* 01-EJERCICIOS
  + PALINDROMO : FGETS() CUIDADO QUE GUARDA HASTA \N
  + TABLAMULTIPLICAR : DOS VERSION DE FORK , UNO PADRE ENVIA MENSAJE TRAS EL FIN DE CUALQUIER HIJO. WAIT(), WATPID()
* EJEMPLOC
  + BINARIO : HAY EXPLICACION DE USO DE FOPEND,FWRITE…
  + SIGNAL : USO DE METODO KILL();
* 03-LLAMADAS
  + REDIRECCION : INCLUYE EJECUCION DE PROGRAMA CON EXECVP(); Y REDIRIGIR CON DUP2();
  + PIPEECONREDIRECCION : HAY EJEMPLO DE USO DE SPRINTF(); Y USO DE SCANF();
* 05-ENTRENAMIENTO
  + ENTRENAMIENTO5 EJEMPLO FORK CON SIGNAL Y EXECVP
  + ENTRENAMIENTO7 DOS PIPE EN UN FORK()

TIPS:

SI QUIERES COPIAR UNA CADENA A UNA ARRAY DE CHAR SE USA STRCPY(CHAR[],CADENA);

write(int fd, const void \*buf, size\_t count);

* **fd**: Es el descriptor de archivo en el cual se escribirán los datos. Puede ser el descriptor de archivo de un archivo, un socket, un pipe, entre otros.
* **buf**: Es un puntero al búfer que contiene los datos que se escribirán.
* **count**: Es el número de bytes que se deben escribir.

scanf en lugar de fgets para leer el número desde la entrada estándar.

scanf("%d", &numero\_a\_enviar); fgets(cadena,sizeof(cadena),stdin);

Necesitas utilizar funciones como **sprintf** para construir una cadena formateada.

char nombreArchivo[20]; // Suficientemente grande para "datos" + número + ".txt"

// sprintf Parámetros: La cadena de destino, la cadena de formato y los datos a guardar

sprintf(nombreArchivo, "datos%d.txt", i);

// Inicializar la semilla del generador de números aleatorios

srand(time(NULL));

int numero\_aleatorio = rand() % 10 + 1; // Genera un número entre 1 y 10

//atoi metodo para pasar un string a int

int n = atoi(argv[1]);