



# C语言 进制转换

## 1-常见进制

进制	名称	基数	可用数字
十进制	Decimal	10	0-9
二进制	Binary	2	0、1
四进制	Quaternary	4	0-3
八进制	Octal	8	0-7
十六进制	Hexadecimal	16	0-9、A-F

## 2-任意进制向十进制转换

**原理：按位展开法（位权）**

从右往左，第 0 位开始

每一位  $\times (\text{进制})^{\text{位数}}$ ，然后求和

### 1、二进制 $\rightarrow$ 十进制

$(10110)_2$

位	数字	计算
$2^4$	1	$1 \times 16$
$2^3$	0	$0 \times 8$
$2^2$	1	$1 \times 4$
$2^1$	1	$1 \times 2$
$2^0$	0	$0 \times 1$

$$= 16 + 4 + 2 = 22$$

```

1  #include <stdio.h>
2
3  int main()
4  {
5      char bin[32];
6      int i = 0;
7      int result = 0;
8
9      scanf("%s", bin);    // 输入二进制字符串
10
11     while (bin[i] != '\0')
12     {
13         result = result * 2 + (bin[i] - '0');
14         i++;
15     }
16
17     printf("%d\n", result);
18     return 0;
19 }

```



## 2、 四进制 → 十进制

$$(132)_4$$

$$= 1 \times 16 + 3 \times 4 + 2 = 30$$

```

1  #include <stdio.h>
2
3  int main()
4  {
5      char base4[32];
6      int i = 0;
7      int result = 0;
8
9      scanf("%s", base4);    // 输入四进制数
10
11     while (base4[i] != '\0')
12     {
13         result = result * 4 + (base4[i] - '0');
14         i++;
15     }
16
17     printf("%d\n", result);
18     return 0;
19 }
20

```



### 3、八进制 → 十进制

$(745)_8$

$$= 7 \times 8^2 + 4 \times 8^1 + 5 \times 8^0$$

$$= 448 + 32 + 5 = 485$$

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char oct[32];
6      int i = 0;
7      int result = 0;
8
9      scanf("%s", oct); // 输入八进制数
10
11     while (oct[i] != '\0')
12     {
13         result = result * 8 + (oct[i] - '0');
14         i++;
15     }
16
17     printf("%d\n", result);
18     return 0;
19 }
```



C:\WINDOWS\sys  
745  
485  
请按任意键继续

### 4、十六进制 → 十进制

$(3AF)_{16}$

$$= 3 \times 16^2 + 10 \times 16^1 + 15 \times 16^0$$

$$= 768 + 160 + 15 = 943$$

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char hex[32];
6      int i = 0;
7      int result = 0;
8      int value;
9
10     scanf("%s", hex); // 输入十六进制数 (如 3AF)
11
12     while (hex[i] != '\0')
13     {
14         if (hex[i] >= '0' && hex[i] <= '9')
15             value = hex[i] - '0';
16         else
17             value = hex[i] - 'A' + 10;
18
19         result = result * 16 + value;
20         i++;
21     }
22
23     printf("%d\n", result);
24     return 0;
25 }
```



C:\WINDOWS\system32\cr  
3AF  
943  
请按任意键继续 . . .

# 3-十进制向任意进制转换

## 原理：除基取余法

除以进制 记录余数 直到商为0 余数倒着写

### 1、十进制 $\rightarrow$ 二进制

$(45)_{10}$

除 2	商	余
$45 \div 2$	22	1
$22 \div 2$	11	0
$11 \div 2$	5	1
$5 \div 2$	2	1
$2 \div 2$	1	0
$1 \div 2$	0	1

👉 逆序:

$(101101)_2$

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n;
6      int a[32];
7      int i = 0;
8
9      scanf("%d", &n);    // 输入十进制数
10
11     while (n > 0)
12     {
13         a[i] = n % 2;
14         n = n / 2;
15         i++;
16     }
17
18     for (i = i - 1; i >= 0; i--)
19         printf("%d", a[i]);
20
21     return 0;
22 }
```

## 2、十进制 → 二进制

$(50)_{10}$

除 4	商	余
$50 \div 4$	12	2
$12 \div 4$	3	0
$3 \div 4$	0	3

$(302)_4$

```
1  #include <stdio.h>
2  int main()
3  {
4      int n;
5      int a[32];
6      int i = 0;
7
8      scanf("%d", &n);    // 输入十进制数
9
10     while (n > 0)
11     {
12         a[i] = n % 4;
13         n = n / 4;
14         i++;
15     }
16
17     for (i = i - 1; i >= 0; i--)
18         printf("%d", a[i]);
19
20     return 0;
21 }
```

### 3、十进制 → 八进制

$(83)_{10}$

除 8	商	余
$83 \div 8$	10	3
$10 \div 8$	1	2
$1 \div 8$	0	1

$(123)_8$

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n;
6      int a[32];
7      int i = 0;
8
9      scanf("%d", &n);    // 输入十进制数
10
11     while (n > 0)
12     {
13         a[i] = n % 8;
14         n = n / 8;
15         i++;
16     }
17
18     for (i = i - 1; i >= 0; i--)
19         printf("%d", a[i]);
20
21     return 0;
22 }
```

## 4、十进制 → 十六进制

$(254)_{10}$

除 16	商	余
254÷16	15	14(E)
15÷16	0	15(F)

$(FE)_{16}$

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n;
6      int a[32];
7      int i = 0;
8
9      scanf("%d", &n);    // 输入十进制数
10
11     while (n > 0)
12     {
13         a[i] = n % 16;
14         n = n / 16;
15         i++;
16     }
17
18     for (i = i - 1; i >= 0; i--)
19     {
20         if (a[i] < 10)
21             printf("%d", a[i]);
22         else
23             printf("%c", a[i] - 10 + 'A');
24     }
25
26     return 0;
27 }
```