**子网掩码**

#include <stdio.h>

#include <stdint.h>

// 定义包含四个位字段的IP类型，每个字段表示8位

typedef struct

{

int A;

int B;

int C;

int D;

} IP;

int countOne(unsigned int n) {

int count = 0;

while (n > 0) {

count += n & 1;

n >>= 1;

}

return count;

}

// 递归计算IP变量中1的位数

int countBits(IP ip)

{

int count = 0;

// 统计每个8位字段中1的个数

count += countOne(ip.A);

count += countOne(ip.B);

count += countOne(ip.C);

count += countOne(ip.D);

return count;

}

int main()

{

// 从标准输入读取IP地址和子网掩码

IP ad;

IP msk;

IP ad2;

scanf("%d.%d.%d.%d", &ad.A, &ad.B, &ad.C, &ad.D);

scanf("%d.%d.%d.%d", &msk.A, &msk.B, &msk.C, &msk.D);

// 通过按位与运算计算网络地址

ad2.A = ad.A & msk.A;

ad2.B = ad.B & msk.B;

ad2.C = ad.C & msk.C;

ad2.D = ad.D & msk.D;

// 打印结果

/\*

printf("%d.%d.%d.%d\n", ad2.A, ad2.B, ad2.C, ad2.D);

printf("%d.%d.%d.%d\n", ad.A, ad.B, ad.C, ad.D);

printf("%d.%d.%d.%d\n", msk.A, msk.B, msk.C, msk.D);

\*/

// 递归计算网络地址中1的位数

int res = countBits(ad2);

// 打印结果到标准输出

printf("%d\n", res);

return 0;

}

|  |  |
| --- | --- |
| Input | Output |
| 164.125.252.57  255.255.0.0（子网掩码） | 9 |
| 125.252.57.164  255.0.0.0 | 6 |