Pregunta 1

Un archivo contiene un diccionario en el siguiente formato:

casa:edificación construida para ser habitada:
lluvia:condensación del vapor de agua contenida en las nubes:
embarcación:todo tipo de artilugio capaz de navegar sobre o bajo el agua:
alimento:sustancia ingerida por un ser vivo:
… etc ...

El primer carácter ':' separa la palabra de su definición. El segundo ':' termina la definición. Todas las líneas del archivo tienen un número fijo de caracteres para que sea sencillo hacer acceso directo con *fseek*. Las palabras están *desordenadas* en el archivo. Ud. no puede usar búsqueda binaria. Este archivo no cabe en la memoria del computador.

Programe la siguiente función:

*void modificar(char *nom_dic, char *palabra, char *def,*
int n_lin, int ancho);

Esta función cambia la definición de *palabra* por *def* en el diccionario almacenado en el archivo *nom_dic*. El parámetro *n_lin* es el número de líneas del archivo (y por lo tanto el número de palabras y definiciones) y *ancho* es el número de caracteres de cada línea en el archivo.

A modo de recuerdo, ésta es la API para manejar archivos:

```
FILE *fopen(char *nom_arch, char *modo);   /* modo es "r+" para lect/escr  */
size_t fread(char *buf, size_t ancho_item, int n_item, FILE *file);
size_t fwrite(char *buf, size_t ancho_item, int n_item, FILE *file);
int fseek(FILE *file, long despl, int w);   /* w==SEEK_SET */
int fclose(FILE *file);
```

Pregunta 2

Las acciones de la empresa ACME se transan en una bolsa de comercio cuyos operadores son representados mediante threads. Para comprar o vender una acción los operadores usan las funciones *compro* y *vendo*. Sus encabezados son:

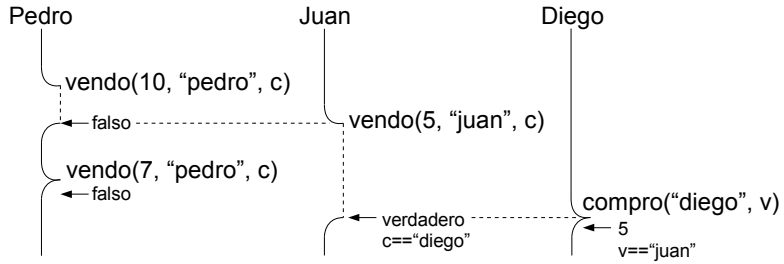
```
int vendo(int precio, char *vendedor, char *comprador);
int compro(char *comprador, char *vendedor);
```

En el cuado siguiente se muestra a la izquierda el código usado por los múltiples threads vendedores de acciones y a la derecha el código de los múltiples compradores.

```
char *nom= miNombre();
int precio= miPrecio();
char comprador[100];
if (vendo(precio, nom,
          comprador)) {
  printf("vendi a %s\n",
         comprador);
}
```

```
char *nom= miNombre();
char vendedor[100];
sleep(tiempoAleatorio());
int precio= compro(nom, vendedor);
if (precio>0) {
  printf("compre a %s en %d\n",
         vendedor, precio);
}
```

La función *compro* transa una sola acción con el vendedor más barato de ese momento, retornando el precio pagado y copia el nombre del vendedor en el *2do parámetro*. Si en ese momento no hay ningún vendedor el precio será 0 y *compro* retorna de inmediato. La función *vendo* ofrece una acción al precio indicado y espera hasta que (i) aparezca un comprador, en cuyo caso retorna verdadero y copia el nombre del comprador en el 3er parámetro, o (ii) aparezca (o ya hay) un vendedor con un precio menor, retornando falso en tal caso. El siguiente diagrama muestra un ejemplo de ejecución.



Pedro llama a *vendo*, que retorna falso cuando Juan llama a *vendo* con un precio menor. La segunda llamada de Pedro fracasa de inmediato porque su precio todavía es mayor al de Juan. Juan sí tiene éxito cuando Diego llama a *compro* y por lo tanto la llamada a *vendo* de Juan retorna verdadero, copiando el nombre "diego" en el parámetro *comprador*. Por su parte *compro* retorna 5 (el precio) y copia "juan" en el parámetro *vendedor*.

Programe las funciones *vendo* y *compro*. Para la sincronización debe usar un monitor (es decir, un mutex y una condición). Observe que nunca habrá más de un vendedor en espera. Use variables globales.