

ICOMDS-OF

Guía 2: File I/O, Random, Math, Funciones, Errores, Recursión

Ejercicio 1

Abra el archivo `texto_español.txt` proporcionado. Lea su contenido completo y guárde la información en un string. A partir del mismo, resuelva:

- ¿Cuántos caracteres tiene en total el archivo?
- ¿Cuántas palabras? ¿Cuántas oraciones?
- Cuente cuántas veces aparece cada letra en el mismo. Si puede, dibuje de alguna forma su distribución.

Una vez resuelto, escriba el resultado en un archivo llamado `solucion_ej1.txt`.

Ejercicio 2

Escriba una función que tome como entrada un vector de N nombres de usuarios y devuelva uno al azar (... un sorteador de instagram!). Si lo desea, puede cargar los nombres desde `nombres.txt`.

Ejercicio 3

Escriba una función que permita obtener el perímetro, área y volumen de un paralelepípedo recto de lados A , B , C .

Ejercicio 4

Escriba una función que permita encontrar el cero o raíz de una función $f(x)$ entre dos valores a y b utilizando el método de la bisección. La misma debe entregar un error si $\text{signo}(f(a)) = \text{signo}(f(b))$, o si $a > b$.

Ejercicio 5

Escriba una función que permita calcular la integral de una función $f(x)$ entre a y b , separando en N intervalos, y utilizando la regla del trapecio (https://es.wikipedia.org/wiki/Regla_del_trapecio) (https://es.wikipedia.org/wiki/Regla_del_trapecio). La misma debe entregar un error si $a > b$.

Ejercicio 6

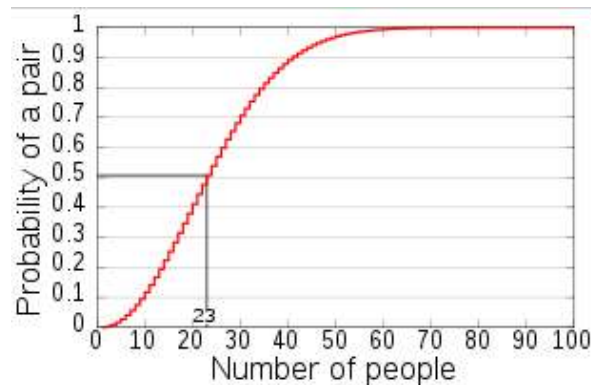
Escriba una función que permita calcular el factorial de un número de forma recursiva.

Ejercicio 7

Para poder obtener la expresión de un número en una base, se puede utilizar un algoritmo recursivo. Escriba un programa que pida al usuario un número y una base e imprima el mismo allí. Utilizar caracteres del abecedario si la base es mayor a 10.

Desafío 1: El problema de los cumpleaños.

Supongamos que hay N personas adentro de una habitación. ¿Cuál es la probabilidad de que *al menos dos* de esas personas cumplan años el mismo día? Sorprendentemente, tiene esta forma (https://en.wikipedia.org/wiki/Birthday_problem)(https://en.wikipedia.org/wiki/Birthday_problem):



Es bastante difícil demostrar eso calculándolo. Pero es razonablemente fácil programarlo. Para ello, vamos a disponer de *experimentos simulados* (¡Monte Carlo!):

- Para grupos de N personas, con N desde 2 hasta 50:
 - Realice 1000 experimentos:
 - Inicialice el cumpleaños de N personas, de forma aleatoria, con un número entre 0 y 364.
 - Si dos personas cumplen el mismo día, el experimento da Sí. Si no, el experimento da No.
 - Divida la cantidad de casos que dieron Sí por la cantidad de experimentos (1000).
 - Imprima la probabilidad de este caso.
- Guarde los resultados de probabilidades en un archivo `cumpleaños.txt`

Desafío 2: La generala

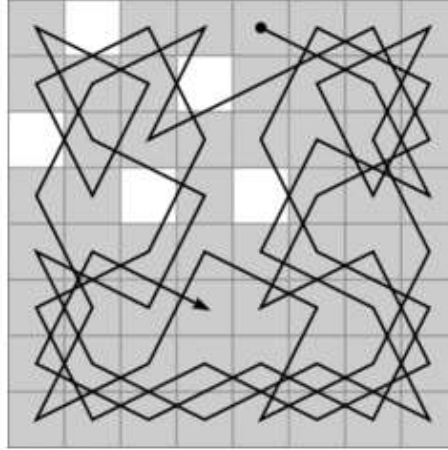
¡Programemos un juego de dados, la generala! Para eso utilizaremos muchos de los conceptos que venimos trabajando. Es un juego conocido por todos. Las reglas y detalles pueden encontrarse en <https://es.wikipedia.org/wiki/Generala> (<https://es.wikipedia.org/wiki/Generala>).

¿Cómo lo programamos? Pensemos en el algoritmo:

- Inicialización:
 - El juego debe tener una inicialización, que pida la cantidad de jugadores. Estos deben ser entre 1 y 5.
 - Luego, el juego debe pedir nombres de los jugadores.
 - Se inicializa la tabla de puntajes en cero.
- Desarrollo:
 - El primer jugador debe tirar cinco dados. Luego, selecciona cuales desea volver a tirar y cuales no.
 - Si guarda todos, su jugada termina y anota lo que corresponde en la tabla de puntajes.
 - Si no puede o no quiere anotar un puntaje, puede anularse una casilla de la tabla de puntajes
 - Si no guarda al menos uno, vuelve a tirar.
 - Una vez concluido este turno, le toca al segundo jugador, etc.
- Finalización:
 - Se suman los puntajes y se ordena de mayor a menor.

Desafío 3: The knight's tour

El camino del caballero (https://en.wikipedia.org/wiki/Knight%27s_tour) es una serie de movimientos de caballo que cubre todos los casilleros de un tablero de ajedrez. Este camino es bastante difícil de encontrar a mano, pero una computadora puede simplificarlos la existencia.



Para ello, necesitaremos un *tablero*, que es un vector de vectores de $L \times L$ elementos. Cada uno de estos elementos es una posición i, j , que representa las coordenadas de nuestro caballo. Naturalmente, el mismo puede moverse según las reglas del ajedrez: dos casilleros en una dirección, uno en la otra.

Sin dudas, para resolver este problema necesitará utilizar el concepto de *Divide y conquista*, y muchas funciones y arreglos.