

## Laboratorio 1

Nombre: Benjamin Martinez Jeldres

Carrera:IECI

### I) Actividades de nivelación

```
[1]+  Stopped                  ./primer
bennix@DESKTOP-EARE9UP:~/arqui$ nano primer.c
bennix@DESKTOP-EARE9UP:~/arqui$ gcc primer.c -o primer
bennix@DESKTOP-EARE9UP:~/arqui$ ./primer
Hola Mundobennix@DESKTOP-EARE9UP:~/arqui$ nano primer.c
bennix@DESKTOP-EARE9UP:~/arqui$ gcc primer.c -o primer
bennix@DESKTOP-EARE9UP:~/arqui$ ./primer
Hola Mundo
  Soy Benjamin Martinez
bennix@DESKTOP-EARE9UP:~/arqui$ _
```

### II) Actividades del laboratorio

1)

Tipo Dato	Bits	Rango Permitido
int	16 bits	-32768 a 32767
signed int	16 bits	-32768 a 32767
Unsigned int	16 bits	0 a 65535
Short int	16 bits	-32768 a 32767
Signed short int	16 bits	-32768 a 32767
unsigned short int	16 bits	0 a 65535
Long int	32 bits	-2147483648 a 2147483647
Signed long int	32 bits	-2147483648 a 2147483647
Unsigned long int	32 bits	0 a 4294967295
Float	32 bits	3.4E-38 a 3.4E+38
Double	64 bits	1.7E-308 a 1.7E+308
Long double	64 ó 80 bits	1.7E-308 a 1.7E+308 ó 3.4E-4932 a 1.1E+4932
Signed char	8 bits	-128 a 127
Unsigned char	8 bits	0 a 255

2)

- a) Las librerías son; `#include<Stdio.h>` y `#include<Stdint.h>`.  
`#include<Stdio.h>` importa funciones, y te permite leer desde o enviar información a periféricos, como leer variables del teclado, etc.  
`#include<Stdint.h>` se utiliza para definir los tamaños de tipos enteros.

b) El comando `return 0`; sirve para que el `main` devuelva un valor 0 para terminar el programa, que se suele usar para informar que el proceso esta correcto.

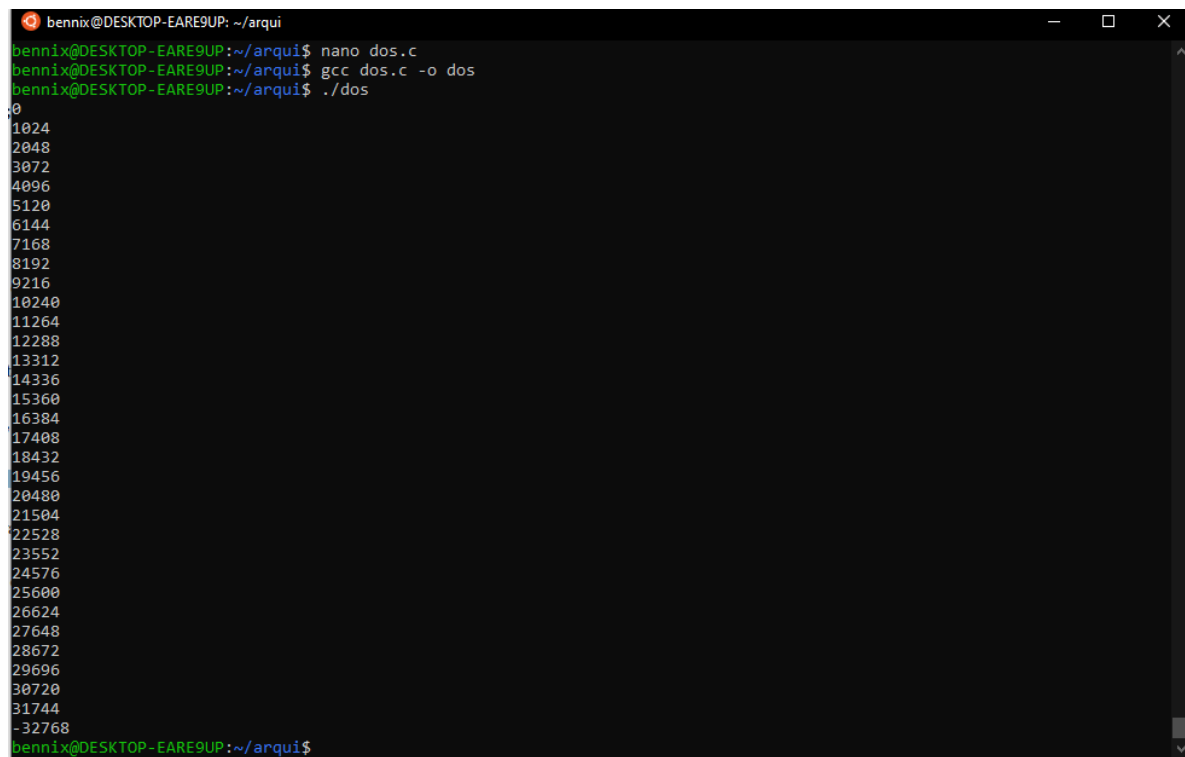
c) La función `main()` sirve como punto de partida para la ejecución del programa, el cual puede contener dentro de ella otras funciones.

¿Quién ejecuta el código declarado por `main`?

R: `while(value>=0){}`

d) sí, ya que `value` ira aumentando en valor, por lo que , `value` siempre sera mayor a 0, y continuara de manera infinita.

e) Ejecute el código, y no entiendo el resultado, si bien el valor de `value` va en aumento como predije, no se en que momento o porque en valor entero positivo cambia a negativo aparte de que no será infinito.



```
bennix@DESKTOP-EARE9UP: ~/arqui
bennix@DESKTOP-EARE9UP:~/arqui$ nano dos.c
bennix@DESKTOP-EARE9UP:~/arqui$ gcc dos.c -o dos
bennix@DESKTOP-EARE9UP:~/arqui$ ./dos
1024
2048
3072
4096
5120
6144
7168
8192
9216
10240
11264
12288
13312
14336
15360
16384
17408
18432
19456
20480
21504
22528
23552
24576
25600
26624
27648
28672
29696
30720
31744
-32768
bennix@DESKTOP-EARE9UP:~/arqui$
```

f) Rango de `value`= 1 a 33 valores.

`Int16_t` significa que el numero entero con signo será de 2 bytes

g) El programa terminara de inmediato ya que se cumplirán las condiciones del While(){}

```
31744
-32768
bennix@DESKTOP-EARE9UP:~/arqui$ nano dos.c
bennix@DESKTOP-EARE9UP:~/arqui$ gcc dos.c -o dos
bennix@DESKTOP-EARE9UP:~/arqui$ ./
-bash: ./: Is a directory
bennix@DESKTOP-EARE9UP:~/arqui$ ./dos
0
-1024
bennix@DESKTOP-EARE9UP:~/arqui$
```

3)

a) Yo creo que, si lo ejecuta de forma infinita, creo lo mismo que en el ejercicio anterior, pero uint\_16t no lo conozco por lo que quizás este errado.

b) Efectivamente se está ejecutando de manera infinita.

c) El rango de valores fue infinito, uint\_16t significa que es de 65535 como mínimo

d) Se diferencia del tipo de entero, Int16\_t es diferente de uint\_16t, ya que el rango es limitado por lo que el programa terminaría mucho antes.

4)

a)

```
printf_bin
bennix@DESKTOP-EARE9UP:~/arqui$ nano cuatro.c
bennix@DESKTOP-EARE9UP:~/arqui$ gcc cuatro.c -o cuatro
cuatro.c: In function 'print_bin':
cuatro.c:6:28: error: expected '}' before numeric constant
    printf("%d", (value & hex)0);
                           ^
bennix@DESKTOP-EARE9UP:~/arqui$ nano cuatro.c
bennix@DESKTOP-EARE9UP:~/arqui$ gcc cuatro.c -o cuatro
bennix@DESKTOP-EARE9UP:~/arqui$ ./cuatro
254(10)=1111111111111111111111111111111100000010(2)
bennix@DESKTOP-EARE9UP:~/arqui$
```

Todavía no aprendo mucho sobre números binarios, así que no puedo contestar esta pregunta.