

1. (2.0 puntos) Se dispone del archivo binario **sinonimos.dat** que contiene una palabra con sus sinónimos asociados (MAX_SINONIMOS como máximo). La estructura de los registros del archivo es la siguiente:

```
typedef _sinonimos_t struct{
    char palabra[TAM_PALABRA+1];
    char cant_sinonimos;
    char [MAX_SINONIMOS][TAM_PALABRA+1] sinonimos;
} sinonimos_t;
```

Implemente un programa que a partir de la información contenida en el archivo binario genere un archivo de texto **sinonimos.txt** con la misma información. En cada línea del archivo de texto debe quedar la palabra con sus sinónimos asociados, separados por el carácter '| '.

2. (1.0 punto) Para cada inciso indique si es verdadero o falso.
- Las variables static tienen visibilidad como las variables globales.
 - Se puede asignar una variable de tipo double a una variable float sin inconvenientes.
 - Es posible asignar tanto true/false como 1/0 a una variable de tipo bool.
 - Para regresar de la ejecución de una función es necesario ejecutar la instrucción return.
 - El lenguaje C no soporta pasaje de parámetros por referencia.
3. (1.8 puntos) Implemente la biblioteca **matriz**, la cual contiene funciones para una matriz cuadrada de números flotantes. La misma debe contener las siguientes funciones:
- crear**: reserva memoria para una matriz de n x n
 - suma**: calcula la suma de todos los elementos de la matriz
 - liberar**: libera la memoria reservada en el punto a)

Implemente las funciones indicando claramente el nombre del archivo donde se encuentran los prototipos, la declaración de tipos y donde va la implementación de las mismas.

4. (1.0 puntos) Para cada inciso indique si es verdadero o falso.
- No es posible determinar la posición de memoria en la que se encuentra un campo de una variable de tipo struct..
 - La función **ftell** con parámetro **SEEK_END** permite posicionarse al final de un archivo.
 - La función **fflush** permite borrar el contenido de un archivo.
 - Los archivos abiertos en modo "a" o "ab" descartan el contenido actual del mismo.
 - La función **fprintf** se puede utilizar tanto para escribir en un archivo como para escribir en pantalla.
5. (0.8 puntos) Marque con verdadero o falso si las siguientes expresiones son correctas (V) o generan algún tipo de error o warning (F), teniendo en cuenta las declaraciones: **char *p;** **char c;** **char s[10];**
- | | |
|--------------------|-------------------|
| a) *(&c) = 'a'; | b) free(*p); |
| c) scanf("%c", c); | d) p[0] = *(s+0); |
| e) c = "z"; | f) p = &s; |
| g) p = s+5; | h) p = &c; |

6. **(1 punto)** Defina una macro que obtenga el máximo de 2 números y utilícela en la definición de una segunda macro que obtenga el máximo de 4 números.

7. **(0.8 puntos)** a) Indique qué imprime el siguiente código:

```
for (i = 0; i < 3; i++){  
    #define FLAG i  
    #if FLAG == 0  
        printf("Flag 0\n");  
    #elif FLAG == 1  
        printf("Flag 1\n");  
    #else  
        printf("Flag 2\n");  
    #endif
```

8. **(1.6 puntos)** Escriba un programa que reciba una secuencia de números enteros como argumentos como argumentos a la función main. El programa debe calcular el promedio de dichos números. En caso de que el programa no reciba al menos un número debe imprimir un mensaje de error.