SSL – Ing Pablo Méndez – Ejercicios tipo PARCIAL

1. Escriba una expresión regular que represente al lenguaje definido por comprensión como:
   $L = \{ a^n bc^m / n \geq 2 \text{ y } m \geq 0\}$.
   Solución: aaa*bc*
2. Sea el LF infinito $L = \{a^n bc^n b / n \geq 1\}$. Describa la Definición Formal de una gramática que genera este LF. Indique si se trata de una GR o una GIC
   Solución: G= ({S, T}, {a, b, c}, {S -> Tb, T -> aTc, T -> abc}, S). Es una GIC

3. Confeccione un programa en C que reciba por línea de comandos:
   a. Una palabra a buscar.
   b. Ruta y nombre de un archivo de texto y cuente la cantidad de veces que aparece la palabra en el archivo mencionado indicándola por pantalla.

   Funciones que puede utilizar para este ejercicio:
   ```
   int fscanf ( FILE * stream, const char * format, ... );
   int strcmp ( const char * str1, const char * str2 );
   FILE * fopen ( const char * filename, const char * mode );
   int fclose ( FILE * stream );
   ```
   Una posible solución:
   ```
   int main(int argc, char *argv[])
   {
       //printf("%s %s", argv[1], argv[2]);
       FILE *fpuntero;
       char str[100];
       int cont = 0;
       if (argc<3)
       {
           printf("Error: faltan parametros\n");
           return 1;
       }
       fpuntero = fopen(argv[2], "r");
       if(fpuntero == NULL) {
           printf ("Error al intentar abrir el archivo\n");
           return 1;
       }

       while (fscanf(fpuntero,"%s", str) != EOF)
       {
             if (strcmp(str, argv[1])==0)
               cont++;
       }
       fclose(fpuntero);
       printf("Cantidad de apariciones de '%s' = %d\n",argv[1],cont );
       return 0;
   }
   ```

4. Escriba un programa en C que reciba por parámetro dos cadenas de texto (arrays de caracteres) y encripte la primer cadena1 sumándole los caracteres de la segunda cadena. Si la cadena 2 es de mayor longitud que la cadena 1 debe ser comenzar a sumarse nuevamente desde el carácter cero de la cadena 2. El resultado puede imprimirse directamente en pantalla.
   Ejemplo:
   Encriptar hola casa
   Una posible solución:
   ```
   int main(int argc, char *argv[])
   {
       int i = 0;
       char s[100];
       strcpy(s, argv[1]);
       printf("s tiene %s.\n",s);
       while (s[i]!='\0')
       {
           //printf("%c + %c \n",s[i],argv[2][i % strlen(argv[2])]);
           s[i] += argv[2][i % strlen(argv[2])] ;
           i++;
       }
       printf("%s\n",s);
       getch();
       return 0;
   }
   ```
   Responder V o F:
   - Una expresión regular puede especificar uno o más lenguajes.
   - En una gramática regular el lado izquierdo debe tener un solo NO TERMINAL sin embargo el lado derecho puede tener más de un NO TERMINAL.