

=====

Bibliografía:

Brian Kernighan y Dennis Ritchie, El lenguaje de programación C
(título original en inglés: The C Programming Language)

Eduardo Grosclaude, Taller de Lenguaje en C
<https://se.fi.uncoma.edu.ar/so/clases/apuntes/taller-c.pdf>

0. Escriba un programa hello world. Compilar y ejecutar.
(utilice como soporte la sección "1.4 Un primer ejemplo" del PDF taller-c.pdf)

Por ejemplo:

```
vi hello.c # edicion. Usar vi u otro editor de predilección
# como pluma
make hello # compilacion
gcc -o hello hello.c # compilación invocando gcc)
./hello # ejecución
```

NOTA: Para los ejercicio 1. a 9. inclusive utilice como soporte el capítulo "Capítulo 3 Tipos de datos y expresiones" del PDF taller-c.pdf

1. Enumerar los tipos de datos básicos en C. Escriba una definición de una variable de cada tipo.

Por ej.: int a;

// Reserva un byte con signo (solo valor de una letra)

char a = 'a';

//pensado en c, para albergar una word (palabra), no tiene tamaño fijo, depende arquitectura y compilador.
Mínimo de 16 bits (2 bytes).

int b = 32000;

// Reserva espacio en memoria en para un punto flotante IEEE 754. Siempre ocupa 32 bits (estándar).

float c = 2,16;

// Tmb punto flotante pero doble precisión. Siempre ocupa 64 bits (estándar).

double pi =3,1415;

int d

//Suma sus valores

d = a+b;

// Queda truncado (32 bits vs 64 bit). Resuelve con formato de variable más alta. En este caso double. La asignación es truncada pues d es de 32bits.

d= c+pi

C tiene un tipo booleano (no lo vamos a usar). Históricamente para estructura de control se usa 0 (falso), todo lo que no es 0 es verdadero. Usaremos esta convención

2. Explique qué significa la palabra word cuando el contexto es la Arquitectura de una Computadora.

Word: es el tamaño de datos natural con la cual trabaja una arquitectura, es un tamaño fijo y se utiliza para representar números, direcciones de memoria, etc. También puede reflejar el ancho de los registros de la CPU, el ancho del bus de datos, el tamaño de las operaciones que puede realizar la ALU.

- 3. ¿Cuántos bits puede almacenar una variable declarada como int en C? (ej. int var;) Suponga 3 computadoras de arquitecturas diferentes: arquitecturas de 8bits, 32bits y 64bits?. ¿Y en su PC?**
NOTA: la respuesta no es tan trivial como parece. Deberá buscar información sobre el lenguaje de programación C. Use wikipedia u otros recursos.

(Encontrado en Strackoverflow)

Dependiendo de la implementación del compilador de C, pero el mínimo es de 16 bits (2 bytes). Tanto en arqs de 86x y 64x, ocupará 32 bits (4 bytes).

8 bits: ocupará 2 palabras de 8bits (mínimo de 16bits preestablecido) (2 bytes)

32 bits: ocupará 1 palabra de 32bits (4bytes)

64 bits: ocupará media palabra de 64bits (4bytes)

Se puede verificar con el operador sizeof().

4. Explique la diferencia en la salida de las siguientes instrucciones:

```
char a = 'M';  
printf ("a = %i \n", a);  
printf ("La letra %c \n", a);
```

¿Cuál es el valor numérico de a?

El valor numérico de de a es 77 (ASCII table value of M).

Ayuda: En la función "printf()" se pasa como parámetro la cadena de caracteres que se desea imprimir y, además, cada una de las variables que serán visualizadas. Si deseamos imprimir más de una variable en una cadena de caracteres, el orden de los parámetros debe corresponder al orden de dichas variables en la cadena.

```
char sensor;  
float temp;  
printf("La temperatura de %c es %f",sensor, temp);
```

5. ¿Cuál es el rango numérico de i y j en su PC?

```
char i;  
unsigned char j;
```

Mediante sizeof() se obtiene que ambos son de 1 byte (8 bits). Es decir:

Rango de i: [0, 255]

Rango de j: [-128, 127]

6. ¿Cuál es el valor en base 2 (binario) de i, j, k?

```
char i = 'a';  
char j = 77;  
char k = 0x4D;
```

7. ¿Cuáles de estas declaraciones contienen errores?

```
integer a;  
short i,j,k;  
long float (h);  
double long d3;  
unsigned float n;  
char 2j;  
int MY;  
float ancho, alto, long;  
bool i;
```

8. Averigüe los tamaños de todos los tipos básicos en su sistema aplicando el operador sizeof().

9. Prepare un programa con una variable de tipo char y otra de tipo unsigned char. Inicialice ambas variables con los valores máximos de cada tipo, para comprobar el resultado de incrementarlas en una unidad. Imprima los valores de cada variable antes y después del incremento. (Nota: vea la sección "3.2. Tamaños de los objetos de datos" de taller-c.pdf.)

10. (utilice como soporte la sección "3.3. Operaciones con distintos tipos" del PDF taller-c.pdf)

- a. ¿Qué hace falta corregir para que la variable r contenga la división exacta de a y b?

```
int a, b;  
float r;  
a = 5;  
b = 2;  
r = a / b;
```

- b. ¿Qué resultado puede esperarse del siguiente fragmento de código?

```
int a, b, c, d;  
a = 1;  
b = 2;  
c = a / b;  
d = a / c;
```

- c. ¿Cuál es el resultado del siguiente fragmento de código? Anticipe su respuesta en base a lo dicho en esta unidad y luego confirme mediante un programa.

```
printf("%d\n", 20/3);  
printf("%f\n", 20/3);  
printf("%f\n", 20/3.);  
printf("%d\n", 10%3);  
printf("%d\n", 3.1416);  
printf("%f\n", (double)20/3);  
printf("%f\n", (int)3.1416);  
printf("%d\n", (int)3.1416);
```

11. Escribir un programa que multiplique e imprima $100000 * 100000$. ¿De qué tamaño son los ints en su sistema?

12. Descargue el código ahorcado.c propuesto por la cátedra.

- a. Investigar cuál es la función que cumplen las siguientes líneas de código:

```
#include <stdio.h>  
#include <stdlib.h>  
system("/bin/stty raw");  
system("/bin/stty sane erase ^H");
```

- b. Complete el código ahorcado.c usando printf() y getchar(), para desarrollar el juego del ahorcado. (Resol en git y lab)

ahorcado.c (<https://se.fi.uncoma.edu.ar/so/>)

```
#include <stdio.h>  
#include <stdlib.h> /* para las funciones system y exit */  
int main() {  
    int c;  
    /* Decirle al sistema que el modo input es RAW */  
    system("/bin/stty raw");  
    while(1) {  
        printf("\n");  
        printf("\r c = %c ingrese una letra (0 para salir): ", c);  
        c = getchar();  
        if (c == '0') break;  
    }  
    system("/bin/stty sane erase ^H");  
}
```