

Guía práctica de estudio 12: FUNCIONES

Codificar en lenguaje C lo siguientes ejercicios:

2. Diseñar y codificar 4 funciones que reciban dos datos tipos short y devuelvan el resultado como tipo float que realicen las 4 operación básicas (suma, resta, multiplicación, división). Leer los datos short en la función main y llamar en secuencia a las 4 funciones imprimiendo el resultado que devuelve cada uno.

CÓDIGO EN C

```
1  #include<stdio.h>
2  void suma(short a, short b);
3  void resta(short a, short b);
4  void multiplicacion(short a, short b);
5  void division(short a, short b);
6
7  int main(int argc, char* argv[])
8  {
9      short num1=0;
10     short num2=0;
11     printf("=====\n");
12     printf("Este programa Usa 4 funciones para realizar las funciones elementales\n");
13     printf("(Suma, Resta, Multiplicaci%cn, Divisi%cn)\n\n",162,162);
14     printf("Ingrese el primer valor: ");
15     scanf("%hd", &num1);
16     printf("Ingrese el segundo valor: ");
17     scanf("%hd",&num2);
18     //funciones que realizan operaciones
19     suma(num1,num2);
20     printf("\n");
21     resta(num1,num2);
22     printf("\n");
23     multiplicacion(num1,num2);
24     printf("\n");
25     division(num1, num2);
26
27     getchar();
28     getchar();
29     return 0;
30 }
```

```

31
32 void suma(short a, short b)
33 {
34     float resultado;
35     resultado=(float)a+b;
36     printf("\n\tEl resultado de la suma:\n\t%hd + %hd = %g",a,b,resultado);
37 }
38
39 void resta(short a, short b)
40 {
41     float resultado;
42     resultado=(float)a-b;
43     printf("\n\tEl resultado de la resta:\n\t%hd - %hd = %g",a,b,resultado);
44 }
45
46 void multiplicacion(short a, short b)
47 {
48     float resultado;
49     resultado=(float)a*b;
50     printf("\n\tEl resultado de la multiplicac%cn:\n\t%hd * %hd = %g",162,a,b,resultado);
51 }
52
53 void division(short a, short b)
54 {
55     float resultado;
56     if (b!=0)
57     {
58         resultado=(float)a/b;
59         printf("\n\tEl resultado de la divisi%cn:\n\t%hd / %hd = %g",162 ,a,b,resultado);
60     }
61     else
62     {
63         printf("Divisi%cn no realizada(No se puede dividir entre 0)",162);
64     }
65 }

```

PANTALLA DE EJECUCIÓN Y PRUEBA DE ESCRITORIO

```

C:\Users\pon_c\Documents\Programaci4n\C\Fundamentos de programaci4n\Pr4cticas\...
Este programa Usa 4 funciones para realizar las funciones elementales
(Suma, Resta, Multiplicaci4n, Divisi4n)

Ingrese el primer valor: 34
Ingrese el segundo valor: 5

    El resultado de la suma:
    34 + 5 = 39

    El resultado de la resta:
    34 - 5 = 29

    El resultado de la multiplicaci4n:
    34 * 5 = 170

    El resultado de la divisi4n:
    34 / 5 = 6.8

```

3. Realizar el código para las siguientes funciones que serán guardadas en una biblioteca para para ser incluidas en un programa donde se declare un arreglo de cinco cuadritos y se manden llamar las funciones para leer el arreglo pidiendo los valores al usuario, para imprimir el arreglo, para calcular el promedio de los elementos del arreglo y para cambiar el contenido el contenido de los cuadritos con índices de posición par por un número pedido al usuario.

```
void leer_arreglo(float arreglo[5], short cuadritos)
```

```
void imprimir_arreglo(float arreglo[5], short cuadritos)
```

```
float promedio_arreglo(float arreglo[5], short cuadritos)
```

```
void cambia_pos_par(float arreglo[5], short cuadritos, float numero)
```

CÓDIGO EN C

LIBRERÍA

```
1  #ifndef _ARREGLO_H_
2  #define _ARREGLO_H_
3
4  void leer_arreglo(float arreglo[5], short cuadritos)
5  {
6      short contador;
7
8      for(contador=0 ; contador<=cuadritos ; contador++)
9      {
10         printf("Ingrese el elemento %hd: ", contador+1); // no me gusta que empiece el conteo en 0
11         scanf("%f",&arreglo[contador]);
12     }
13 }
14
15 void imprimir_arreglo(float arreglo[5], short cuadritos)
16 {
17     short contador;
18
19     printf("===== [ARREGLO_IMPRESO] =====\n\n");
20     for(contador=0 ; contador<=cuadritos ; contador++)
21     {
22         if(contador==0)
23             printf("\t( %g,", arreglo[0]);
24
25         else if(contador==cuadritos)
26             printf(" %g)\n",arreglo[cuadritos]);
27         else
28             printf(" %g,",arreglo[contador]);
29     }
30 }
```

```

31
32 float promedio_arreglo(float arreglo[5], short cuadritos)
33 {
34     float promedio=0;
35     short contador;
36
37     for (contador=0 ; contador<=cuadritos ; contador++)
38     {
39         promedio=promedio+arreglo[contador];
40     }
41     promedio=promedio/(cuadritos+1);
42
43     return promedio;
44 }
45
46
47 void cambia_pos_par(float arreglo[5], short cuadritos, short numeros)
48 {
49     short contador;
50
51     printf("=====[ CAMBIO_DE_PARES]=====\n");
52     printf("\tCambio de contenido de espacios pares\n");
53     for (contador=0; contador<=cuadritos; contador++)
54     {
55         if (contador%2==0)
56         {
57             printf("\tIngrese el nuevo elemento %hd: ", contador);
58             scanf("%f",&arreglo[contador]);
59         }
60     }
61 }
62
63 #endif

```

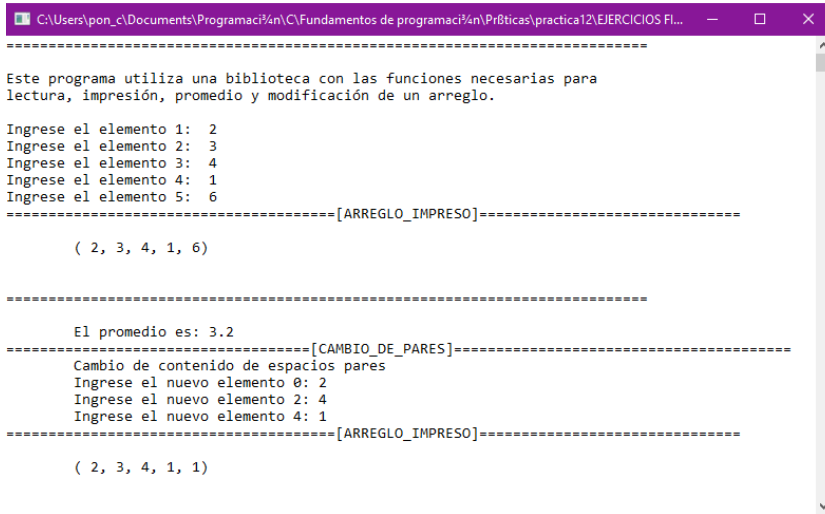
CÓDIGO PRINCIPAL

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "arreglo.h"
4
5  int main(int argc, char *argv[])
6  {
7      float Arreglo[5];
8      short espacios=4;
9      float promedio;
10     short numero=0;
11
12     printf("=====\n\n");
13     printf("Este programa utiliza una biblioteca con las funciones necesarias para \n");
14     printf("lectura, impresi%cn, promedio y modificaci%cn de un arreglo.\n\n",162,162);
15
16     leer_arreglo(Arreglo, espacios);
17     imprimir_arreglo(Arreglo, espacios);
18     promedio=promedio_arreglo(Arreglo, espacios);
19
20     printf("\n\n=====\n\n");
21     printf("\tEl promedio es: %g\n", promedio);
22
23     cambia_pos_par (Arreglo, espacios, numero);
24     imprimir_arreglo(Arreglo, espacios);
25
26     getchar ();
27     getchar ();
28     return 0;
29 }
30

```

PANTALLA DE EJECUCIÓN Y PRUEBA DE ESCRITORIO



```
-----
Este programa utiliza una biblioteca con las funciones necesarias para
lectura, impresión, promedio y modificación de un arreglo.

Ingrese el elemento 1: 2
Ingrese el elemento 2: 3
Ingrese el elemento 3: 4
Ingrese el elemento 4: 1
Ingrese el elemento 5: 6
=====[ARREGLO_IMPRESO]=====

( 2, 3, 4, 1, 6)

-----

El promedio es: 3.2
=====[CAMBIO_DE_PARES]=====
Cambio de contenido de espacios pares
Ingrese el nuevo elemento 0: 2
Ingrese el nuevo elemento 2: 4
Ingrese el nuevo elemento 4: 1
=====[ARREGLO_IMPRESO]=====

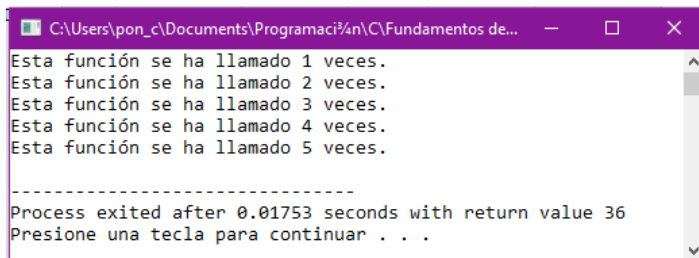
( 2, 3, 4, 1, 1)
```

4. Realizar el código de la práctica de la sala correspondiente a variable estática analizando el comportamiento y valores que toman dicha variable

CÓDIGO EN C

```
1  #include <stdio.h>
2
3  //Este programa hace uso de dos funciones, La main (principal ) La cual llama a la función secundaria "llamarfuncion"
4
5  void llamarFuncion();//cuestión estética
6
7  int main (int argc, char* argv[])//Funcion principal que hace llamar a la funcion Llamarfuncion
8  {
9      int j;
10     for (j=0 ; j < 5 ; j++)
11     {
12         llamarFuncion();
13     }
14     //El ciclo for sirve para llamar a la función llamar varias veces
15 }
16
17 void llamarFuncion();//Función secundaria
18 {
19     static int numVeces = 0; //Declaración de numVeces como variable estática
20     printf("Esta funci%cn se ha llamado %d veces.\n",162,++numVeces);
21 }
22
23 /*La primera impresión puesto es singular "Esta función se ha llamado 1 veces" podríamos corregirla
24 mediante una simple condición de impresión*/
```

PANTALLA DE EJECUCIÓN Y PRUEBA DE ESCRITORIO



```
-----
Esta función se ha llamado 1 veces.
Esta función se ha llamado 2 veces.
Esta función se ha llamado 3 veces.
Esta función se ha llamado 4 veces.
Esta función se ha llamado 5 veces.

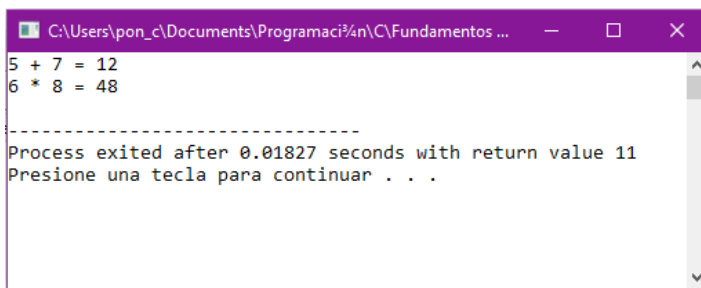
-----
Process exited after 0.01753 seconds with return value 36
Presione una tecla para continuar . . .
```

5. Realizar el código de la práctica de la sala correspondiente a funciones estáticas y comprobar el alcance

CÓDIGO EN C

```
1 //  
2 //PARTE PRINCIPAL DEL PROGRAMA  
3  
4 #include <stdio.h>  
5  
6 int suma(int,int); //Llama a la función suma  
7 int producto(int,int); //Llama a la función producto (multiplicación)  
8  
9 int main(int argc, char* argv[]) //Función principal del programa  
10 {  
11     printf("5 + 7 = %i\n", suma(5,7)); //Valores inicializados para sumar  
12     printf("6 * 8 = %i\n", producto(6,8)); //Valores inicializados para hacer el producto  
13 }  
14 //NOTA: La resta y el cociente no se despliegan en la función principal  
15  
16  
17 //  
18 #include <stdio.h> //biblioteca para static  
19  
20 //declaración de las funciones  
21 int suma(int,int);  
22 static int resta(int,int);  
23 int producto(int,int);  
24 static int cociente (int,int);  
25  
26 //Función que lee los valores a y b para posteriormente sumarlos dentro de la misma  
27 int suma (int a, int b)  
28 {  
29     return a + b;  
30 }  
31  
32 static int resta (int a, int b)  
33 {  
34     return a - b;  
35 }  
36  
37 //Función que lee los valores a y b para posteriormente multiplicarlos dentro de la misma  
38 int producto (int a, int b)  
39 {  
40     return (int)(a*b);  
41 }  
42  
43 static int cociente (int a, int b)  
44 {  
45     return (int)(a/b);  
46 }
```

PANTALLA DE EJECUCIÓN Y PRUEBA DE ESCRITORIO



```
C:\Users\pon_c\Documents\Programaci3n\C\Fundamentos ...  
5 + 7 = 12  
6 * 8 = 48  
-----  
Process exited after 0.01827 seconds with return value 11  
Presione una tecla para continuar . . .
```

CONCLUSIONES

Alfonso Murrieta Villegas

Las funciones como los arreglos son una de las muchas alternativas para poder reducir nuestros códigos, en el caso concreto de las funciones, son usadas principalmente para que estas minimicen la mayor cantidad de líneas que suelen repetirse en un código.

En la presente práctica a través de la declaración de una función principal y de varias funciones externas podemos ver como ahorramos muchas líneas de código e inclusive optimizamos a nuestros programas.