Pregunta 1

El problema de la suma de subconjuntos es este: dado un conjunto *a* de *n* enteros, ¿existe algún subconjunto de *a* no vacío cuya suma sea exactamente cero? Por ejemplo, dado el conjunto { −7, −3, −2, 5, 8}, la respuesta es sí, porque el subconjunto { −3, −2, 5} suma cero. Considere 3≤n≤64. La siguiente solución toma tiempo $O(n \cdot 2^n)$.

```
typedef unsigned long long Set;
Set buscar(int a[], int n) {
  Set comb= (1<<(n-1)<<1)-1; // 2^n-1: nº de combinaciones
  for (Set k= 1; k<=comb; k++) {
    // k es el mapa de bits para el subconjunto { a[i] | bit k_i de k es 1 }
    long long sum= 0;
    for (int i= 0; i<n; i++) {
      if ( k & ((Set)1<<i) ) // si bit k_i de k es 1
        sum+= a[i];
    }
    if (sum==0) {   // éxito: el subconjunto suma 0
      return k;     // y el mapa de bits para el subconjunto es k
  } }
  return 0;         // no existe subconjunto que sume 0
}
```

En la función *buscar* los subconjuntos de *a* se representan mediante una máscara de bits $k= k_{n-1} \ldots k_1 k_0$. El elemento *a[i]* está en el subconjunto si k_i es 1. Esta función recorre los 2^n-1 subconjuntos posibles (se descarta el subconjunto vacío) hasta encontrar un subconjunto que sume 0, en cuyo caso se retorna su mapa de bits. Retorne 0, si ningún subconjunto suma 0.

Reprograme completamente la función *buscar* de modo que la búsqueda se haga paralelamente en 8 threads. Si hay múltiples subconjuntos que suman 0, entregue cualquiera de ellos.

Pregunta 2

El cuadro de abajo a la izquierda muestra el contenido de un archivo que representa un diccionario no ordenado. El cuadro de la derecha es el mismo diccionario en donde se ha eliminado la palabra "*alimento*".

casa	edificacion construida para ser habitada
alimento	sustancia ingerida por un ser vivo
celular	aparato portatil de un sistema de telefonia celular
gato	felido domestico
radio	receptor de ondas electromagneticas para audio

casa	edificacion construida para ser habitada
radio	receptor de ondas electromagneticas para audio
celular	aparato portatil de un sistema de telefonia celular
gato	felido domestico

Cada línea del archivo es de 80 caracteres y almacena una definición. Los primeros 10 caracteres corresponden a la palabra definida, luego vienen 69 caracteres para su definición y se termina la línea con un *\n* para completar los 80 caracteres. El archivo podría contener 0 o más líneas en blanco al final (y solo al final).

Programe la función *void eliminar(char *arch, char *pal)* que elimina la palabra *pal* del diccionario almacenado en el archivo de nombre *arch*. Para eliminar la palabra Ud. debe mover la última definición del diccionario hacia la posición de la palabra eliminada.

Restricción: El diccionario no cabe en la memoria. Use *fseek* astutamente para reducir la lectura/escritura del archivo. Sin embargo, la búsqueda de la palabra debe ser secuencial. Tenga cuidado en considerar que *strcmp("casa", "casa ")* es distinto de 0.

Pregunta 3

Un pub posee un único baño que debe ser compartido por damas y varones. El baño es amplio y admite un número ilimitado de personas. El problema consiste en evitar que las damas se encuentren con los varones dentro del baño. Las damas son representadas por threads que solicitan entrar al baño invocando la función *entrarDama* y notifican su salida llamando a *salirDama*, análogamente los threads varones invocan *entrarVaron* y *salirVaron*. Las funciones *entrarDama* y *entrarVaron* deben esperar cuando en el baño se encuentran personas del sexo opuesto. La siguiente es una solución incorrecta e ineficiente para la programación de estas 4 funciones:

```
int damas= 0, varones=0;
```

```
void entrarVaron() {
  while (damas>0)
    ;
  varones++;
}

void salirVaron() {
  varon--;
}
```

```
void entrarDama() {
  while (varones>0)
    ;
  damas++;
}

void salirDama() {
  damas--;
}
```

A pesar de todo esta solución funciona correctamente el 99,9% de las veces. Reprograme el código de más arriba de manera correcta y eficiente. No importa que su solución sufra de hambruna.

Ayuda: el código de más arriba está casi bueno, no cambie nada, solo agregue el código faltante.