

# Laboratorio de Paradigmas de Programación

## Práctica 1-2

Nombre: Daniel Cruz Enriquez

1. Revise el siguiente código:

```
#include <stdio.h>

int main()
{
    char c;

    c=0xff;
    if(c==0xff)
    {
        printf("Verdadero\n");
    }
    else
    {
        printf("Falso\n");
    }

    return 0;
}
```

2. Sin compilar ni ejecutar el programa, ¿cuál es la salida de este código?, explique por qué.

Se declara la variable c de tipo carácter dentro del main y no se le pone un valor al inicio por lo que la variable puede contener basura, pero después se le asigna un valor, el if compara la variable c con el valor 0xff que es el valor que se le dio por lo tanto la salida debería de ser verdadero

3. Compile el código y responda a las siguientes preguntas:

- 3.1. ¿El compilador arroja errores o advertencias?

El compilador no arroja errores ni advertencias para el programa

- 3.2. ¿Cuál es la salida del programa?

La salida del programa fue Falso

4. Investigue los rangos permitidos para los siguientes tipos de datos:

Tipo	Tamaño en bytes	Valor mínimo	Valor máximo	Formato para printf	Formato para constante
char	1	-128	127	%c	char a ="a";
unsigned char	1	0	255	%c	unsigned char a = "a";
int	2	-2147483648	2147483647	%d	int b = 1;
unsigned int	4	0	4294967295	%d	unsigned int c;
long	8	-9223372036854775807	9223372036854775807	%D o %ld	long = 1234;
unsigned long	4	0	18446744073709551615	%lu	unsigned long valor;
long long	8	-9,223,372,036,854,775,808	9.223.372.036.854.775.807	%lld	long long valor;
unsigned long long	8	0	18.446.744.073.709.551.615	%llu	unsigned long valor;
float	4	-----	-----	%f	float n;
double	8	-----	-----	%f	Double n;
long double	12	-----	-----	%Lf	long double n;

*Fuentes:*

<https://altamiraweb.net/especificadores-de-formato-en-c/>

<https://learn.microsoft.com/es-es/cpp/c-language/cpp-integer-limits?view=msvc-170>

<https://learn.microsoft.com/es-es/cpp/cpp/integer-limits?view=msvc-170>

<https://ccia.ugr.es/~jfv/ed1/c/cdrom/cap2/cap24.htm>

[http://maxus.fis.usal.es/FICHAS\\_C.WEB/01xx\\_PAGS/0101.html](http://maxus.fis.usal.es/FICHAS_C.WEB/01xx_PAGS/0101.html)

5. Haga un programa que:

- 5.1. Declare una variable de cada uno de los tipos del punto anterior
- 5.2. Inicialice cada una de las variable de forma estática con el valor máximo o mínimo permisible, según el caso
- 5.3. Imprima con printf correctamente el tamaño en bytes y el valor asignado de cada una de las variables

```
DocParadigmas > C tamaños.c > ul
1  #include <stdio.h>
2  #include <limits.h>
3
4  // Variables inicializadas
5  char c = CHAR_MIN;
6  unsigned char uc = UCHAR_MAX;
7  int i = INT_MAX;
8  unsigned int ui = UINT_MAX;
9  long l = LONG_MIN;
10 unsigned long ul = ULONG_MAX;
11 long long ll = LLONG_MIN;
12 unsigned long long ull = ULLONG_MAX;
13
14 //float f = FLT_MAX;
15 //double d = DBL_MAX;
16 //long double ld = LDBL_MAX;
17
18 int main()
19 {
20     // Imprimir resultados
21     printf("char: %d\n", (int)c);
22     printf("unsigned char: %u\n", (unsigned int)uc);
23     printf("int: %d\n", i);
24     printf("unsigned int: %u\n", ui);
25     printf("long: %ld\n", l);
26     printf("unsigned long: %lu\n", ul);
27     printf("long long: %lld\n", ll);
28     printf("unsigned long long: %llu\n", ull);
29     //printf("float: %f\n", f);
30     //printf("double: %f\n", d);
31     //printf("long double: %Lf\n", ld);
32     return 0;
```

```
char: -128
unsigned char: 255
int: 2147483647
unsigned int: 4294967295
long: -9223372036854775808
unsigned long: 18446744073709551615
long long: -9223372036854775808
unsigned long long: 18446744073709551615
```

ProgramasParadigmas > main ~3 > 0.458s

danxct / fedora

6. Regrese al código del punto número 1, explique claramente que es lo que hace la computadora para arrojar el resultado obtenido en el punto número 3.

El programa arroja el resultado de falso ya que el valor maximo que puede guardar char es 127 y a este se le esta asignando 0xff que es 255 en valor decimal, 255 no cabe en char y se trunca a -1 entonces dentro de la comparación **if** se compara -1 con 255 lo que resulta en **falso**