



Unidad Curricular:
Técnicas de Programación I
II Semestre

ING. DUBRASKA ROCA
CORREO: DROCA754@GMAIL.COM
TELÉFONO: 04140463588



El lenguaje de Programación C

- **Lenguaje de alto nivel de propósito general.**
 - **Sintaxis sumamente compacta**
 - **Alta portabilidad (independencia del hardware subyacente)**
 - **Muy buenas facilidades para acceso a bajo nivel**
 - **Gran parte de la funcionalidad se implementa en bibliotecas externas**
 - **E/S**
 - **Gestión de tiras de caracteres**
 - **Asignación de memoria**
 - **...**



Ejemplos de programas en C

- **ESTRUCTURA DE UN PROGRAMA EN C**
- **VARIABLES. TIPOS. ÁMBITO. CONVERSIÓN. MODIFICADORES**
- **OPERADORES LÓGICOS, ARITMÉTICOS, RELACIONALES**
- **ESTRUCTURAS CONDICIONALES**
- **ESTRUCTURAS DE CONTROL**



Ejemplo 1: Estructura de un programa

- Inclusión de bibliotecas, declaraciones de variables y tipos, y secuencia de funciones
- Una y sólo una función denominada `main` (programa principal)
- Función:

```
tipoRetorno Nombre (parametros) {sentencias}
```

```
/* ejemplo 1.- Escribe un mensaje en pantalla */  
  
# include <stdio.h> /* incluye biblioteca donde se define E/S */  
int main( )  
{  
    /*Este comentario es ignorado por el compilador y*/  
    /*no genera código */  
    printf("Introducción a la programación en lenguaje C\n");  
    return 0;  
}
```

Ejemplo 2: Definición de variables

- **Declaración variable:**

```
tipo Nombre [ =valor];
```

- **Asignación:**

```
variable = expresión;
```

```
/* ejemplo 2.- multiplica dos números enteros y muestra el  
resultado por pantalla */
```

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

```
int main( )
```

```
{
```

```
    int multiplicador; /*se define multiplicador como un entero */
```

```
    int multiplicando; /*se define multiplicando como un entero */
```

```
    int res; /*se define resultado como un entero*/
```

```
    multiplicador = 1000; /*se asignan valores*/
```

```
    multiplicando=2;
```

```
    res=multiplicador*multiplicando;
```

```
    printf("Resultado =%d\n",res); /*se muestra el resultado */
```

```
    return 0;
```

```
}
```



Ejemplo 3: Definición de variables

```
/* ejemplo 3.- .- multiplica dos números enteros y muestra el
   resultado (utiliza definición múltiple de variables) */

#include <stdio.h>

int main( ) {
    int multiplicador, multiplicando; /*se definen 2 variables*/

    multiplicador =1000; /*se les asigna valor*/
    multiplicando=2;
    printf("Resultado = %d\n", multiplicador*multiplicando);
    /*se muestra el resultado por pantalla*/
    return 0;
}
```



Tipos de variables

- **Variables de tipo entero**

TIPO	BYTES	VALOR MINIMO	VALOR MAXIMO
signed char	1	-128	127
unsigned char	1	0	255
unsigned short	2	-32.768	+32.767
unsigned short	2	0	+65.535
signed int	2	-32.768	+32.767
unsigned int	2	0	+65.535
signed long	4	-2.147.483.648	+2.147.483.647
unsigned long	4	0	+4.294.967.295

- **Nota:** Si se omite el clasificador delante de la variable de tipo entero por defecto se considera "signed"



Tipos de variables

- Variables de tipo real

TIPO	BYTES	VALOR MINIMO	VALOR MAXIMO
float	4	3.4E-38	3.4E+38
double	8	1.7E-308	1.7E+308
long double	10	3.4E-4932	3.4E+4932

- Nota: Las variables de punto flotante son siempre con signo



Ejemplo 4: Conversiones de tipo

```
/* ejemplo 4.- se realizan conversiones de tipos implícitas y
   explícitas */

#include <stdio.h>
int main() {
    double d , e , f = 2.33 ;
    int i = 6 ;

    e = f * i ; /* e es un double de valor 13.98*/
    printf( "Resultado = %f\n", e);
    d = (int) ( f * i ) ; /* d es un double de valor 13.00*/
    printf( "Resultado = %f\n", d);
    d = (int) f * i ; /* f se ha convertido a un entero truncando*/
                       /*sus decimales, d es un double de valor 13.00*/
    printf( "Resultado = %f\n", d);
    return 0;
}
```



Ejemplo 5: Variable tipo carácter

```
/* ejemplo 5.- Distintas formas de asignar un carácter a una
variable de tipo char */
```

```
int main () {
    char c;

    c=97; /* el valor en decimal del código ASCII*/
    c='a'; /* el carácter entre comillas*/
    c=0x61; /* el valor en hexadecimal del código ASCII*/
    c=0141; /* el valor en octal del código ASCII*/
    return 0;
}
```

Secuencias de escape

- Existe una serie de caracteres no imprimibles que el editor los toma como órdenes, por lo que la manera de acceder a ellos es mediante una secuencia de escape

CODIGO	SIGNIFICADO	VALOR ASCII (decimal)	VALOR ASCII (hexadecimal)
'\n'	nueva línea	10	0x0A
'\r'	retorno de carro	13	0x0D
'\f'	nueva página	12	0x0C
'\t'	tabulador horizontal	9	0x09
'\b'	retroceso (<i>backspace</i>)	8	0x08
'\"'	comilla simple	39	0x27
'\''	comillas	34	0x22
'\\'	barra hacia atrás (<i>backslash</i>)	92	0x5C
'\?'	Interrogación	63	0x3F
'\nnn'	cualquier carácter (donde nnn es el código ASCII expresado en octal)		
'\xnn'	cualquier carácter (nn es el código ASCII expresado en hexadecimal)		

Ejemplo 6: Tamaño de las variables

```
/* ejemplo 6.- Uso del operador sizeof para determinar el tamaño de
una variable */
#include <stdio.h>

int main () {
    char c;
    int n, d;

    n= sizeof(c);
    printf(" el número de bytes de la variable c es %d\n", n);
    n= sizeof(d);
    printf(" el número de bytes de la variable d es %d\n", n);
    n= sizeof(int);
    printf(" el número de bytes que ocupa el tipo entero es %d\n",
n);
    printf(" el número de bytes que ocupa el tipo double es %d\n",
sizeof(double));
    return 0;
}
```

Ejemplo 7: Constantes y E/S simple

```
/* ejemplo 7.- Calcula el perímetro de una circunferencia cuyo
   radio se introduce por teclado */

#include <stdio.h>
#define PI 3.1416      /* definición de constante */

int main () {
    float perim, radio; int dos=2;

    printf(" Calcula el perímetro de una circunferencia\n");
    printf(" Indique el tamaño de radio de la circunferencia");
    scanf("%f", &radio);
    perim= dos*PI*radio;
    printf(" El perímetro de la circunferencia es %f\n", perim);
    printf("Valores utilizados para calcular el perímetro:\n");
    printf(" Constante PI=%f, valor de dos = %d, radio=%f ",
    PI, dos,radio);
    return 0;
}
```

Ejemplo 8: La función printf ()

```
/* ejemplo 8.- Ilustra formatos con reales y enteros */

#include <stdio.h>
#define va_int 805
#define va_float 332.41e-1

int main () {
    printf(" %f \n", va_float); /* imprime 33.241*/
    printf(" %.1f \n", va_float); /* imprime 33.2*/
    printf(" %.4f \n", va_float); /* imprime 33.2410*/
    printf(" %1.4e \n", va_float); /* imprime 3.3241e+01*/
    printf(" %d \n", va_int); /* imprime 805*/
    printf(" %10f\n", va_float); /* imprime 33.241*/
    return 0;
}
```



Ejemplo 9: La función scanf ()

```
/* Ejemplo 9.- Introducción de datos por teclado */  
  
#include <stdio.h>  
  
int main () {  
    int i;  
    float x;  
    printf(" teclee el número entero i ");  
    scanf("%d", &i);  
    printf(" teclee el número real x ");  
    scanf("%f", &x);  
    return 0;  
}
```

Ejemplo 10: La instrucción `if-else`

- Cuando hay muchas opciones el programa se hace difícil de entender
- `getchar()`, lee un carácter de la entrada estándar (teclado)

```
#include <stdio.h>
int main() {
    int c;

    printf(" Menu:\n");
    printf(" A=Añadir a la lista\n");
    printf(" B=Borrar de la lista\n");
    printf(" O=Ordenar la lista\n");
    printf(" I=Imprimir la lista\n");
    printf(" Escriba su selección y luego <enter>\n");
    if ((c=getchar()) != '\n') {
        if (c=='A') printf(" Has seleccionado añadir\n");
        else if (c=='B') printf(" Has seleccionado borrar\n");
        else if (c=='O') printf(" Has seleccionado ordenar\n");
        else if (c=='I') printf(" Has seleccionado imprimir\n");
    } else printf(" No has seleccionado una opcion valida");
}
```




Ejemplo 11: La instrucción switch

```
#include <stdio.h>
int main() {
    int nota;

    printf(" Inserte una nota: "); scanf("%d",&nota);
    switch(nota) {
        case 0: printf("\nSuspenso"); break;
        case 1: printf("\nSuspenso"); break;
        case 2: printf("\nSuspenso"); break;
        case 3: printf("\nSuspenso"); break;
        case 4: printf("\nSuspenso"); break;
        case 5: printf("\nAprobado"); break;
        case 6: printf("\nBien"); break;
        case 7: printf("\nNotable"); break;
        case 8: printf("\nNotable"); break;
        case 9: printf("\nSobresaliente"); break;
        case 10: printf("\nSobresaliente"); break;
        default: printf("esa nota es incorrecta");
    }
    return 0;
}
```



Ejemplo 12: La instrucción switch

```
#include <stdio.h>

int main() {
    int nota;

    printf(" Inserte una nota: "); scanf("%d",&nota);
    switch(nota) {
        case 0: case 1: case 2: case 3: case 4: printf("\nSuspenso"); break;
        case 5: printf("\nAprobado"); break;
        case 6: printf("\nBien"); break;
        case 7:
        case 8: printf("\nNotable"); break;
        case 9:
        case 10: printf("\nSobresaliente"); break;
        default: printf("esa nota es incorrecta");
    }
    return 0;
}
```



Ejemplo 13: La instrucción switch

```
/* ejemplo 13.- Conversión entre euros y pesetas */

#include <stdio.h>
#define euro 166.386

int main() {
    float n,x; int opcion;

    printf("la cantidad: "); scanf("%f",&n);
    printf("1-Ptas a Euros 2-Euros a ptas"); scanf("%d",&opcion);
    switch(opcion) {
        case 1: x=n/euro;
                printf("%f Pesetas son %f Euros\n",n,x); break;
        case 2: x=n*euro;
                printf("%f Euros son %f Pesetas\n",n,x); break;
        default: printf("opción incorrecta\n");
    }
    return 0;
}
```



Ejemplo 14: Instrucción while

```
/* ejemplo 14.- Lee una palabra y lo escribe al revés */  
  
#include <stdio.h>  
  
int main () {  
    char l, palabra[21];  
    int i;  
  
    printf("Teclee una palabra de menos de 20 letras:");  
    scanf("%s", palabra);  
    i = 0;  
    while(palabra[i++] != '\\0') ;  
    l = i-1; printf("%s tiene %d letras\\n", palabra, l);  
    printf("%s escrita al revés es: ", palabra);  
    i = l;  
    while (i > 0)  
        printf("%c", palabra[--i]);  
    return 0;
```

Ejemplo 15: do-while

```
/* ejemplo 15.- Suma de n números introducidos por teclado (valor
   <0 para terminar) */

#include <stdio.h>

int main() {
    int num=0,suma=0;

    do {
        suma=suma+num;
        printf("un número: ");
        scanf("%d",&num);
    } while(num>=0);
    printf("suma es: %d",suma);
    return 0;
}
```

Ejemplo 16: for y parámetros por valor

```
#include <stdio.h>

int f(int m, int n);

int main() {
    int i;
    printf("Prueba: función entero
    elevado a potencia");
    for (i=0; i<10; i++) {
        printf("2^%d es:%d",i,f(2,i));
        printf("3^%d es:%d",i,f(3,i));
    }
    return 0;
}

int f(int base,int n) {
    int i,p=1;

    for (i=1;i<=n ;i++) p=p*base;
    return p;
}
```

```
#include <stdio.h>

int f(int m, int n);

int main() {
    int i;
    printf("Prueba: función entero
    elevado a potencia");
    for (i=0; i<10; i++) {
        printf("2^%d es:%d",i,f(2,i));
        printf("3^%d es:%d",i,f(3,i));
    }
    return 0;
}

int f(int base,int n) {
    int p=1;

    while (n-- >0) p=p*base;
    return p;
}
```



Gracias por su
atención....