

Trabajo Práctico 2 - Parte 2



Parte 2 - Temas: Funciones estándar y definidas por el usuario. Funciones que no retornan valor (void) y que retornan valor. Paso de parámetros por valor.

ES OBLIGATORIO UTILIZAR NOMBRES DE VARIABLES Y DE FUNCIONES SIGNIFICATIVOS

Recordar

Para hacer uso de **funciones predefinidas o estándar** debe:

1. **Incluir** la **librería** en la compilación
2. **Corroborar** el directorio donde se encuentra la librería, para de acuerdo a eso utilizar la sintaxis adecuada
3. Tener presente el valor de retorno de la función

Recuerden que las funciones definidas por el usuario deben ser:

- a) **declaradas** (declaración),
- b) **implementadas** (definición) y
- c) **usadas** (invocar o llamar).

El **paso de parámetros** a una función puede hacerse:

- **Por valor**
- **Por referencia**

En el caso de paso por valor:

1. **copia el valor** de un argumento real en el parámetro formal de la función.
2. **Comunicación** de datos **unidireccional** desde el programa llamador a la función.
3. al volver no se ha modificado la variable original.

Retorno de valor

- pueden **retornar valor** (tipos de datos simples o definidos por el usuario)
- pueden **no retornar** un valor (**void**)

Problemas Resueltos

VARIABLES SIMPLES. USO DE PARÁMETROS POR VALOR

Realizar un programa que solicite tres números enteros positivos al usuario e indique cual es mayor o muestre un mensaje en caso de que haya dos o más ejercicios. Usar una función para controlar que los números que se ingresen sean positivos.

```
C ejresuelto1.c > main(int, char * [])
1  #include <stdio.h>
2  // Declaración de funciones
3  /* es_positivo:
4     parámetros: entero num
5     valor de retorno: entero.
6     Si retorna 1 significa que num es positivo,
7     si retorna 0 significa que es negativo
8  */
9  int es_positivo(int num);
10
11 //inicio de la función main
12 int main(int argc, char *argv[]) {
13     int a, b, c;
14     //ingreso de datos
15     do {
16         printf("Ingrese el primer número: ");
17         scanf("%d", &a);
18     } while (!es_positivo(a));
19     do {
20         printf("Ingrese el segundo número: ");
21         scanf("%d", &b);
22     } while (!es_positivo(b));
23     do {
24         printf("Ingrese el tercer número: ");
25         scanf("%d", &c);
26     } while (!es_positivo(c));
```

```

27     //control para saber cuál es el mayor
28     if (a > b && a > c) {
29         printf("El número %d es el mayor\n", a);
30     } else if (b > a && b > c) {
31         printf("El número %d es el mayor\n", b);
32     } else if (c > a && c > b) {
33         printf("El número %d es el mayor\n", c);
34     } else {
35         printf("Hay dos o más números iguales que son los mayores\n");
36     }
37     return 0;
38 }
39 //fin de la función main
40
41 // Definición de funciones
42 int es_positivo(int num) {
43     if (num > 0) {
44         return 1;
45     } else {
46         printf("Error: el número ingresado debe ser positivo\n");
47         return 0;
48     }
49 }
50

```

VARIABLES SIMPLES Y REGISTROS (STRUCT). USO DE PARÁMETROS POR VALOR.

Realizar un programa que permita ingresar N números por cada número ingresado se debe tener en cuenta además el orden en que fue ingresado, para ello debe utilizar una estructura para contener el número ingresado y el orden. Luego utilizar una función para encontrar el mayor valor de los números ingresados, indicando el orden en que ingresó. Presentar por pantalla los resultados. Debe controlar que el valor de N (cantidad de números a evaluar sea positivo)

```

C ejresuelto2.c > ...
1  #include <stdio.h>
2  // Definición de la estructura
3  typedef struct {
4      int numero;
5      int orden;
6  } estructura;
7  // Declaración de funciones
8  /* mayor:
9      parámetros: estructura ent, estructura aux, int c
10     valor de retorno: estructura
11     Contiene el valor del mayor número y su posición
12  */
13  estructura mayor(estructura ent, estructura aux, int c); // Retorna una estructura de tipo estructura
14
15  // inicio de la función main
16  int main(int argc, char *argv[]) {
17      estructura may, entrada; // se utilizan dos variables tipo estructura
18      int N, c;
19      c = 1;
20      printf("Ingresar cantidad de números a evaluar: ");
21      scanf("%d", &N); //aquí falta el control para que N sea positivo
22

```

```

23  while (c <= N) {
24      printf("Ingresar un nro. entero: ");
25      scanf("%d", &entrada.numero);
26      entrada.orden = c;
27      may = mayor(entrada, may, c);
28      c++;
29  }
30  printf(
31      "\nEl mayor número ingresado es %d y el orden en que ingreso es %d\n",
32      may.numero, may.orden);
33  return 0;
34  }
35  // fin de la función main
36
37
38  // Definición de funciones
39  estructura mayor(estructura ent, estructura aux, int c) {
40      if (ent.orden == 1)
41          aux = ent; // inicializ como mayor al primer valor ingresado
42  else {
43      if (ent.numero > aux.numero) aux = ent;
44  }
45  return aux;
46  }

```



Consideraciones para resolución de problemas propuestos

Los siguientes problemas, deben ser resueltos utilizando una o más funciones

En la función principal `int main(int argc, char *argv[])` **SOLAMENTE deberá:**

- Ingresar los datos de entrada solicitados en cada problema.
- Invocar a la/s función/es para el procesamiento de los datos de entrada.
- Presentar los resultados retornados por la/s función/es.

Resolver utilizando, de ser necesario, parámetros por VALOR.

NO usar paso de parámetros por referencia.

Problemas propuestos

Resolver utilizando funciones que NO retornen valor

- Implementar un programa que utilice una función para realizar una presentación de la materia, por ejemplo:

"Programacion I"

"Anio de cursada: 2023"

"Facultad de Ciencias Exactas y Tecnología"

Usar secuencias de escape para salto de línea (`\n`) y tabulaciones (`\t`).

2. Implementar un programa en el cual: se define una constante numérica llamada "PI" con el valor 3.14, se solicite que se ingrese un número que será el radio de una circunferencia, se calcule el valor de la longitud de la circunferencia y se lo presente por pantalla.

*Recordar que la longitud de la circunferencia = $2 * PI * radio$.*

Para este problema diseñar una función que reciba los dos valores y calcule el valor de la solicitado.

3. Ingresar, en una variable tipo estructura, los datos referentes a una fecha (día, mes, año). Luego usar una función para presentar por pantalla el contenido de dicha variable con el siguiente formato "la fecha ingresada es 5-4-1996"(dónde 5 corresponde a los días, 4 a los meses y 1996 a los años).

Usar la siguiente estructura para la fecha

```
typedef struct {
    int dia;
    int mes;
    int anio;
}fecha;
```

Resolver utilizando funciones que retornen valor

1. Codificar un programa que solicite al usuario un número complejo en la forma binómica y lo almacene en una estructura como el siguiente formato:

```
typedef struct {
    float real;
    float imaginario;
}complejo;
```

Luego mostrar por pantalla el mismo.

Para resolver este problema debe utilizar funciones que:

- a)Permita solicitar al usuario un número complejo
- b)Muestre un número complejo dado por pantalla con el siguiente formato: "El número ingresado es a + b i" (esta función no es necesario que retorne valor)

2. Utilizando las funciones diseñadas en el ejercicio anterior, codificar un programa en el que se solicita el ingreso de dos números complejos y muestre la suma de ambos. Para esto deberá ingresar parte real y parte imaginaria de la forma binómica en una variable tipo estructura para cada número complejo, luego utilizar otra función para calcular la suma de los mismos, además debe retornar el resultado utilizando una variable tipo estructura. Presentar por pantalla los números ingresados y el resultado.
3. Ingresar un número entero positivo, luego presentar por pantalla un mensaje que indique si el número ingresado es o no es primo. Para resolver implementar:
 - a- una función que realice el control de que el número sea positivo.
 - b- una función que determine si un número es primo.

4. Realizar un programa en lenguaje C que permita ingresar un valor N y muestre las N letras en mayúsculas en orden descendente.

Por ejemplo si se ingresa N = 5, deberá mostrar: Z, Y, X, W, V

Para resolver implementar:

a- una función que realice el control de que N sea positivo

b- una función que muestre las letras en mayúsculas en orden descendente.

5. Ingresar dos números enteros positivos N1 y N2, en donde N1 tiene que ser menor que N2. Luego sumar los números primos mayores que N1 y menores que N2, Posteriormente mostrar por pantalla el resultado de la suma.

Por ej: se ingresan N1=1 y N2=5. Los números primos entre N1 Y N2 son 2 y 3. Entonces $2 + 3 = 5$
Entonces debe mostrar por pantalla "El resultado de la suma es: 5 "

Para resolver este problema deberá **implementar las definiciones y declaraciones de las funciones que se solicitan a continuación.**

RESPETANDO LOS NOMBRES PARÁMETROS Y DE LA FUNCIÓN Y QUE PARÁMETROS RETORNA.

a- **ingresarEntero:**

Datos de entrada: ninguno

Proceso: permite ingresar un número entero y controlar si el mismo es positivo.

Salida: retorna el número ingresado.

b- **controlarNumeros:**

Datos de entrada: dos valores enteros, N1 y N2

Proceso: controlar que un valor (N1) es menor que otro (N2).

Salida: retorna una variable entera que representa si es menor o no.

c- **sumarPrimos:**

Datos de entrada: dos valores enteros (N1 y N2)

Proceso: sumar los números primos, comprendidos entre dos valores dados (N1 y N2) y mostrar por pantalla el valor.

Salida: no retorna nada

d- **Una función para determinar si un número es primo.**

Datos de entrada: un número entero

Proceso: determinar si el número es primo

Salida: retorna un entero 1= si el numero es primo, 0= si el número no es primo.

Aclaración: La suma debe efectuarse en la función "sumarPrimos" que llamará a una segunda función (ud. puede elegir el nombre) que determinará si un número es primo.

6. Leer un número entero positivo de dos o más dígitos, y luego presentar por pantalla al número ingresado y su invertido. Para resolver deberá implementar

a- función para controlar si un número es un entero positivo de dos o más dígitos

b- función para generar el invertido del número dado

7. Ingresar dos números enteros **número** y **base**, controlar mediante una función que **número** sea positivo y mediante otra función controlar que **base** sea mayor que 1 y menor que 10. Utilizar una función para realizar la conversión de **número** a la base representada por el valor **base**. Presentar ambos números por pantalla.
8. Codificar un programa en el que se solicita el ingreso de un carácter, luego presentar por pantalla un mensaje que indique si dicho carácter es un dígito, una letra del alfabeto en mayúscula, una letra del alfabeto en minúscula o de otro tipo de carácter. Para resolver deberá implementar:
 - a- una función para ingresar el carácter
 - b-una función para mostrar por pantalla qué tipo de carácter es (esta función no es necesario que retorne valor)
9. Codificar un programa en el que se solicita el ingreso de un carácter alfabético, luego presentar por pantalla el carácter en mayúscula si el mismo fue ingresado en minúscula y viceversa. Para resolver este ejercicio deberá implementar:
 - a- una función para ingresar el carácter y controlar que sea alfabético.
 - b- una función para devolver el carácter convertido
 - c-una función para mostrar por pantalla el caracter convertido(esta función no es necesario que retorne valor)
10. Codificar un programa en el que se solicita el ingreso de N caracteres, luego presentar por pantalla las consonantes ingresadas en minúscula y las vocales en mayúscula.
Para resolver este ejercicio deberá implementar:
 - a- función para controlar si el número N es un entero positivo.
 - b-función para solicitar el ingreso de los caracteres, evaluarlos y presentarlos por pantalla (las vocales en mayúscula y las consonantes en minúscula)
11. Implementar un programa que permita determinar si una **fecha es válida**, para esto :
 - a- Implementar una función permita el ingreso de una fecha y que retorne la misma al programa principal.
 - b- Implementar una función para controlar la validez de la fecha: día entre 1 y 31, mes entre 1 y 12, año mayor a 1900. Puede además controlar para cada mes la cantidad de días que corresponden.
 - c- Implementar una función para mostrar una fecha con el formato dd/mm/aaaa, por ejemplo 10/4/2022.

Almacenar la fecha en un dato tipo estructura con el siguiente formato:

```
typedef struct{  
    int dia;  
    int mes;  
    int anio;  
} fecha;
```

12. Implementar un programa que permita determinar si un medicamento se encuentra o no vencido, conociendo la fecha de vencimiento y la fecha actual. Para resolver este ejercicio deberá:

a- Usar del ejercicio anterior las funciones para leer una fecha y controlar su validez.

b- Implementar una función que determine si el medicamento está vencido y que presente por pantalla el mensaje correspondiente.

Almacenar la fecha en un dato tipo estructura con el siguiente formato:

```
typedef struct{
    int dia;
    int mes;
    int anio;
}fecha;
```

13. Codificar un programa en el que se solicita el ingreso del peso y la talla de N personas, utilizando una estructura para contener peso y talla. Debe mostrar por pantalla cuál es la persona que tiene mayor peso y talla. Para resolver debe emplear una función que use parámetros por valor tipo estructura y que retorne una estructura.
14. La empresa DATASYSTEM se encarga de realizar instalaciones de cableado estructurado. Desde la gerencia se le pide al encargado de recursos humanos que informe cuál es el empleado que tiene el menor salario para darle un aumento de sueldo del 10%.

Para ello se nos solicita realizar un programa en lenguaje C que permita encontrar entre los N empleados de la empresa el de menor salario.

Para resolver implementar:

a- Solicitar al usuario que ingrese la cantidad de empleados que posee la empresa.

Mediante una función realizar el control de que la cantidad de empleados sea un valor entero positivo.

b- Una función que permita almacenar los datos del empleado en un tipo de dato struct. El mismo contendrá el legajo del empleado (número entero mayor que cero), la inicial del apellido y el monto que recibe como salario (el mismo debe ser un valor positivo).

c- Una función que me permita encontrar el menor salario

d- Una función que permita proporcionarle al menor salario un aumento del 10%

e- Presentar por pantalla el menor sueldo con el aumento dado y sin el mismo.