Funciones

Ejercicio 4.1

Escribir una función que reciba como parámetros la longitud de los lados de un triángulo y que **retorne** el área del mismo. Implementarla en un programa que reciba los parámetros como datos e informe el área calculada.

Ejemplo:
Ingrese lado 1: ***2***
Ingrese lado 1: ***6***
Ingrese lado 1: ***7***
El área del triángulo es = 5.56

Ayuda: El área de un triángulo se puede calcular como $\sqrt{p(p-a)(p-b)(p-c)}$
siendo $p = (a + b + c) \div 2$

Ejercicio 4.2

Programar una función que reciba como parámetros un número real 'x' y otro entero positivo 'n' mayor que 0. Esta función deberá **retornar** la raíz enésima de x. Utilizarla en un programa que solicite el número real, la raíz y luego imprima el valor calculado.

Ejemplo:
Ingrese un número real: ***14***
Ingrese la raíz a calcular: ***3***
La raíz 3 de 14 es = 2.410142

Ayuda: Utilizar la función pow de la librería math.h cuyo prototipo es
double pow(**double** *base*, **double** *exponente*);

Ejercicio 4.3

Realizar una función que reciba como parámetros un número entero positivo (en sistema decimal) y otro entero positivo entre 2 y 9. Luego, la función deberá **retornar** el número decimal convertido a la base solicitada. Aplicar dicha función a un programa que solicite el número decimal, la base a convertir y luego imprima el valor calculado.

Ejemplo:
Ingrese el número decimal: ***527***
Ingrese la base: ***8***
527 (10) = 1017 (8)

Ejercicio 4.4

Programar una función que reciba como parámetros 2 números enteros y que **retorne** un número aleatorio entre estos 2 números (inclusive).

Por ejemplo, si se invocara la función con parámetros 1 y 6, deberá devolver un número aleatorio entre 1 y 6 inclusive. Validar que el valor máximo no sea superior al valor máximo que es capaz de generar la función rand.

Implementarla en un programa que solicite al usuario que ingrese el rango de valores aleatorios y la cantidad deseada de valor. A continuación imprimir la lista de números aleatorios según el criterio solicitado.

Ejemplo:
Ingrese mínimo: ***1***
Ingrese máximo: ***20***
Ingrese cantida de valores: ***10***
1, 8, 20, 14, 11, 17, 2, 1, 5, 6.

Funciones

Ejercicio 4.5

Dado el siguiente programa, construir las funciones cuyos prototipos se indican y de modo que el programa genere la salida correspondiente. Todas las funciones cuyos prototipos figuran deben ser programadas y utilizadas **sin modificar** los tipos de los parámetros ni el tipo de dato de retorno.

```
#include <stdio.h>

void dibujar_fila(int);
void dibujar_rectangulo(int, int);

int main()
{
        int alto, ancho;
        printf("Ingrese ancho:");
        scanf("%d", &ancho);
        printf("Ingrese altura:");
        scanf("%d", &alto);
        dibujar_rectangulo(ancho, alto);
        return 0;
}
```

Ejercicio 4.6

Dado el siguiente programa, construir las funciones cuyos prototipos se indican y de modo que el programa genere la salida correspondiente. Todas las funciones cuyos prototipos figuran deben ser programadas y utilizadas **sin modificar** los tipos de los parámetros ni el tipo de dato de retorno.

```
#include <stdio.h>

int ingresar_texto(void);
int es_letra(unsigned char);

int main()
{
         int len;
        printf("Ingrese texto (punto para finalizar):");
        len = ingresar_texto();
        printf("\nEl texto tiene %d letras.", len);
        return 0;
}
```

Ejercicio 4.7

Programar una funcion que tome como parametro un texto y que convierta sus letras mayusculas en minusculas y todo lo que no sean ni letras ni numeros, se conviertan en espacios. Utilizarla en un programa en el que se ingrese un texto de hasta 1000 caracteres y luego se imprima la version modificada del texto. El texto debe contener '\0' como caracter final a fin de que pueda imprimirse con printf("%s", texto);

Ejercicio 4.8

Realizar una funcion que permita ingresar en un arreglo de caracteres (string) solamente caracteres numericos (ignorar y no mostrar en pantalla cuando se ingrese un caracter que no sea numero). Luego, programar otra funcion que tome ese arreglo de caracteres y lo convierta a un valor

numerico entero. Utilizar ambas funciones en un programa en que se ingrese un numero y luego se convierta a entero para ser impreso mediante printf("%d", numero).

Ejercicio 4.9
Elabore un programa que cargue un arreglo con 100 números al azar entre 1 y 100. Luego obtener el valor máximo y el valor mínimo presentes en el arreglo.

Ejercicio 4.10
Realizar un programa que solicite al usuario una cantidad de números al azar que se generara entre el 0 y el 36. Presentar luego un informe que indique que cantidad de veces salio cada valor y el porcentaje que representa. No mostrar aquellos números que no hayan salido.
Ejemplo:
Ingrese cantidad de valores: ***10***
Informe:
El numero 8 salió 1 vez (10%).
El numero 14 salió 2 veces (20%).
El numero 20 salió 4 veces (40%).
El numero 23 salió 1 veces (10%).
El numero 30 salió 2 veces (20%).

Ejercicio 4.11
Realice una funcion que reciba como parametro un arreglo de hasta 10 enteros y luego calcule el desvio estandar de esos valores. Programar otra funcion que calcule el promedio de un arreglo. Aplicar ambas funciones en un programa en el que se carguen los valores del arreglo y luego imprima el desvio estandar de dicho arreglo.
Ayuda:
El desvio estandar (σ) es una medida estadistica de dispersion y la formula usual para su calculo es:

$$\sigma = \frac{\sqrt{\sum_{i=1}^{N} (x_i - m)^2}}{N}$$

N = Cantidad de valores
Xi = Cada uno de los valores
m = Es el promedio

Funciones

Ejercicio 4.12
Rehacer los ejercicios 3.3 y 3.4 de la práctica de arreglos, utilizando funciones. Preste especial atención al código que se repite; identifíquelo, transcribalo a una función y reescriba el main para utilizarla.

Ejercicio 4.13
Realizar un programa que contenga las siguientes funciones:
a. Una funcion que permita cargar un texto de hasta 1000 caracteres finalizando dicha carga con ENTER. El arreglo debe contener el caracter '\0' como marca de finalizacion de acuerdo con la convencion utilizada en el lenguaje C.
b. Una funcion que reciba como parametro una frase y que retorne cuantas palabras tiene esa frase.
c. Una funcion que reciba como parametro una frase y que retorne la longitud promedio de las palabras de esa frase.
d. Una funcion que reciba como parametro una frase y que modifique el contenido de dicha frase de modo que la primera letra de cada palabra quede en mayuscula.
En base a estas funciones, Realizar un programa en el que se ingrese una frase y a continuacion se muestren por pantalla los resultados obtenidos mediante las funciones creadas en los puntos b, c y d. Considere programar funciones adicionales.

Ejercicio 4. 14
Realizar un programa que contenga las siguientes funciones:

a. Una funcion que permita cargar un texto de hasta 1000 caracteres finalizando dicha carga con ENTER. El arreglo debe contener el caracter '\0' como marca de finalizacion de acuerdo con la convencion utilizada en el lenguaje C.
b. Una funcion booleana que reciba como parametros dos palabras (en dos arreglos independientes) y que indique si ambas palabras son iguales (considerar mayusculas y minusculas como iguales).
c. Una funcion que reciba como parametros una frase y una palabra (considerar mayusculas y minusculas como iguales) y que retorne cuantas veces aparece dicha palabra en la frase.
d. Una funcion booleana que reciba como parametros dos frases (en dos arreglos independientes) y que determine si ambas frases son iguales. Es decir, cuando contengan las mismas palabras en el mismo orden. Por ejemplo: "Hola, como te va?" debe ser considerada igual a "hola… Como te va!"

En base a estas funciones, Realizar un programa en el que se ingresen por teclado dos frases y una palabra. A continuacion, el programa debera mostrar por pantalla cuantas veces aparece la palabra ingresada en cada una de las frases ingresadas y si ambas frases son iguales. Considere programar funciones adicionales y utilizar la funcion del punto b en los puntos c y d.

Funciones

ANEXO: Generación de números aleatorios

```
#include<stdio.h>
#include<stdlib.h> /* Agrega las funciones "rand" y "srand"*/
#include<time.h> /* Agrega la funcion "time"*/
#define N 7
int main()
{
      int num;
      srand(time(NULL)); /* Genera la semilla utilizando la hora actual en
segundos */
      num=rand()%N; /* Genera un número al azar entre 0 y N-1 inclusive */
      printf("Numero generado al azar = %u\n", num);
      return 0;
}
```

La función rand() genera números pseudo aleatorios entre 0 y el valor almacenado en la constante RAND_MAX (imprimir esta constante en un programa a fin de determinar su valor). Por otro lado, la función srand genera la semilla necesaria para que la serie de números aleatorios no se repita entre ejecuciones. Observar el comportamiento del siguiente programa ejecutándolo varias veces. ¿Qué ocurre con las series de 5 números generadas en cada ejecución?

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define CANT 5
#define N 10
int main()
{
       int num, i;
      for(i=0;i<CANT;i++)
            { num=rand()%N;
            printf("Numero generado al azar = %u\n", num);
            }
      return 0;
}
```

Notar qué ocurre cuando se agrega la función srand(time(NULL)) al principio de la función main:

```
int main()
{
      int num, i;
      srand(time(NULL));
      for(i=0;i<CANT;i++)
            {
            num=rand()%N;
            printf("Numero generado al azar = %u\n", num);
            }
return 0;
}
```

Por último, determinar qué ocurre y por qué, cuando se coloca el llamado a la función srand dentro del ciclo for.