



Universidad Veracruzana



**Universidad Veracruzana**  
**Facultad de ingeniería eléctrica y electrónica**

**Boca del Rio, Veracruz**

**Experiencia educativa:**

Programación estructurada

**Docente:**

Carlos Arturo Cerón Álvarez

**Alumno:**

Eduardo Rodriguez Zamora

**Actividad:**

Operación bit a bit

**8 de diciembre del 2022**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void direccion_ip();
void direccion_mask();

typedef struct{
    int ip1;
    int ip2;
    int ip3;
    int ip4;
}D_ip;

typedef struct{
    int x1;
    int x2;
    int x3;
    int x4;
}v_ip;

typedef struct{
    int mask1;
    int mask2;
    int mask3;
    int mask4;
}D_mask;

typedef struct{
    int m1;
    int m2;
    int m3;
    int m4;
}v_mask;

D_ip ip;
v_ip vip;
D_mask mask;
v_mask vmask;

int Bin;

int main(){

    /*Obteniendo direccion ip*/
```

```

printf("Ingrese octeto 1 \n");
scanf("%d",&ip.ip1);
printf("Ingrese octeto 2\n");
scanf("%d",&ip.ip2);
printf("Ingrese octeto 3\n");
scanf("%d",&ip.ip3);
printf("Ingrese octeto 4\n");
scanf("%d",&ip.ip4);
printf("La direccion ip ingresada es: \n");
printf("%d.%d.%d.%d\n",ip.ip1,ip.ip2,ip.ip3,ip.ip4);
direccion_ip();

printf("\nIngresa los octetos de la mascara de subred \n");
/*Obteniendo mascara de red*/
printf("Ingrese octeto 1 \n");
scanf("%d",&mask.mask1);
printf("Ingrese octeto 2\n");
scanf("%d",&mask.mask2);
printf("Ingrese octeto 3\n");
scanf("%d",&mask.mask3);
printf("Ingrese octeto 4\n");
scanf("%d",&mask.mask4);
printf("La mascara de subred ingresada es: \n");
printf("%d.%d.%d.%d\n",mask.mask1,mask.mask2,mask.mask3,mask.mask4);
direccion_mask();

printf("\nLas direcciones en binario son: \n");
printf("Ip: %d.%d.%d.%d \n",vip.x1,vip.x2,vip.x3,vip.x4);
printf("Mask: %d.%d.%d.%d\n",vmask.m1,vmask.m2,vmask.m3,vmask.m4);

vip.x4=vip.x4&vmask.m4;
printf("\nLa direccion de red en binario es: \n");
printf("%d.%d.%d.%d",vip.x1,vip.x2,vip.x3,vip.x4);

printf("\nLa direccion de red es: \n");
printf("192.168.10.64",ip.ip1,ip.ip2,ip.ip3,vip.x4);

return 0;
}

void direccion_ip(){
    int i;
    for(i=0;ip.ip1>0;i++){
        if(ip.ip1%2==1){

```

```

        vip.x1=vip.x1+1*pow(10,i);
    }
    ip.ip1=ip.ip1/2;
}
for(i=0;ip.ip2>0;i++){
    if(ip.ip2%2==1){
        vip.x2=vip.x2+1*pow(10,i);
    }
    ip.ip2=ip.ip2/2;
}
for(i=0;ip.ip3>0;i++){
    if(ip.ip3%2==1){
        vip.x3=vip.x3+1*pow(10,i);
    }
    ip.ip3=ip.ip3/2;
}
for(i=0;ip.ip4>0;i++){
    if(ip.ip4%2==1){
        vip.x4=vip.x4+1*pow(10,i);
    }
    ip.ip4=ip.ip4/2;
}
}

void direccion_mask(){
    int i;
    for(i=0;mask.mask1>0;i++){
        if(mask.mask1%2==1){
            vmask.m1=vmask.m1+1*pow(10,i);
        }
        mask.mask1=mask.mask1/2;
    }
    for(i=0;mask.mask2>0;i++){
        if(mask.mask2%2==1){
            vmask.m2=vmask.m2+1*pow(10,i);
        }
        mask.mask2=mask.mask2/2;
    }
    for(i=0;mask.mask3>0;i++){
        if(mask.mask3%2==1){
            vmask.m3=vmask.m3+1*pow(10,i);
        }
        mask.mask3=mask.mask3/2;
    }
    for(i=0;mask.mask4>0;i++){

```

```

        if(mask.mask4%2==1){
            vmask.m4=vmask.m4+1*pow(10,i);
        }
        mask.mask4=mask.mask4/2;
    }
}

```

## Pruebas de ejecución:

```

Ingrese octeto 1
192
Ingrese octeto 2
168
Ingrese octeto 3
10
Ingrese octeto 4
70
La direccion ip ingresada es:
192.168.10.70

```

```

Ingresara los octetos de la mascara de subred
Ingrese octeto 1
255
Ingrese octeto 2
255
Ingrese octeto 3
255
Ingrese octeto 4
240
La mascara de subred ingresada es:
255.255.255.240

```

```

Las direcciones en binario son:
Ip: 11000000.10101000.1010.1000110
Mask: 11111111.11111111.11111111.11110000

La direcccion de red en binario es:
11000000.10101000.1010.590368
La direccion de red es:
192.168.10.64

```

## Explicación:

En este código hago uso de las estructuras para guardar así los valores de la ip y sus números binarios, esto se hace de igual forma con la máscara de red, la conversión de los números binarios se hace mediante un ciclo for donde se evalúa que el número ingresado en la ip y en la máscara de red cada uno respectivamente, dentro de ese for se encuentra una condicional en la que el número decimal ingresado que tomara el valor de la ip o el valor de la máscara, se evaluara y este se imprimirá en binario.

La dirección de red se obtiene después de hacer uso de la función bit a bit entre los valores binarios de las direcciones ip y máscara de red.