

Алгоритмы - Системы счисления ^{И+ПРГ}

Перевод чисел из десятичной позиционной системы счисления в другую и наоборот

При переводе целого числа (целой части числа) из десятичной системы счисления в любую другую: исходное число (или целую часть) надо разделить на основание системы счисления, в которую выполняется перевод. Деление выполнять, пока частное не станет меньше основания новой системы счисления. Результат перевода определяется остатками от деления: первый остаток дает младшую цифру результирующего числа, последнее частное от деления дает старшую цифру.

Операция перевода в десятичную систему выглядит гораздо проще, так как любое десятичное число можно представить в виде

$$x = a_0 \cdot p^n + a_1 \cdot p^{n-1} + \dots + a_{n-1} \cdot p^1 + a_n \cdot p^0,$$

где $a_0 \dots a_n$ – это цифры данного числа в системе счисления с основанием p .

Например, перевести число 4A3F в десятичную систему.

По определению, $4A3F = 4 \cdot 16^3 + A \cdot 16^2 + 3 \cdot 16^1 + F \cdot 16^0$.

При замене A на 10, а F на 15, получается $4 \cdot 16^3 + 10 \cdot 16^2 + 3 \cdot 16 + 15 \cdot 1 = 19007$.

Алгоритмы - Системы счисления

Перевод из десятичной системы в другую (от 2-ой до 16-ой)

Выбор цифр систем счисления старше 10-тичной
в операторе множественного выбора **switch**

C/C++

```
// перевод из десятичной системы счисления в любую другую позиционную систему
#include <stdio.h>
#include <string.h>
#define Size 15

void main()
{   int na, a, i, j, k=0, osn;   int mas[Size];   char num[Size]={'\x0'};
    printf("\nВведите десятичное число: ");   scanf("%i", &na);   a=na;
    printf("\nВведите основание системы счисления: ");   scanf("%i", &osn);
    i=0;
    while (a!=0)
        {   mas[i]=a%osn;   i++;   a=(int)a/osn;   }
    for (j=i-1; j>=0; j--, k++)
        if (mas[j]<10)
            num[k]=(char)(mas[j]+48);
        else
            switch (mas[j]) {
                case 10: {num[k]='A'; break;}
                case 11: {num[k]='B'; break;}
                case 12: {num[k]='C'; break;}
                case 13: {num[k]='D'; break;}
                case 14: {num[k]='E'; break;}
                case 15: {num[k]='F'; break;}
            }
    printf("\nДесятичное число %d равно %d-ичному числу %s\n", na, osn, num);
}
```

Алгоритмы - Системы счисления

Перевод из десятичной системы в другую (от 2-ой до 16-ой)

Выбор цифр систем счисления от 2-ой до 16-ричной
с помощью строки, хранящей эти цифры

C/C++

```
// перевод из десятичной системы счисления в любую другую позиционную систему
#include <stdio.h>
#include <string.h>
#define Size 15

void main()
{ int na, a10, i=0, j, k=0, osn;
  char cifr[]="0123456789ABCDEF";
  int mas[Size];
  char num[Size]={'\x0'};
  printf("\nВведите десятичное число: ");
  scanf("%i", &na);
  a10=na;
  printf("\nВведите основание системы счисления: ");
  scanf("%i", &osn);
  while (a10!=0)
  {
    mas[i]=a10%osn;
    i++;
    a10=(int)a10/osn;
  }
  for (j=i-1; j>=0; j--, k++)
    num[k]=cifr[mas[j]];

  printf("\nДесятичное число %d равно %d-ичному числу %s\n", na, osn, num);
}
```

Алгоритмы - Системы счисления

Перевод из десятичной системы в другую (от 2-ой до 16-ой)

Выбор цифр систем счисления от 2-ой до 16-ричной
с помощью приращения кода ASCII

C/C++

// перевод из десятичной системы счисления в любую другую позиционную систему

```
#include <stdio.h>
#include <string.h>
#define Size 15

void main()
{ int na, a10, i=0, j, k=0, osn;
  int mas[Size];
  char num[Size]={'\x0'};
  printf("\nВведите десятичное число: ");
  scanf("%i", &na);
  a10=na;
  printf("\nВведите основание системы счисления: ");
  scanf("%i", &osn);
  while (a10!=0)
  {
    mas[i]=a10%osn;
    i++;
    a10=(int)a10/osn;
  }
  for (j=i-1; j>=0; j--, k++)
    if (mas[j] < 10)
      num[k]=(char)(mas[j]+48);
    else
      num[k]=(char)(mas[j]+55);
  printf("\nДесятичное число %d равно %d-ичному числу %s\n", na, osn, num);
}
```

Алгоритмы - Системы счисления ^{И+ПРГ}

Перевод чисел из десятичной позиционной системы счисления в другую и наоборот

При переводе целого числа (целой части числа) из десятичной системы счисления в любую другую: исходное число (или целую часть) надо разделить на основание системы счисления, в которую выполняется перевод. Деление выполнять, пока частное не станет меньше основания новой системы счисления. Результат перевода определяется остатками от деления: первый остаток дает младшую цифру результирующего числа, последнее частное от деления дает старшую цифру.

Операция перевода в десятичную систему выглядит гораздо проще, так как любое десятичное число можно представить в виде

$$x = a_0 \cdot p^n + a_1 \cdot p^{n-1} + \dots + a_{n-1} \cdot p^1 + a_n \cdot p^0,$$

где $a_0 \dots a_n$ – это цифры данного числа в системе счисления с основанием p .

Например, перевести число 4A3F в десятичную систему.

По определению, $4A3F = 4 \cdot 16^3 + A \cdot 16^2 + 3 \cdot 16^1 + F \cdot 16^0$.

При замене A на 10, а F на 15, получается $4 \cdot 16^3 + 10 \cdot 16^2 + 3 \cdot 16 + 15 \cdot 1 = 19007$.

Алгоритмы - Системы счисления И+ПРГ

Перевод из систем с основанием от 2-ой до 16-ой в десятичную

Выбор цифр систем счисления от 2-ой до 16-ричной
в операторе множественного выбора **switch**

C/C++

```
#include <stdio.h> // перевод числа из системы счисления от 2-х до 16-и в десятичную
#include <string.h>
main()
{ int osn, rez, st, i;
  /* osn - основание системы счисления переводимого числа,
   rez - результат перевода: число в десятичной системе счисления,
   st - основание системы счисления в степени текущей позиции цифры в числе */
  char str[20]; // строка содержащая исходное число в системе счисления от 2-х до 16-и
  printf("Перевод числа в 10-ичную систему счисления. \n");
  printf("Введите число "); gets(str);
  printf("\nВведите основание системы счисления: "); scanf("%i", &osn);
  st=1; rez=0; i=strlen(str)-1;
  while (i>=0)
  { if ((int(str[i])-48)<10)
    rez=int(str[i]-48)*st+rez;
    else switch (str[i]) {
      case 'A': case 'a': {rez=10*st+rez; break;}
      case 'B': case 'b': {rez=11*st+rez; break;}
      case 'C': case 'c': {rez=12*st+rez; break;}
      case 'D': case 'd': {rez=13*st+rez; break;}
      case 'E': case 'e': {rez=14*st+rez; break;}
      case 'F': case 'f': {rez=15*st+rez; break;}
    }
    st=st*osn; i--;
  }
  printf("%i", rez); return 0;
}
```

Алгоритмы - Системы счисления ^{И+ПРГ}

Перевод из систем с основанием от 2-ой до 16-ой в десятичную

Выбор цифр систем счисления от 2-ой до 16-ричной
с помощью строки, хранящей эти цифры

C/C++

```
#include <stdio.h> // перевод из любой системы счисления в десятичную (цифры из строки)
#include <string.h>
main()
{
    int osn, rez=0, st=1, i, k=0;
    char cifr[]="0123456789ABCDEF";
    int mas[Size];
    char str[20]; // строка содержащая исходное число в системе счисления от 2-х до 16-и
    printf("Перевод числа в 10-ичную систему счисления. \n");
    printf("Введите число для перевода - "); gets(str);
    printf("\nВведите основание системы счисления: "); scanf("%i", &osn);

    for (i=strlen(str)-1; i>=0; i--)
    {
        for (k=0; k<strlen(cifr); k++)
        {
            if (toupper(str[i])==cifr[k])
            { rez=rez+k*st; break; }
        }
        st=st*osn;
    }
    printf("%i", rez);
    return 0;
}
```

Алгоритмы - Системы счисления ^{И+ПРГ}

Перевод из систем с основанием от 2-ой до 16-ой в десятичную

Выбор цифр систем счисления от 2-ой до 16-ричной
с помощью приращения кода ASCII

C/C++

```
#include <stdio.h> // перевод из любой системы счисления в десятичную (цифры из строки)
#include <string.h>
#include <ctype.h>
#define Size 15
main()
{
    int osn, rez, st, i;
    char str[20]; // строка содержащая исходное число в системе счисления от 2-й до 16-и
    printf("Перевод числа в 10-ичную систему счисления. \n");
    printf("Введите число для перевода - ");    gets(str);
    printf("\nВведите основание системы счисления: ");    scanf("%i", &osn);
    st=1;    rez=0;
    i=strlen(str)-1;
    while (i>=0)
        { if ((int(str[i])-48)<10)
            rez=int(str[i]-48)*st+rez;
          else
            { str[i]=toupper(str[i]);
              rez=int(str[i]-55)*st+rez; }
          }
        st=st*osn;
        i--;
    }
    printf("%i", rez);
    return 0;
}
```