



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

Dada una dirección IP (inicializada en hexadecimal)

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

Las IP de prueba son las siguientes, favor de llenar la tabla



	Dirección IP	Dir en hexadecimal	Clase	Dir. IP Red	Broadcast
Z	10.250.1.1	0x0A.0xFA.0x01.0x01	A	10.0.0.0	10.255.255.255
Y	150.10.15.0	0x96 0x0A,0x0F 0x00	B	150.10.0.0	150.10.255.255
X	192.14.2.0	c0.e.2.0	C	192.14.2.0	192.14.2.255
W	148.17.9.1	94.11.9.1	B	148.17.0.0	148.17.255.255
V	193.0.0.255	c1.0.0.ff	C	193.0.0.0	193.0.0.255
U	220.200.23.1	dc.c8.17.1	C	220.200.23.0	220.200.23.255
T	177.100.18.4	b1.64.12.4	B	177.100.0.0	177.100.255.255
S	95.0.21.0	5f.0.15.0	A	95.0.0.0	95.255.255.255
R	111.111.111.111	6f.6f.6f.6f	A	111.0.0.0	111.255.255.255
Q	219.26.51.0	db.1a.33.0	C	219.26.51.0	219.26.51.255
P	167.0.0.0	a7.0.0.0	B	167.0.0.0	167.0.255.255
O	240.1.1.1	f0.1.1.1	E	-	-
N	10.1.1.1	a.1.1.1	A	10.0.0.0	10.255.255.255
M	225.1.1.1	e1.1.1.1	D	-	-

Instrucciones.

- Abrir un block de notas y escribir el código correspondiente en C.
- La dirección IP se inicializará en hexadecimal en un arreglo de caracteres sin signo
- La máscara de red se inicializará en decimal en un arreglo de caracteres sin signo
- Compilar en consola haciendo uso de gcc (en caso de no tenerlo, instalarlo).
- Una vez terminado deberán probar con las 14 direcciones IP de la tabla. Tomar captura de pantalla de la salida para cada IP y pegarlas al final de este documento.





Se entrega este documento, incluir:

Todo tu Código

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

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

```
unsigned char* Clase_Y_Mascara(unsigned char *IP0,unsigned char* limInf);  
void Rango(unsigned char* IP,unsigned char* MASK);  
void Tipo(unsigned char* IP,unsigned char *MASK);  
int main(){  
    unsigned char IP[4],limInf;  
    unsigned char* MASK;  
    printf("Ingrese La IP: ");  
    scanf("%x.%x.%x.%x",&IP[0],&IP[1],&IP[2],&IP[3]);  
    printf("IPv4: %.2x.%.2x.%.2x.%.2x\n",IP[0],IP[1],IP[2],IP[3]);  
    MASK = Clase_Y_Mascara(&IP[0],&limInf);  
    if (MASK[0]){printf("Mascara de subred: %.2x.%.2x.%.2x.%.2x\n",MASK[0],MASK[1],MASK[2],MASK[3]);  
    }else printf("No tiene mascara de subred default!!\n");  
    Rango(IP,MASK);  
  
    return 0;  
}
```

```
unsigned char* Clase_Y_Mascara(unsigned char *IP0,unsigned char* limInf){  
    unsigned char* Mask = (unsigned char*)calloc(4,sizeof(unsigned char));  
    if (*IP0&128){  
    if(*IP0&64){  
    if(*IP0&32){  
    if(*IP0&16){  
        printf("Clase: E\n");Mask[0] = 0;Mask[1] = 0;Mask[2] = 0;Mask[3] = 0;*limInf = 240;  
    }else {printf("Clase: D\n"); Mask[0] = 0;Mask[1] = 0;Mask[2] = 0;Mask[3] = 0;*limInf = 224;}  
    }else {printf("Clase: C\n"); Mask[0] = 255;Mask[1] = 255;Mask[2] = 255;Mask[3] = 0;*limInf = 192;}  
    }else {printf("Clase: B\n"); Mask[0] = 255;Mask[1] = 255;Mask[2] = 0;Mask[3] = 0;*limInf = 128;}  
    }else {printf("Clase: A\n"); Mask[0] = 255;Mask[1] = 0;Mask[2] = 0;Mask[3] = 0;*limInf = 0;}  
    return Mask;  
}
```

```
void Rango(unsigned char* IP,unsigned char* MASK){  
    unsigned char* Auxiliar = (unsigned char*)calloc(4,1);
```

Dra. Nidia A. Cortez Duarte





```
*(int*)Auxiliar = *(int*)IP;
if (MASK[0])
{
    *(int*)Auxiliar = *(int*)IP & *(int*)MASK;
    printf("IP de RED: %.2x.%.2x.%.2x.%.2x\n",Auxiliar[0],Auxiliar[1],Auxiliar[2],Auxiliar[3]);
    printf("Rango de IPs Disponibles: %.2x.%.2x.%.2x.%.2x => ",Auxiliar[0],Auxiliar[1],Auxiliar[2],Auxiliar[3]+1);
    *(int*)Auxiliar = *(int*)IP | ~*(int*)MASK;
    printf("%.2x.%.2x.%.2x.%.2x\n",Auxiliar[0],Auxiliar[1],Auxiliar[2],Auxiliar[3]-1);
    printf("IP de broadcast: %.2x.%.2x.%.2x.%.2x\n",Auxiliar[0],Auxiliar[1],Auxiliar[2],Auxiliar[3]);
    *(int*)MASK = ~*(int*)MASK;
}
else{printf("SIN MASCARA!\n");}
```

Deberan crear y mostrar **el mapa de memoria** utilizado en su programa (considerar registros de 8 bits)

Ejemplo de mapa de memoria

Unsigned char IP[0]		0	0	1	0	0	0	1	0
1	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	0	0	0
Unsigned char MR[0]		1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0

Esto para: 10.1.1.1

MR: 255.0.0.0

Finalmente incluir tus **Conclusiones individuales**

* sobre el uso de

If ()

Else, como falso y verdadero

Dra. Nidia A. Cortez Duarte



Comparar el costo computacional entre.

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

Contra

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

¿Habías usado los operadores binarios?

¿Qué opinas del uso de operadores binarios?



Dra. Nidia A. Cortez Duarte



```
~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez ✓  
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4  
IP: 10.250.1.1  
HEX: a.fa.1.1  
Es clase A  
MR: 255.0.0.0  
Rango de: 10.0.0.0 a 10.255.255.255
```

```
~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez ✓  
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4  
IP: 150.10.15.0  
HEX: 96.a.f.0  
Es clase B  
MR: 255.255.0.0  
Rango de: 150.10.0.0 a 150.10.255.255
```

```
~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez ✓  
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4  
IP: 192.14.2.0  
HEX: c0.e.2.0  
Es clase C  
MR: 255.255.255.0  
Rango de: 192.14.2.0 a 192.14.2.255
```

```
~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez INT x  
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4  
IP: 148.17.9.1  
HEX: 94.11.9.1  
Es clase B  
MR: 255.255.0.0  
Rango de: 148.17.0.0 a 148.17.255.255
```

```
~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez ✓  
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4  
IP: 193.0.0.255  
HEX: c1.0.0.ff  
Es clase C  
MR: 255.255.255.0  
Rango de: 193.0.0.0 a 193.0.0.255
```

S



Dra. Nidia A. Cortez Duarte



```
~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez INT x
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4
IP: 111.111.111.111
HEX: 6f.6f.6f.6f
Es clase A
MR: 255.0.0.0
Rango de: 111.0.0.0 a 111.255.255.255

~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez ✓
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4
IP: 219.26.51.0
HEX: db.1a.33.0
Es clase C
MR: 255.255.255.0
Rango de: 219.26.51.0 a 219.26.51.255

~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez ✓
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4
IP: 167.0.0.0
HEX: a7.0.0.0
Es clase B
MR: 255.255.0.0
Rango de: 167.0.0.0 a 167.0.255.255

~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez ✓
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4
IP: 240.1.1.1
HEX: f0.1.1.1
Es clase E
Rango de: 240.0.0.0 a 240.255.255.255
```

```
~/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez INT x
cd "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/" && gcc Ipv4.c -o Ipv4 && "/home/alfredo/Voyager/ESCOM/23-2/Redes de Comp - Nidia Cortez/"Ipv4
IP: 10.1.1.1
HEX: a.1.1.1
Es clase A
MR: 255.0.0.0
Rango de: 10.0.0.0 a 10.255.255.255
```

