Universidad de Concepción
Facultad de Ingeniería Agrícola

# FUNCIONES EN C

Facultad de Ingeniería Agrícola
www.fiaudec.cl

Juan Carlos Sandoval Avendaño

Edición Enero 2020

# FUNCIONES

## 1 Funciones

Las *funciones* son un conjunto de instrucciones que realizan una tarea específica, y nos permiten crear módulos reutilizables de código.
Existen funciones predefinidas que están incluidas en las bibliotecas, y funciones definidas por el programador.
Todas las funciones definidas por el programador deben ser declaradas e implementadas.
La declaración de una función se denomina *prototipo*.
La implementación está relacionada con el código que permite realizar la tarea específica de la función.

Sintaxis de la declaración o prototipo:

**tipo_funcion nombre_funcion(tipo_p1 nombre_p1, ...,tipo_pN nombre_pN);**

Sintaxis de la implementación o definición:

```
tipo_funcion nombre_funcion(tipo_p1 nombre_p1, ...,tipo_pN nombre_pN)
{
  Instrucciones
}
```

donde

**tipo_funcion** es el tipo de dato que retorna la función, pudiendo ser void que significa que la función no devuelve un valor

**nombre_funcion** es el nombre que se le asigna a la función y que debiese ser lo suficientemente descriptivo de la tarea que realiza la función

**tipo_px** es el tipo de dato asociado al parámetro x-ésimo

**nombre_px** es el nombre que se le asigna al parámetro x-ésimo

En la declaración de la función es opcional la aparición de los nombres de los parámetros.

La mayoría de las funciones incluyen la orden $\mathrm{return}$ que permite devolver o retornar un valor. En la implementación pueden aparecer tantos $\mathrm{return}$ como valores devuelva la función.

## 2 Función Principal main()

Todos los programas en C cuentan con, a lo menos, una función y esta es la función principal main(), que retorna un valor entero igual a 0, si la ejecución ha sido correcta.
Las funciones definidas por el programador pueden ser implementadas antes o después de la función principal main(), pero el prototipo debe aparecer entre la sección de inclusión (include) y la función principal main().

## 3 Función Recursiva

Una función es recursiva cuando se llama a si misma al interior de su definición. Por ejemplo, la función factorial puede ser considerada recursiva porque $n! = (n-1)! \cdot n$, es decir, para calcular $n!$ necesitamos calcular $(n-1)!$

La recursión se ocupa cuando la definición de la función a implementar es recursiva y, por lo tanto, es fácil codificarla.

# Ejercicios
# Funciones

**[1]** Escriba un programa en C que muestre por pantalla los valores de la función $f(x) = e^{-x} + x^2$, para $x = 1,\ 1.2,\ 1.4,\ ...., 6$
Defina y use una función apropiada.

**[2]** $a)$ ¿Qué es una función?. ¿Para qué sirve?. ¿Cómo se declara una función?. ¿Qué es un valor de retorno?. ¿Qué son los parámetros o argumentos?.
$b)$ Escriba una función que genere un número aleatorio en el intervalo $[\,a\,,\,b]$, donde $a$ y $b$ son los argumentos de la función. Asuma que $a$ y $b$ son números enteros y además $a < b$

**[3]** Escriba un programa en C que muestre por pantalla un mensaje indicando si se cumple la relación $|\,f(x_0)\,f''(x_0)| < (f'(x_0))^2$ para $f(x) = e^x sen(2x)$ y para un $x_0$ ingresado por el usuario. Incorpore en el programa las funciones que considere necesarias.
El programa debe repetir el procedimiento hasta que el usuario ingrese un valor de $x_0$ que satisfaga la relación dada.

**[4]** Escriba un programa en C que haciendo uso de una función recursiva muestre por pantalla los primeros 15 números de la sucesión de Fibonacci.
La sucesión de Fibonacci queda definida por :

$f_0 = 0\ ;\ f_1 = 1\,;\ f_n = f_{n-1} + f_{n-2}\,,\ n = 2, 3, 4, ....$

**[5]** Recordando que $n! = n \cdot (n-1)!$ escriba una función, definida por usted, que permite calcular y mostrar por pantalla el factorial de un número natural ingresado por el usuario.

**[6]** Escriba un programa en C que, haciendo uso de una función apropiada, calcule la primera derivada de cuatro funciones polinomiales de la forma

$f(x) = ax^5 + bx^4 + cx^3 + dx^2 + ex + f$, donde $a, b, c, d, e$ y $f$ son números reales ingresados por el usuario al igual que el valor de $x$. Muestre los resultados usando el circunflejo (^) para representar potencias.

**[7]** Escriba un programa en C que muestre por pantalla la tabla

| angulo | seng(angulo) | cosg(angulo) |
|---|---|---|
| 0 | 0.00 | 1.00 |
| 45 | 0.71 | 0.71 |
| 90 | 1.00 | 0.00 |
| 180 | 0.00 | -1.00 |
| 270 | -1.00 | 0.00 |

haciendo uso de funciones seng y cosg, que calculan el seno y el coseno del argumento en grados respectivamente, definidas por usted.

**[8]** Escriba un programa en C que, haciendo uso de una función que determine si un número entero es par o impar, muestre por pantalla la suma de todos los pares y el producto de todos los impares de un conjunto de $150$ numeros aleatorios generados por el computador, entre $-67$ y $140$ inclusive.

**[9]** Escriba una función que calcule el factorial de un número natural, y use tal función para calcular la suma $\sum_{k=0}^{8} \frac{1}{k!}$

**[10]** Escriba un programa en C que calcule y muestre por pantalla el valor de la función

$$f(x) = \begin{cases} e^{\sqrt{x^2+1}} & \text{, si } x > \sqrt{7} \\ |\,2x - 1\,| & \text{, si } 0 \leq x \leq \sqrt{7} \\ 10^{-3}\frac{x^4+x^6}{x^2+1} & \text{, si } x < 0 \end{cases}$$

para un valor de $x$ real ingresado por el usuario.
Use una función apropiada definida por usted.

**[11]** Escriba un programa que determine el menor entero $n$ para el cual la suma $1+2+3+\ ...\ +n$ es mayor que $1000$. Haga uso de una función apropiada.

**[12]** Escriba un programa en C que muestre por pantalla, en color, la ecuación y las soluciones de la ecuación cúbica $dx^3+ax^2+bx+c=0$, donde los coeficientes $d$, $a$, $b$ y $c$ son ingresados por el usuario, usando el siguiente esquema de trabajo:

Para la ecuación cúbica $x^3+a\,x^2+b\,x+c=0$ (si el coeficiente que acompaña a $x^3$ no es $1$, entonces divida la ecuación por tal coeficiente), con $a,\ b,\ c\ \in \mathbb{R}$, primero calcule

$$Q \equiv \frac{a^2-3b}{9} \qquad\qquad R \equiv \frac{2a^3-9ab+27c}{54}$$

Si $R^2 < Q^3$, entonces la ecuación posee tres raíces reales, las cuales son:

$$x_1 = -2\sqrt{Q}\,cos(\frac{\theta}{3}) - \frac{a}{3}$$

$$x_2 = -2\sqrt{Q}\,cos(\frac{\theta+2\pi}{3}) - \frac{a}{3}$$

$$x_3 = -2\sqrt{Q}\,cos(\frac{\theta-2\pi}{3}) - \frac{a}{3}$$

donde $\theta = Arccos\left(\frac{R}{\sqrt{Q^3}}\right)$.

Si $R^2 \geq Q^3$, entonces las tres raíces son:

$$x_1 = (A+B) - \frac{a}{3}$$

$$x_2 = -\tfrac{1}{2}(A+B) - \frac{a}{3} + i\frac{\sqrt{3}}{2}\,(A-B)$$

$$x_3 = -\tfrac{1}{2}(A+B) - \frac{a}{3} - i\frac{\sqrt{3}}{2}\,(A-B)$$

donde $A = -signo(R)\left[|R| + \sqrt{R^2-Q^3}\right]^{1/3}$

$$B = \begin{cases} Q/A\,,\ si\ A \neq 0 \\ \\ 0\,,\ si\ A = 0 \end{cases}$$

$$signo(R) = \begin{cases} +1 & ,\ si\ R \geq 0 \\ -1 & ,\ si\ R < 0 \end{cases}$$

Incluya funciones apropiadas.

**[13]** Escriba un programa en C que calcule y muestre por pantalla el valor de la función

$$f(x) = \begin{cases} \frac{1-x}{\sqrt{x^2-1}} & ,\ \text{si}\ x > 1 \\ x + 1 & ,\ \text{si}\ 0 \leq x \leq 1 \\ ln(|\,x\,|) & ,\ \text{si}\ x < 0 \end{cases}$$

para un valor de $x$ real ingresado por el usuario. El programa debe preguntar al usuario si desea repetir la evaluación o si desea terminar. Si el usuario ingresa NO el programa termina, en caso contrario el programa continúa.
Use una función apropiada definida por usted.

**[14]** Escriba una aplicación de consola que sume y reste dos números enteros aleatorios positivos y menores o iguales a $10$.
Defina y use funciones suma y resta apropiadas.

**[15]** Escriba una aplicación en C que calcule y muestre por pantalla la suma de los primeros $21$ polinomios de Laguerre evaluados en $x_0$, donde $x_0$ es ingresado por el usuario.
El programa debe definir y utilizar una función apropiada.
(Ind. : $P_0(x) = 1;\ P_1(x) = 1 - x;\ P_{n+1}(x) = \frac{(2n+1-x)}{n+1}\,P_n(x) - \frac{n}{n+1}\,P_{n-1}(x)$ )

**[16]** Usando una función apropiada, calcule y muestre por pantalla el valor de $f(x,y) = \begin{cases} x^2 + y^2 & ,\ \text{si}\ x > y \\ \frac{3}{2}x - 6y & ,\ \text{en otro caso} \end{cases}$ donde $x$ y $y$ son ingresados por el usuario.

**[17]** Escriba un programa en C que muestre por pantalla los resultados de evaluar la función

$f(x, y) = x^2 - 3y^3$

para $x = m, m+1, ...., n-1, n$ y $y = m, m+1, ..., n-1, n$
donde $m$ y $n$ son números enteros ingresados por el usuario, con $m < n$
Debe implementar una función apropiada.

**[18]** Use el lenguaje C para mostrar por pantalla los valores de la función $f(x) = \sqrt{x+3}$ para $x = 0, h, 2h, ..., xmax$, donde $h$ y $xmax$ son ingresados por el usuario.
Defina y use una función apropiada.

**[19]** Escriba un programa en C que dibuje N líneas horizontales, la primera de longitud 1, la segunda de longitud 2, hasta la última de longitud N, donde N es un número natural ingresado por el usuario, haciendo uso de una función que dibuje una línea horizontal de tamaño L.

# PROBLEMAS RESUELTOS
# FUNCIONES

**3**

Escriba un programa en C que muestre por pantalla un mensaje indicando si se cumple la relación $|f(x_0)\, f''(x_0)| < (f'(x_0))^2$ para $f(x) = e^x sen(2x)$ y para un $x_0$ ingresado por el usuario. Incorpore en el programa las funciones que considere necesarias.
El programa debe repetir el procedimiento hasta que el usuario ingrese un valor de $x_0$ que satisfaga la relación dada.

**Solución:**

**Código Fuente**

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

float f(float);
float fp(float);
float fpp(float);

float f(float x)
{
  return exp(x)*sin(2*x);
}

float fp(float x)
{
  return exp(x)*sin(2*x)+2*exp(x)*cos(2*x);
}

float fpp(float x)
{
```

```
 return 4*exp(x)*cos(2*x)-3*exp(x)*sin(2*x);
}


int main()
{
 float x0;

 do
 {
  printf("\nIngrese x0=");
  scanf("%f",&x0);
 }
 while (fabs(f(x0)*fpp(x0))>= fp(x0)*fp(x0));

 printf("\nLa condici%cn se cumple para %0.1f\n",162,x0);

 printf("\n\n");
 system("PAUSE");
 return 0;
}
```

**Salida**

```
Ingrese x0=6
Ingrese x0=1
Ingrese x0=0.5
La condición se cumple para 0.5

Presione una tecla para continuar . . .
```

## Observaciones:

1) En un programa podemos escribir tantas funciones como necesitemos. En este ejercicio se han utilizado tres funciones, una para $f(x)$, otra para $f'(x)$ y la última para $f''(x)$.

2) Las derivadas hemos de calcularlas manualmente.
Tenemos que

$f(x) = e^x sen(2x)$
$f'(x) = e^x sen(2x) + 2\,e^x cos(2x)$
$f''(x) = e^x sen(2x) + 2\,e^x cos(2x) + 2\,e^x cos(2x) - 4\,e^x\, sen(2x)$
$= 4\,e^x\, cos(2x) - 3\,e^x sen(2x)$ ❑

## 4

Escriba un programa en C que haciendo uso de una función recursiva muestre por pantalla los primeros 15 números de la sucesión de Fibonacci.
La sucesión de Fibonacci queda definida por :

$f_0 = 0\ ;\ f_1 = 1\ ;\ f_n = f_{n-1} + f_{n-2}\ ,\ n = 2, 3, 4, ....$

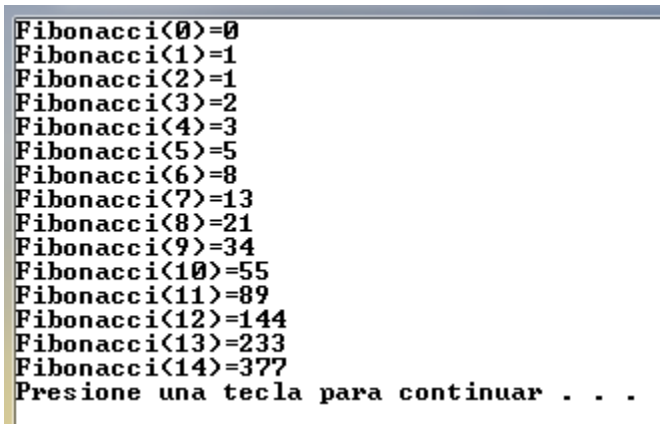
**Solución:**

**Código Fuente**

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

int Fibonacci(int);

int main()
{
  int Total=15;
  int n;
  for (n=0;n<=Total-1;n++)
   printf("Fibonacci(%i)=%i\n",n,Fibonacci(n));
  system("PAUSE");
  return 0;
}

int Fibonacci(int n)
{
   if (n==0) return 0;
   if (n==1) return 1;
```

```
    if (n>=2) return Fibonacci(n-1)+Fibonacci(n-2);
}
```

**Salida**




## Observaciones:

1) La implementación de una función puede aparecer antes o después de la función principal main().

2) La función de Fibonacci es un ejemplo de función recursiva, porque en su definición se hacen llamados a la misma función.

3) En una función se pueden incluir tantos return como valores puede entregar la función. ❑

## 8

Escriba un programa en C que, haciendo uso de una función que determine si un número entero es par o impar, muestre por pantalla la suma de todos los pares y el producto de todos los impares de un conjunto de $25$ numeros aleatorios generados por el computador, entre $-67$ y $140$ inclusive.

**Solución:**

**Código Fuente**

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

int par(int);

int par(int x)
{
 if(x%2==0) return 1;
 else return 0;
}


int main()
{
 int i, num, sumaPar=0, prodImpar=1;
 srand(time(NULL));
 for(i=1;i<=25;i++)
 {
  num=rand()%208-67;
  printf("\nnum %i = %i\n",i,num);
  if (par(num)==1) sumaPar=sumaPar+num;
  else if(num%3==0) prodImpar=prodImpar*num;
 }
 printf("\nLa suma de los pares es %i\n",sumaPar);
 printf("\nEl producto de los impares es %i\n",prodImpar);

 system("PAUSE");
 return 0;
}
```

**Salida**

```
num 1 = 76
num 2 = 11
num 3 = 88
num 4 = -57
num 5 = 23
num 6 = 62
num 7 = -10
num 8 = 51
num 9 = -34
num 10 = -8
num 11 = 90
num 12 = 116
num 13 = 7
num 14 = -54
num 15 = -66
num 16 = 42
num 17 = 19
num 18 = 127
num 19 = -28
num 20 = 61
num 21 = 54
num 22 = 59
num 23 = 26
num 24 = -42
num 25 = 57
La suma de los pares es 312
El producto de los impares es -165699

Presione una tecla para continuar . . .
```

## Observaciones:

1) En este ejercicio se asumió que si el número es par, entonces la función devuelve un 1, y si es impar devuelve un 0. Esta asignación de valores es arbitraria, esto significa que podemos relacionar par e impar con los valores que queramos. ❑

## 12

Escriba un programa en C que muestre por pantalla, en color, la ecuación y las soluciones de la ecuación cúbica $dx^3 + ax^2 + bx + c = 0$, donde los coeficientes $d$, $a$, $b$ y $c$ son ingresados por el usuario, usando el siguiente esquema de trabajo:

Para la ecuación cúbica $x^3 + a\,x^2 + b\,x + c = 0$ (si el coeficiente que acompaña a $x^3$ no es $1$, entonces divida la ecuación por tal coeficiente), con $a,\ b,\ c \in \mathbb{R}$, primero calcule

$$Q \equiv \frac{a^2 - 3b}{9} \qquad\qquad R \equiv \frac{2a^3 - 9ab + 27c}{54}$$

Si $R^2 < Q^3$, entonces la ecuación posee tres raíces reales, las cuales son:

$$x_1 = -2\sqrt{Q}\,cos(\frac{\theta}{3}) - \frac{a}{3}$$

$$x_2 = -2\sqrt{Q}\,cos(\frac{\theta + 2\pi}{3}) - \frac{a}{3}$$

$$x_3 = -2\sqrt{Q}\,cos(\frac{\theta - 2\pi}{3}) - \frac{a}{3}$$

donde $\theta = Arccos\left(\frac{R}{\sqrt{Q^3}}\right)$.

Si $R^2 \geq Q^3$, entonces las tres raíces son:

$$x_1 = (A + B) - \frac{a}{3}$$

$$x_2 = -\tfrac{1}{2}(A + B) - \frac{a}{3} + i\frac{\sqrt{3}}{2}(A - B)$$

$$x_3 = -\tfrac{1}{2}(A + B) - \frac{a}{3} - i\frac{\sqrt{3}}{2}(A - B)$$

donde $A = -signo(R)\left[|R| + \sqrt{R^2 - Q^3}\right]^{1/3}$

$$B = \begin{cases} Q/A\,,\ si\ A \neq 0 \\ 0\,,\ si\ A = 0 \end{cases}$$

$$signo(R) = \begin{cases} +1 & , si\ R \geq 0 \\ -1 & , si\ R < 0 \end{cases}$$

Incluya funciones apropiadas.

**Solución:**

**Código Fuente**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <conio2.h>

int signo(float);

int main()
{
  float a, b, c, d;
  float Q, R;
  float angulo;
  float x1, x2, x3;
  float x2r, x2i, x3r, x3i;
  float A, B;

  printf("\nIngrese d = ");
  scanf("%f",&d);
  printf("\nIngrese a = ");
  scanf("%f",&a);
  printf("\nIngrese b = ");
  scanf("%f",&b);
  printf("\nIngrese c = ");
  scanf("%f",&c);

  textbackground(WHITE);

  if (d!=0.0)
  {
    if (d!=1)
    {
```

```
        a=a/d; b=b/d; c=c/d;
      }
    Q=(a*a-3*b)/9.0; R=(2*pow(a,3)-9*a*b+27*c)/54;
    angulo=acos(R/pow(sqrt(Q),3));
    if (R*R<pow(Q,3))
    {
        x1=-2*sqrt(Q)*cos(angulo/3.0)-a/3;
        x2=-2*sqrt(Q)*cos((angulo+2*M_PI)/3.0)-a/3;
        x3=-2*sqrt(Q)*cos((angulo-2*M_PI)/3.0)-a/3;
        textcolor(RED);
        printf("\nx1 = %0.5f\nx2 = %0.5f\nx3 = %0.5f\n",x1,x2,x3);
    }
    else
    {
        A=-signo(R)*pow((fabs(R)+sqrt(R*R-pow(Q,3))),1/3.);
        if (A!=0.0)
         B=Q/A;
        else
         B=0;
        x1=(A+B)-a/3.;
        x2r=-1/2.*(A+B)-a/3.;
        x2i=sqrt(3)/2*(A-B);
        x3r=x2r;
        x3i=-x2i;
        textcolor(CYAN);
        printf("\nx1 = %0.5f\n",x1);
        printf("\nx2 = %0.5f + %0.5f i\n",x2r,fabs(x2i));
        printf("\nx3 = %0.5f - %0.5f i\n",x3r,fabs(x3i));
    }
   }
  printf("\n\n");
  system("PAUSE");
  return 0;
}
int signo(float x)
{
   if (x>=0.0) return 1;
   else return -1;
}
```

**Salida**

```
Ingrese d = -3
Ingrese a = 2
Ingrese b = 1
Ingrese c = -5
x1 = -1.07768
x2 = 0.87218 + 0.88647 i
x3 = 0.87218 - 0.88647 i

Presione una tecla para continuar . . .
```

## Observaciones:

1) En este ejercicio se asocia, arbitrariamente, signo positivo con el $1$ y signo negativo con el $-1$.

2) Recordemos que si una función posee raíces complejas estas aparecen de a pares, es decir, si $a+bi$ es raíz, entonces $a-bi$ también lo será. ❑

## 13

Escriba un programa en C que calcule y muestre por pantalla el valor de la función

$$f(x)=\begin{cases}\frac{1-x}{\sqrt{x^2-1}} & \text{, si } x>1\\ x+1 & \text{, si } 0\le x\le 1\\ ln(|\,x\,|) & \text{, si } x<0\end{cases}$$

para un valor de $x$ real ingresado por el usuario. El programa debe preguntar al usuario si desea repetir la evaluación o si desea terminar. Si el usuario ingresa NO el programa termina, en caso contrario el programa continúa.
Use una función apropiada definida por usted.

**Solución:**

**Código Fuente**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

float f(double);

float f(double x)
{
  if (x>1.0) return ((1-x)/sqrt(x*x-1));
  else if ((x>=0.0) && (x<=1.0)) return (x+1);
  else return (log(fabs(x)));
}

int main()
{
 float valor;
 char Resp[3];
 do
 {
  printf("\nIngrese un valor real en donde evaluar la funci%cn: ",162);
  scanf("%f",&valor);
  printf("\nf(%0.2f) = %0.2f\n",valor,f(valor));
  printf("\nDesea continuar (SI o NO) : ");
  scanf("%s",Resp);
 }
 while(strcmp(Resp,"NO")!=0);

 printf("\n\n");
 system("PAUSE");
 return 0;
}
```

**Salida**

```
Ingrese un valor real en donde evaluar la función: 0.5
f(0.50) = 1.50
Desea continuar (SI o NO) : SI
Ingrese un valor real en donde evaluar la función: 1.2
f(1.20) = -0.30
Desea continuar (SI o NO) : SI
Ingrese un valor real en donde evaluar la función: -2.5
f(-2.50) = 0.92
Desea continuar (SI o NO) : NO

Presione una tecla para continuar . . .
```

## Observaciones:

1) En este ejercicio se asume que el usuario ingresa NO sólo en mayúsculas.

2) El bucle do...while permite repetir el proceso hasta que el usuario ingrese NO.

3) El tipo de la variable $x$ es double pudiendo haber sido float, la diferencia es sólo el intervalo en que se mueve cada tipo (Ver 8)

## 14

Escriba una aplicación de consola que sume y reste dos números enteros aleatorios positivos y menores o iguales a 10.
Defina y use funciones suma y resta apropiadas.

**Solución:**

**Código Fuente**

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

```
#include <time.h>

int suma(int, int);

int suma(int a,int b)
{
  return a+b;
}

int resta(int,int);

int resta(int a, int b)
{
  return a-b;
}

int main()
{

  srand(time(NULL));
  int x,y;
  x=rand()%10+1;
  y=rand()%10+1;
  printf("\n%i + %i = %i\n",x,y,suma(x,y));
  printf("\n%i - %i = %i\n",x,y,resta(x,y));

  printf("\n\n");
  system("PAUSE");
  return 0;
}
```

**Salida**

```
4 + 1 = 5
4 - 1 = 3

Presione una tecla para continuar . . .
```

Observaciones:

1) La función suma posee dos argumentos de tipo entero. ❑

## 15

Escriba una aplicación en C que calcule y muestre por pantalla la suma de los primeros $21$ polinomios de Laguerre evaluados en $x_0$, donde $x_0$ es ingresado por el usuario.
El programa debe definir y utilizar una función apropiada.
(Ind. : $P_0(x) = 1;\ P_1(x) = 1 - x;\ P_{n+1}(x) = \frac{(2n+1-x)}{n+1} P_n(x) - \frac{n}{n+1} P_{n-1}(x)$ )

**Solución:**

**Código Fuente**

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

float P(float, int);

float P(float x, int n)
{
 if(n==0) return 1;
 else if(n==1) return 1-x;
 else return (2*n-1.-x)/n*P(x,n-1)-(n-1.)/n*P(x,n-2);
}

int main()
{
 float x0, suma=0.0;
 int n;

 printf("\nIngrese x0 : ");
 scanf("%f",&x0);
```

```
  for(n=0;n<=20;n++)
   suma=suma+P(x0,n);

  printf("\nLa suma de los 21 primeros polinomios evaluados");
  printf(" en x0 = %0.1f es %0.4f\n",x0,suma);

  printf("\n\n");
  system("PAUSE");
  return 0;
}
```

**Salida**

```
Ingrese x0 : 0.8
La suma de los 21 primeros polinomios evaluados en x0 = 0.8 es 1.9641

Presione una tecla para continuar . . .
```

## Observaciones:

1) Los polinomios de Laguerre se obtienen mediante el uso de una función recursiva, porque es fácil implementar la función dadas las relaciones $P_0(x) = 1;\ P_1(x) = 1 - x;\ P_{n+1}(x) = \frac{(2n+1-x)}{n+1} P_n(x) - \frac{n}{n+1} P_{n-1}(x)$ ❑

## 16

Usando una función apropiada, calcule y muestre por pantalla el valor de $f(x,y) = \begin{cases} x^2 + y^2 & \text{, si } x > y \\ \frac{3}{2}x - 6y & \text{, en otro caso} \end{cases}$ donde $x$ y $y$ son ingresados por el usuario.

**Solución:**

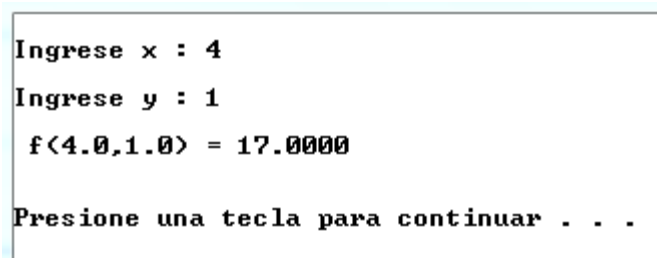
**Código Fuente**

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

float f(float , float);

float f(float x, float y)
{
 if(x>y) return x*x+y*y;
 else return 3./2*x-6*y;
}

int main()
{
 float x,y;

 printf("\nIngrese x : ");
 scanf("%f",&x);

 printf("\nIngrese y : ");
 scanf("%f",&y);

 printf("\n f(%0.1f,%0.1f) = %0.4f\n",x,y,f(x,y));

 printf("\n\n");
 system("PAUSE");
 return 0;
}
```

**Salida 1**

```
Ingrese x : 0.6
Ingrese y : 3
 f(0.6,3.0) = -17.1000

Presione una tecla para continuar . . .
```

**Salida 2**




## 17

Escriba un programa en C que muestre por pantalla los resultados de evaluar la función

$f(x,y) = x^2 - 3y^3$

para $x = m, m+1, ...., n-1, n$ y $y = m, m+1, ..., n-1, n$
donde $m$ y $n$ son números enteros ingresados por el usuario, con $m < n$
Debe implementar una función apropiada.

**Solución:**

**Código Fuente**

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

int f(int x, int y);

int f(int x, int y)
{
    return pow(x,2)-3*pow(y,3);
}

int main()
```

```
{
 int x,y;
 int m,n;
 do
 {
  printf("\nIngrese m : ");
  scanf("%i",&m);
  printf("\nIngrese n : ");
  scanf("%i",&n);
 }
 while(m>=n);
 for(x=m;x<=n;x++)
  for(y=m;y<=n;y++)
   printf("\nf(%i,%i) = %i\n",x,y,f(x,y));

 printf("\n\n");
 system("PAUSE");
 return 0;
}
```

**Salida**

```
Ingrese m : 3
Ingrese n : 1
Ingrese m : 1
Ingrese n : 3
f(1,1) = -2
f(1,2) = -23
f(1,3) = -80
f(2,1) = 1
f(2,2) = -20
f(2,3) = -77
f(3,1) = 6
f(3,2) = -15
f(3,3) = -72

Presione una tecla para continuar . . .
```

Observaciones:

1) Si $m$ y $n$ son enteros, entonces los valores de $x$ y $y$ también son enteros de acuerdo al enunciado.

2) En el prototipo o declaración de la función se indican los nombres de los argumentos, pero recordemos que esto es opcional. ❑

## 18

Use el lenguaje C para mostrar por pantalla los valores de la función $f(x) = \sqrt{x+3}$ para $x = 0, h, 2h, ..., xmax$, donde $h$ y $xmax$ son ingresados por el usuario.
Defina y use una función apropiada.

**Solución:**

**Código Fuente**

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

float f(float);    // declaración de  f

int main()
{
  float xmax, h;
  float x;
  printf("Ingrese el l%cmite superior del intervalo : ",161);
  scanf("%f",&xmax);
  printf("Ingrese el tama%co de paso h : ",164);
  scanf("%f",&h);
  printf("\n");
```

```
  for (x=0;x<=xmax;x=x+h)
   printf("f(%0.2f)= %0.2f\n",x,f(x));

  printf("\n\n");
  system("PAUSE");
  return 0;
}

float f(float x)       // definición de f
{
   return sqrt(x+3);
}
```

**Salida 1**

```
Ingrese el límite superior del intervalo : 5
Ingrese el tamaño de paso h : 0.5

f(0.00)= 1.73
f(0.50)= 1.87
f(1.00)= 2.00
f(1.50)= 2.12
f(2.00)= 2.24
f(2.50)= 2.35
f(3.00)= 2.45
f(3.50)= 2.55
f(4.00)= 2.65
f(4.50)= 2.74
f(5.00)= 2.83

Presione una tecla para continuar . . .
```

**Salida 2**

```
Ingrese el límite superior del intervalo : 1
Ingrese el tamaño de paso h : 1

f(0.00)= 1.73
f(1.00)= 2.00

Presione una tecla para continuar . . .
```

## Observaciones:

1) En este programa se asume que el usuario ingresa valores apropiados para $h$ y $xmax$.

2) Si el usuario ingresa un valor de $xmax$ menor que $h$, entonces la salida es

```
Ingrese el límite superior del intervalo : 1
Ingrese el tamaño de paso h : 4

f(0.00)= 1.73


Presione una tecla para continuar . . .
```

Es decir, se evalúa la función sólo en $x = 0$. ❑

## 19

Escriba un programa en C que dibuje N líneas horizontales, la primera de longitud 1, la segunda de longitud 2, hasta la última de longitud N, donde N es un número natural ingresado por el usuario, haciendo uso de una función que dibuje una línea horizontal de tamaño L.

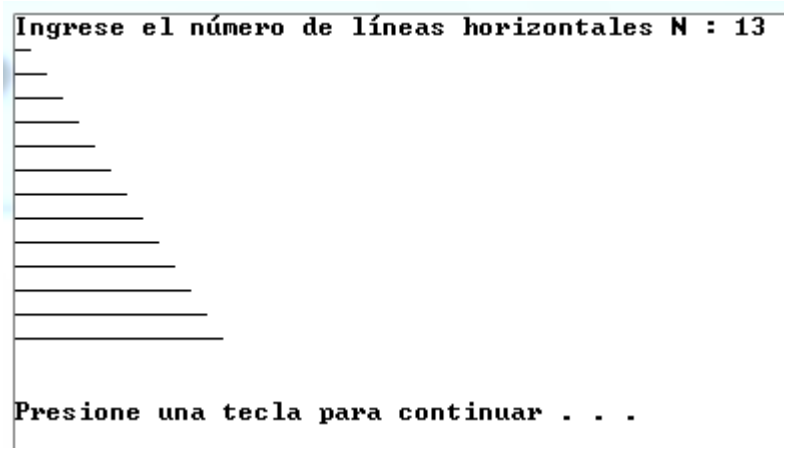
**Solución:**

**Código Fuente**

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

void LineaHorizontal(int);

void LineaHorizontal(int L)
{
   int i;
   for (i=1;i<=L;i++)
    printf("%c",196);
   printf("\n");
}

int main()
{
```

```
  int i, N;
  printf("Ingrese el n%cmero de l%cneas horizontales N : ",163,161);
  scanf("%i",&N);
  for (i=1;i<=N;i++)
   LineaHorizontal(i);

  printf("\n\n");
  system("PAUSE");
  return 0;
}
```

**Salida**



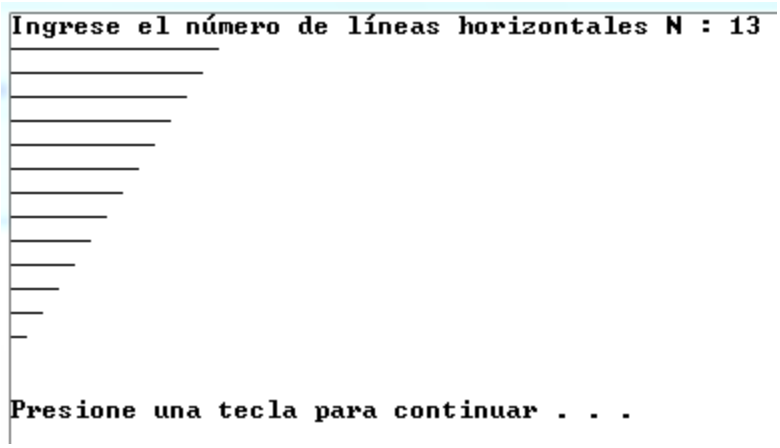

## Observaciones:

1) Se ha usado el código ascii 196 para dibujar las líneas.

2) Podemos generar el efecto contrario escribiendo en el bucle for lo siguiente

```
for (i=N;i>=1;i--)
  LineaHorizontal(i);
```

cuya salida es




3) La función LineaHorizontal no entrega un resultado (no aparece el return y el tipo asociado a la función es void o vacío), sino que realiza la acción de dibujar una línea de longitud L.
Este tipo de funciones que no devuelve un resultado se le llama, en otros lenguajes, procedimiento. ❑