070             |
| 100001000.1011   | 100210.2011   | 108.B7EF             |
| -1100100100.1111 | -1002210.2210 | -324.EB50            |
| -110111011.0000  | -121102.0000  | -1BB.00E9            |
| 1011101.1100     | 10110.2022    | 5D.C356              |
| 1110010011.0000  | 1020220.0001  | 393.0380             |
| 1110100001.1100  | 1021102.2100  | 3A1.C6EE             |

**Содержимое вектора Р:**

|             |
|-------------|
| 629,447373  |
| 811,583874  |
| -746,026367 |
| 826,751712  |
| 264,718492  |
| -804,91919  |
| -443,003562 |
| 93,763038   |
| 915,013671  |
| 929,77707   |

**Содержимое вектора arr\_error:**

|           |           |          |           |           |           |          |           |           |           |
|-----------|-----------|----------|-----------|-----------|-----------|----------|-----------|-----------|-----------|
| -0,009873 | -0,021374 | 0,026367 | -0,001712 | -0,030992 | -0,018310 | 0,003562 | -0,013038 | -0,013671 | -0,02707  |
| -0,002928 | -0,003627 | 0,001676 | 0,001374  | -0,002443 | -0,006736 | 0,003562 | 0,002394  | -0,001325 | 0,000707  |
| 0         | 0,000003  | 0        | -0,000003 | -0,000002 | 0,000001  | 0,000007 | -0,000007 | 0,000001  | -0,000001 |

**Содержимое arr\_num:**

|            |            |             |            |            |             |             |           |            |            |
|------------|------------|-------------|------------|------------|-------------|-------------|-----------|------------|------------|
| 629,4375   | 811,5625   | -746        | 826,75     | 264,6875   | -804,9375   | -443        | 93,75     | 915        | 929,75     |
| 629,444444 | 811,580247 | -746,024691 | 826,753086 | 264,716049 | -804,925926 | -443        | 93,765432 | 915,012346 | 929,777778 |
| 629,447372 | 811,583878 | -746,026367 | 826,751709 | 264,718491 | -804,919189 | -443,003555 | 93,763031 | 915,013672 | 929,777069 |