



## Práctica 1.-Calculadora IP

Objetivo: el alumno programará una calculadora IP versión 1 en C. Haciendo uso de un block de notas y compilando en gcc

Dada una direccion IP (inicializada en hexadecimal)

Output: Clase, Dirección IP de Red y dirección de Broadcast.

Las IP de prueba son las siguientes

	Dirección IP	Dir en hexadecimal	Clase	Dir. IP Red	Broadcast
<b>Z</b>	10.250.1.1	0x0A, 0xFA, 0x01, 0x01	<b>A</b>	<b>10.0.0.0</b>	<b>10.255.255.255</b>
<b>Y</b>	150.10.15.0	0x96, 0x0A, 0x0F, 0x00	<b>B</b>	<b>150.10.0.0</b>	<b>150.10.255.255</b>
<b>X</b>	192.14.2.0	0XC0, 0X0E, 0X02, 0x00	<b>C</b>	<b>192.14.2.0</b>	<b>192.14.2.255</b>
<b>W</b>	148.17.9.1	0X94, 0X11, 0X09, 0X01	<b>B</b>	<b>148.17.0.0</b>	<b>148.17.255.255</b>
<b>V</b>	193.0.0.255	0XC1, 0X00, 0X00, 0XFF	<b>C</b>	<b>193.0.0.0</b>	<b>193.0.0.255</b>
<b>U</b>	220.200.23.1	0xDC, 0xC8, 0x17, 0x01	<b>C</b>	<b>220.200.23.0</b>	<b>220.200.23.255</b>
<b>T</b>	177.100.18.4	0xB1, 0x64, 0x12, 0x04	<b>B</b>	<b>177.100.0.0</b>	<b>177.100.255.255</b>
<b>S</b>	95.0.21.0	0x5F, 0x00, 0x15, 0x00	<b>A</b>	<b>95.0.0.0</b>	<b>95.255.255.255</b>
<b>R</b>	111.111.111.111	0x6F, 0x6F, 0x6F, 0x6F	<b>B</b>	<b>111.111.0.0</b>	<b>111.111.255.255</b>
<b>Q</b>	219.26.51.0	0xDB, 0x1A, 0x33, 0x00	<b>C</b>	<b>219.26.51.0</b>	<b>219.26.51.255</b>
<b>P</b>	167.0.0.0	0xA7, 0x00, 0x00, 0x00	<b>B</b>	<b>167.0.0.0</b>	<b>167.0.255.255</b>
<b>N</b>	10.1.1.1	0x0A, 0x01, 0x01, 0x01	<b>A</b>	<b>10.0.0.0</b>	<b>10.255.255.255</b>
<b>M</b>	225.1.1.1	0xE1, 0x01, 0x01, 0x01	<b>D</b>		
<b>O</b>	240.1.1.1	0xF0, 0x01, 0x01, 0x01	<b>E</b>		



## Práctica 1.-Calculadora IP

Código:

```
• #include<stdio.h>
•
• void ImprimeIPR(unsigned char *, unsigned char *);
• void ImprimeIPB(unsigned char *, unsigned char *);
• void MuestraClaseIPRB(unsigned char *, unsigned char *);
•
• int main(){
•     unsigned char MR[4] = {0,0,0,0};
•     unsigned char IP[4] = {0x0A, 0xFA, 0x01, 0x01}; //IP[4] = {10,250,1,1}
•     //unsigned char IP[4] = {0x96, 0x0A, 0x0F, 0x00}; //IP[4] = {150,10,15,0};
•     //unsigned char IP[4] = {0xC0, 0x0E, 0x02, 0x00}; //IP[4] = {192,14,2,0};
•     //unsigned char IP[4] = {0x94, 0x11, 0x09, 0x01}; //IP[4] = {148,17,9,1};
•     //unsigned char IP[4] = {0xC1, 0x00, 0x00, 0xFF}; //IP[4] = {193,0,0,255};
•     //unsigned char IP[4] = {0xDC, 0xC8, 0x17, 0x01}; //IP[4] = {220,200,23,1};
•     //unsigned char IP[4] = {0xB1, 0x64, 0x12, 0x04}; //IP[4] = {177,100,18,4};
•     //unsigned char IP[4] = {0x5F, 0x00, 0x15, 0x00}; //IP[4] = {95,0,21,0};
•     //unsigned char IP[4] = {0x6F, 0x6F, 0x6F, 0x6F}; //IP[4] = {111,111,111,111};
•     //unsigned char IP[4] = {0xDB, 0x1A, 0x33, 0x00}; //IP[4] = {219,26,51,0};
•     //unsigned char IP[4] = {0xA7, 0x00, 0x00, 0x00}; //IP[4] = {167,0,0,0};
•     //unsigned char IP[4] = {0x0A, 0x01, 0x01, 0x01}; //IP[4] = {10,1,1,1};
•     //unsigned char IP[4] = {0xE1, 0x01, 0x01, 0x01}; //IP[4] = {225,1,1,1};
•     //unsigned char IP[4] = {0xF0, 0x01, 0x01, 0x01}; //IP[4] = {240,1,1,1};
•
•     MuestraClaseIPRB(MR, IP);
•
•     return 0;
• }
•
• void ImprimeIPR(unsigned char *MR, unsigned char *IP){
•     printf("La direccion IP de RED es: %d.%d.%d.%d\n\n", IP[0]&MR[0], IP[1]&MR[1], IP[2]&MR[2],
IP[3]&MR[3]);
• }
•
• void ImprimeIPB(unsigned char *MR, unsigned char *IP){
•     printf("La direccion IP de BROADCAST es: %d.%d.%d.%d\n\n", (((unsigned char)IP[0])|((unsigned
char)~MR[0])),
•     (((unsigned char)IP[1])|((unsigned char)~MR[1])), (((unsigned char)IP[2])|((unsigned
char)~MR[2])), (((unsigned char)IP[3])|((unsigned char)~MR[3])));
• }
```

M. en C. Nidia A. Cortez Duarte



## Práctica 1.-Calculadora IP

```
• }  
•  
• void MuestraClaseIPRB(unsigned char *MR, unsigned char *IP){  
•  
•     if(IP[0]&128){  
•         if(IP[0]&64){  
•             if(IP[0]&32){  
•                 if(IP[0]&16)  
•                     printf("\nLa direccion IP pertenece a la Clase E.\n\n");  
•                 else  
•                     printf("\nLa direccion IP pertenece a la Clase D.\n\n");  
•             }else{  
•                 MR[0] = 255;  
•                 MR[1] = 255;  
•                 MR[2] = 255;  
•                 printf("\nLa direccion IP pertenece a la Clase C.\n\n");  
•                 ImprimeIPR(MR, IP);  
•                 ImprimeIPB(MR, IP);  
•             }  
•         }else{  
•             MR[0] = 255;  
•             MR[1] = 255;  
•             printf("\nLa direccion IP pertenece a la Clase B.\n\n");  
•             ImprimeIPR(MR, IP);  
•             ImprimeIPB(MR, IP);  
•         }  
•     }else{  
•         MR[0] = 255;  
•         printf("\nLa direccion IP pertenece a la Clase A.\n\n");  
•         ImprimeIPR(MR, IP);  
•         ImprimeIPB(MR, IP);  
•     }  
• }
```



## Práctica 1.-Calculadora IP

**Mapa de memoria** utilizado en el programa (considerando registros de 8 bits)

Unsigned char IP[0]	0	0	0	0	1	0	1	0
	1	1	1	1	1	0	1	0
	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	1
Unsigned char MR[0]	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

## Conclusiones

Normalmente necesitamos ser capaces de programar cosas condicionalmente, es decir, si esta cosa es verdad, entonces haz X, pero si esta otra cosa es verdad, entonces haz Y. En nuestros programas es posible conseguir esto al utilizar condicionales if/else. Básicamente, una declaración if le dice al programa que ejecute un bloque de código, si una condición es 0 o un valor diferente de 0, durante el desarrollo de esta práctica esto fue posible observarlo de una mejor manera ya que al utilizar un operador binario el resultado podía ser 0 o 128 y no necesariamente 0 o 1 como comúnmente lo utilizamos, lo cual fue de gran utilidad para optimizar los recursos que el programa necesita para ejecutarse, al utilizar un if/else no todo lo que programemos se va a ejecutar, ya que habrá partes del código que se ejecuten o no, por lo tanto, el if/else como falso y verdadero es extremadamente útil pues permite definirle a nuestro programa múltiples vías de ejecución contemplando así todas las posibilidades.

Comparar el costo computacional entre.

```
If(IP[0]&128)
Else
    "clase A"
```

Contra

```
If (IP[0]>0 && IP[0]<127)
    "clase A"
```

El primer bloque de código tiene un menor costo computacional puesto que solo hace una comparación a nivel de bits mientras que el segundo bloque de código necesita hacer más comparaciones para



## Práctica 1.-Calculadora IP

asegurar que se encuentre o no en el rango establecido y esto implica un costo computacional mucho mayor.

¿Habías usado los operadores binarios?

Anteriormente ya había utilizado operadores binarios en la materia de Estructura de datos donde es necesario utilizar operadores binarios para implementar algunas funciones, además en teoría computacional en un programa para comprobar si un autómata acepta una cadena, en análisis de algoritmos para implementar de una mejor manera los algoritmos y en programación orientada a objetos de igual manera para implementar funciones.

¿Qué opinas del uso de operadores binarios?

Un operador binario es muy importante, ya que trabaja sobre números binarios a nivel de sus bits individuales, esto es una acción primitiva pero bastante rápida y eficaz, soportada directamente por los procesadores. Entonces en procesadores simples con un rendimiento bajo, las operaciones a nivel de bits, junto con los de desplazamiento, son normalmente más rápidas que la multiplicación y la división convencionales, lo cual resulta muy útil para la implementación de programas sin utilizar muchos recursos, mientras que en los modernos procesadores de alto rendimiento usualmente las operaciones se realizan a la misma velocidad y puede parecer que su uso no es relevante, sin embargo, siempre debemos optimizar los recursos que usamos en nuestros programas y los operadores binarios son de gran ayuda para esto.

## Capturas

```
C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
La direccion IP pertenece a la Clase A.
La direccion IP de RED es: 10.0.0.0
La direccion IP de BROADCAST es: 10.255.255.255

C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
La direccion IP pertenece a la Clase B.
La direccion IP de RED es: 150.10.0.0
La direccion IP de BROADCAST es: 150.10.255.255
```



## Práctica 1.-Calculadora IP

```
C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
```

```
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
```

La direccion IP pertenece a la Clase C.

La direccion IP de RED es: 192.14.2.0

La direccion IP de BROADCAST es: 192.14.2.255

```
C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
```

```
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
```

La direccion IP pertenece a la Clase B.

La direccion IP de RED es: 148.17.0.0

La direccion IP de BROADCAST es: 148.17.255.255

```
C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
```

```
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
```

La direccion IP pertenece a la Clase C.

La direccion IP de RED es: 193.0.0.0

La direccion IP de BROADCAST es: 193.0.0.255

```
C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
```

```
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
```

La direccion IP pertenece a la Clase C.

La direccion IP de RED es: 220.200.23.0

La direccion IP de BROADCAST es: 220.200.23.255

```
C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
```

```
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
```

La direccion IP pertenece a la Clase B.

La direccion IP de RED es: 177.100.0.0

La direccion IP de BROADCAST es: 177.100.255.255



## Práctica 1.-Calculadora IP

```
C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
La direccion IP pertenece a la Clase A.
La direccion IP de RED es: 95.0.0.0
La direccion IP de BROADCAST es: 95.255.255.255

C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
La direccion IP pertenece a la Clase A.
La direccion IP de RED es: 111.0.0.0
La direccion IP de BROADCAST es: 111.255.255.255

C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
La direccion IP pertenece a la Clase C.
La direccion IP de RED es: 219.26.51.0
La direccion IP de BROADCAST es: 219.26.51.255
```

```
C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
La direccion IP pertenece a la Clase B.
La direccion IP de RED es: 167.0.0.0
La direccion IP de BROADCAST es: 167.0.255.255
```



## Práctica 1.-Calculadora IP

```
C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
La direccion IP pertenece a la Clase A.
La direccion IP de RED es: 10.0.0.0
La direccion IP de BROADCAST es: 10.255.255.255

C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
La direccion IP pertenece a la Clase D.

C:\Users\braya\Downloads\Redes\Practica 1>gcc Pract2.c -o Pract2
C:\Users\braya\Downloads\Redes\Practica 1>Pract2
La direccion IP pertenece a la Clase E.
```





## Práctica 1.-Calculadora IP

Pract2: Bloc de notas

Archivo Edición Formato Ver Ayuda

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

```
void ImprimeIPR(unsigned char *, unsigned char *);  
void ImprimeIPB(unsigned char *, unsigned char *);  
void MuestraClaseIPRB(unsigned char *, unsigned char *);
```

```
int main(){  
    unsigned char MR[4] = {0,0,0,0};  
    //unsigned char IP[4] = {0x0A, 0xFA, 0x01, 0x01}; //IP[4] = {10,250,1,1}  
    //unsigned char IP[4] = {0x96, 0x0A, 0x0F, 0x00}; //IP[4] = {150,10,15,0};  
    //unsigned char IP[4] = {0xC0, 0x0E, 0x02, 0x00}; //IP[4] = {192,14,2,0};  
    //unsigned char IP[4] = {0x94, 0x11, 0x09, 0x01}; //IP[4] = {148,17,9,1};  
    //unsigned char IP[4] = {0xC1, 0x00, 0x00, 0xFF}; //IP[4] = {193,0,0,255};  
    //unsigned char IP[4] = {0xDC, 0xC8, 0x17, 0x01}; //IP[4] = {220,200,23,1};  
    //unsigned char IP[4] = {0xB1, 0x64, 0x12, 0x04}; //IP[4] = {177,100,18,4};  
    //unsigned char IP[4] = {0x5F, 0x00, 0x15, 0x00}; //IP[4] = {95,0,21,0};  
    //unsigned char IP[4] = {0x6F, 0x6F, 0x6F, 0x6F}; //IP[4] = {111,111,111,111};  
    //unsigned char IP[4] = {0xDB, 0x1A, 0x33, 0x00}; //IP[4] = {219,26,51,0};  
    //unsigned char IP[4] = {0xA7, 0x00, 0x00, 0x00}; //IP[4] = {167,0,0,0};  
    //unsigned char IP[4] = {0x0A, 0x01, 0x01, 0x01}; //IP[4] = {10,1,1,1};  
    //unsigned char IP[4] = {0xE1,0x01, 0x01, 0x01}; //IP[4] = {225,1,1,1};  
    unsigned char IP[4] = {0xF0, 0x01, 0x01, 0x01}; //IP[4] = {240,1,1,1};  
  
    MuestraClaseIPRB(MR, IP);  
  
    return 0;  
}  
  
void ImprimeIPR(unsigned char *MR, unsigned char *IP){  
    printf("La direccion IP de RED es: %d.%d.%d.%d\n", IP[0]&MR[0], IP[1]&MR[1], IP[2]&MR[2], IP[3]&MR[3]);  
}  
  
void ImprimeIPB(unsigned char *MR, unsigned char *IP){  
    printf("La direccion IP de BROADCAST es: %d.%d.%d.%d\n", (((unsigned char)IP[0])|((unsigned char)~MR[0])),  
        (((unsigned char)IP[1])|((unsigned char)~MR[1])),(((unsigned char)IP[2])|((unsigned char)~MR[2])),(((unsigned char)IP[3])|((unsigned char)~MR[3])));  
}  
  
void MuestraClaseIPRB(unsigned char *MR, unsigned char *IP){  
    if(IP[0]&128){  
        if(IP[0]&64){  
            if(IP[0]&32){  
                if(IP[0]&16)  
                    printf("\nLa direccion IP pertenece a la Clase E.\n");  
                else  
                    printf("\nLa direccion IP pertenece a la Clase D.\n");  
            }else{  
                MR[0] = 255;  
                MR[1] = 255;  
                MR[2] = 255;  
                printf("\nLa direccion IP pertenece a la Clase C.\n");  
                ImprimeIPR(MR, IP);  
                ImprimeIPB(MR, IP);  
            }  
        }else{  
            MR[0] = 255;  
            MR[1] = 255;  
            printf("\nLa direccion IP pertenece a la Clase B.\n");  
            ImprimeIPR(MR, IP);  
            ImprimeIPB(MR, IP);  
        }  
    }else{  
        MR[0] = 255;  
        printf("\nLa direccion IP pertenece a la Clase A.\n");  
        ImprimeIPR(MR, IP);  
        ImprimeIPB(MR, IP);  
    }  
}
```



## Práctica 1.-Calculadora IP

Criterio	Valor	Tu evaluación
El programa se escribe en un block de notas y se compila con gcc	1	1
Se incluye todo el código y se hace uso de operadores binarios	1	1
El mapa de memoria refleja todas las variables utilizadas en el programa y han sido seleccionadas de forma consciente.	1	1
Las imágenes son claras	1	1
Se incluyen las conclusiones solicitadas	1	1
<b>TOTAL</b>	<b>5</b>	<b>5</b>