



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M. I. Marco Antonio Martínez Quintana

Asignatura: Fundamentos de programación

Grupo: 3

No de Práctica(s): #12

Integrante(s): Fausto Ángel Reséndiz Álvarez

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista o Brigada: 38

Semestre: 2021-1

Fecha de entrega: 22/01/2021

Observaciones:

CALIFICACIÓN: _____

Objetivo

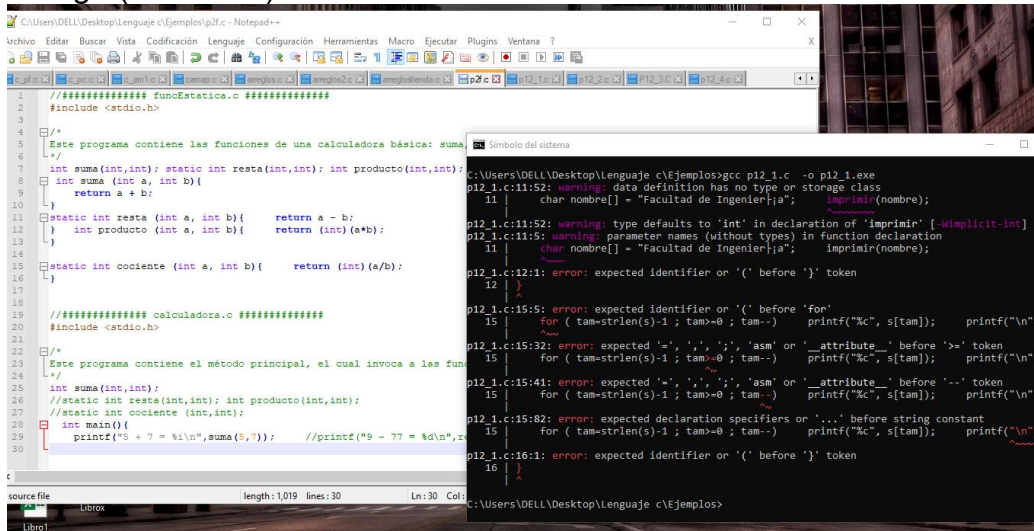
Elaborar programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

Introducción

Podemos definir una función como un conjunto de líneas de código que realizan una tarea en específico. El objetivo principal de las funciones es el agilizar los procesos tales como la depuración o el mismo rendimiento del código que se está realizando. Lenguaje C maneja principalmente es "main", la cual es un claro ejemplo de función ya que al iniciar es la encargada de ejecutar las instrucciones que ya se encuentren dentro de la misma y puede ejecutar un sinfín de otras funciones.

Desarrollo

Funciones Código (funciones)

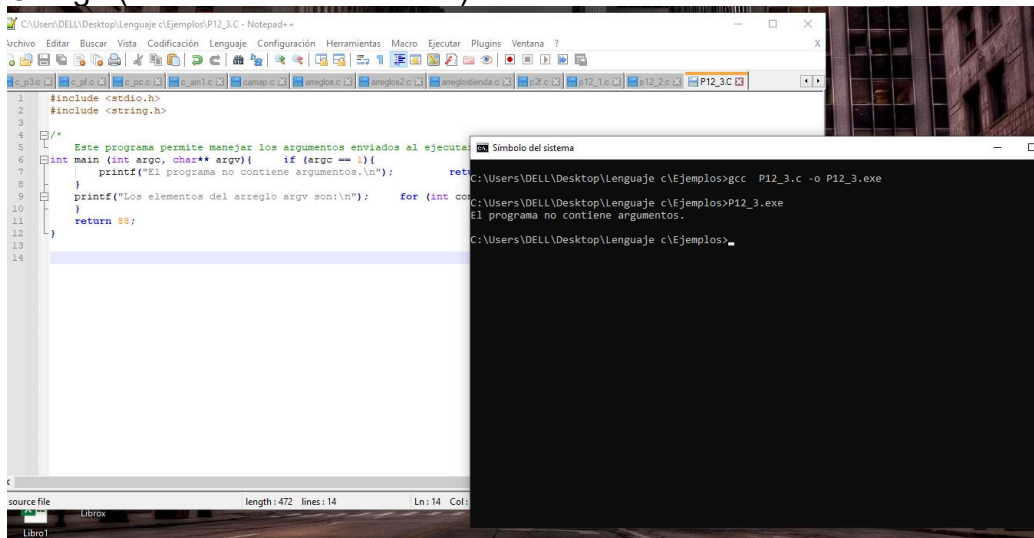


The screenshot shows a C++ IDE with two files: `funcEstadistica.c` and `calculadora.c`. The `funcEstadistica.c` file contains functions for basic arithmetic operations. The `calculadora.c` file contains the `main` function that uses these functions. The terminal window shows the following compilation errors:

```
p12_1.c:11:52: warning: data definition has no type or storage class
11 | char nombre[] = "Facultad de Ingenieria"; imprimir(nombre);
    |
p12_1.c:11:52: warning: type defaults to 'int' in declaration of 'imprimir' [-Wimplicit-int]
p12_1.c:11:55: warning: parameter names (without types) in function declaration
11 | char nombre[] = "Facultad de Ingenieria"; imprimir(nombre);
    |
p12_1.c:12:1: error: expected identifier or '(' before ')' token
12 | }
    |
p12_1.c:15:5: error: expected identifier or '(' before 'for'
15 | for ( tam=strlen(s)-1 ; tam>0 ; tam--) printf("%c", s[tam]); printf("\n");
    |
p12_1.c:15:32: error: expected 'u', 'l', 'i', 'asm' or ' _attribute_' before '>' token
15 | for ( tam=strlen(s)-1 ; tam>0 ; tam--) printf("%c", s[tam]); printf("\n");
    |
p12_1.c:15:41: error: expected 'u', 'l', 'i', 'asm' or ' _attribute_' before '...' token
15 | for ( tam=strlen(s)-1 ; tam>0 ; tam--) printf("%c", s[tam]); printf("\n");
    |
p12_1.c:15:82: error: expected declaration specifiers or '...' before string constant
15 | for ( tam=strlen(s)-1 ; tam>0 ; tam--) printf("%c", s[tam]); printf("\n");
    |
p12_1.c:16:1: error: expected identifier or '(' before ')' token
16 | }
```

- Código con error

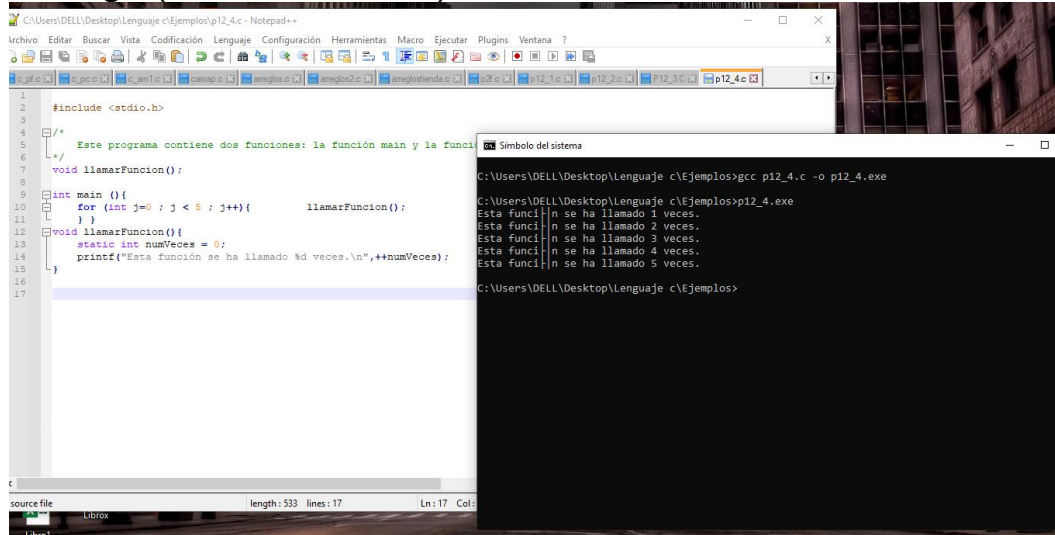
Código (Ámbito de las variables)



The screenshot shows a C++ IDE with a single file `P12_3.c`. The program demonstrates variable scope by declaring a variable `argc` in the `main` function and using it to check if any arguments were passed to the program. The terminal window shows the following output:

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc P12_3.c -o P12_3.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>P12_3.exe
El programa no contiene argumentos.
```

Código (variable estática)



The screenshot shows a C++ IDE with a source file named `p12_4.c`. The code defines a function `llamarFuncion()` and a `main` function that calls it in a loop. A static variable `numVeces` is used to count the number of calls. The terminal window shows the output of the program, which prints the number of times the function was called (1 to 5).

```
#include <stdio.h>

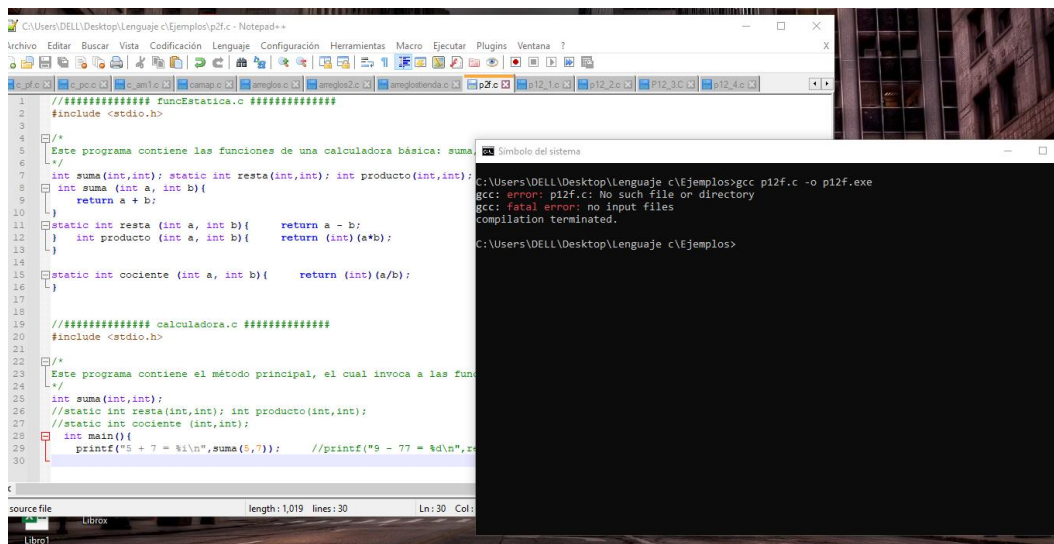
/*
 * Este programa contiene dos funciones: la función main y la función
 */
void llamarFuncion();

int main () {
    for (int j=0 ; j < 5 ; j++) {        llamarFuncion();
    }
}

void llamarFuncion() {
    static int numVeces = 0;
    printf("Esta función se ha llamado %d veces.\n", ++numVeces);
}
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc p12_4.c -o p12_4.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>p12_4.exe
Esta funci|n se ha llamado 1 veces.
Esta funci|n se ha llamado 2 veces.
Esta funci|n se ha llamado 3 veces.
Esta funci|n se ha llamado 4 veces.
Esta funci|n se ha llamado 5 veces.
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

Código (función estática)



The screenshot shows a C++ IDE with a source file named `p12f.c`. The code defines several static functions for basic arithmetic operations and a `main` function that calls them. The terminal window shows a compilation error: `gcc: error: p12f.c: No such file or directory`.

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

/*
 * Este programa contiene las funciones de una calculadora básica: suma,
 */
int suma(int,int); static int resta(int,int); int producto(int,int);
int suma (int a, int b){
    return a + b;
}

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

static int cociente (int a, int b){    return (int)(a/b);
}

##### calculadora.c #####
#include <stdio.h>

/*
 * Este programa contiene el método principal, el cual invoca a las fun
 */
int suma(int,int);
//static int resta(int,int); int producto(int,int);
//static int cociente (int,int);
int main(){
    printf("5 + 7 = %i\n",suma(5,7));    //printf("9 - 77 = %d\n",x
}
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc p12f.c -o p12f.exe
gcc: error: p12f.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

- Código con error

Conclusión

Durante el desarrollo de la práctica abordamos el tema de funciones, el papel que involucra en el desarrollo de un código, el alcance que tiene una variable y el como su tiempo de vida esta predeterminado por la posición donde esta sea declarada. De igual forma se hablo acerca de los argumentos destinados a la función main y la declaración de un elemento de tipo estático.

Una vez que se abordo el tema anteriormente mencionado y luego de haber experimentado el funcionamiento del mismo podemos concluir que efectivamente el papel que desempeñan las funciones en programación va más haya que solo iniciar un código o realizar una tarea en específico, pues las funciones le brindan al programador la facilidad de realizar múltiples acciones que puede ejecutar una tras otra y facilitar el desempeño del proceso que se desea realizar o simplemente mejorar el propio desempeño del código de manera significativa.