UNIVERSIDAD DE MÁLAGA

Procesadores de Lenguajes

PARCIAL 3 (2023)

Implementar mediante JFlex y CUP un programa que realice una serie de operaciones sobre listas de números enteros positivos. Las listas de números vendrán delimitadas por paréntesis y los números separados por comas. Por ejemplo `(1, 4, 56)`.

El fichero de entrada puede contener varias lista.

Apartado 1. (2,5 puntos) Si una lista viene precedida de la letra 'C' entonces el programa mostrará la lista en la salida sustituyendo las secuencias de valores iguales consecutivos por el número de repeticiones del valor seguido del valor. Por ejemplo, `C(3, 3, 3, 3)` se mostraría en la salida del programa como `(4, 3)`. Esto es, 4 repeticiones del valor 3. En el caso de `C(4, 4, 5, 5, 5, 6)` se mostraría en la salida `(2, 4, 3, 5, 1, 6)`. Esto es, 2 repeticiones del número 4; 3 del número 5; y 1 del número 6.

La operación C debe permitir que los elementos de las listas sean, a su vez, listas. En este caso, cuando un elemento de la lista a comprimir sea, a su vez, una lista, entonces se mostrará la sublista comprimida. Pero no se comparará con el siguiente elemento. Esto es, aunque haya dos lista idénticas consecutivas se comprimirán de forma independiente. De esta forma, solo se comprimen los números dentro de una lista, no listas de listas. Por ejemplo, `C( 6, 6, (6, 6, 6), (6, 6, 6))` mostraría `(2, 6, (3, 6), (3, 6))`.

Apartado 2. (2.5 puntos) Si una lista viene precedida de la letra `'D'` entonces el programa mostrará la lista en la salida sustituyendo cada par de valor por las repeticiones del segundo valor tantas veces como indique el primer valor. Esto es, realizará la operación opuesta a la operación `'C'`. Por ejemplo, `D(2, 3)` se mostraría como `(3, 3)`. Esto es, 2 repeticiones del valor 3. En el caso de `D( 4, 2, 1, 6)` se mostraría `(2, 2, 2, 2, 6)`. Si una lista no tuviese un número par de elementos entonces se generaría un error sintáctico y se detendría el análisis.

En el caso de la operación `D`, cuando un elemento de la lista sea una lista, la sublista no vendrá precedida del número de repeticiones. Por ejemplo, `D( 3, 2, (2, 4))` mostraría `(2, 2, 2, (4, 4))`. De esta forma, la operación `D` se corresponderá con la operación opuesta de la operación `C`.

Apartado 3. (2.5 puntos) Si una lista viene precedida de la letra 'I' entonces el programa mostrará la lista en la salida mostrándola en orden invertido. Esto es, el primer elemento de la lista en primer lugar, el penúltimo en segundo lugar y así sucesivamente hasta el primer elemento de la lista que se mostraría en último lugar. Por ejemplo, `I( 1, 2, 3)` se mostraría en la salida `(3, 2, 1)`.

La operación `I` también permitirá que un elemento de la lista sea, a su vez, una sublista e invertirá también los valores de la sublista. Por ejemplo, `I (1, (2, 3), 4)` mostrará en la salida `(4, (3, 2), 1)`.

Apartado 4. (2,5 puntos) Si una lista viene precedida de la letra `'L'` entonces el programa mostrará la lista en la salida seguida por dos puntos y la longitud de la lista. Por ejemplo, `L(1, 2, 30)` mostrará en la salida `(1, 2, 30):3` puesto que 3 es la longitud de la lista `(1, 2, 30)`.

La operación `L` también deberá permitir que los elementos de la lista sean, a su vez, listas. En este caso, también se mostrará la longitud detrás de cada sublista. Por ejemplo, `L( 1, 2, (3, 4), 5)` mostraría `(1, 2, (3, 4):2, 5):4` ya que la lista interior tiene dos elementos y la lista principal tiene 4 elementos (la lista interior cuenta como un elemento más de la lista principal).

NOTAS:

- **NO ES OBLIGATORIO USAR LOS FICHEROS DE AYUDA.** Es decir, estos ficheros se proporcionan como ayuda para facilitar la implementación, pero no es imprescindible usarlos. También se pueden modificar si hace falta, como cada cual lo crea conveniente, a fin de cumplir con las especificaciones del enunciado.
- Como resultado de este ejercicio deben enviarse TODOS los ficheros necesarios para la compilación del compilador comprimidos en un fichero denominado **vectores.zip**, (tenga cuidado de no incluir ficheros innecesarios que puedan dificultar la compilación automática). Para obtener el fichero comprimido puede usar la siguiente instrucción:
  `zip vectores.zip Vectores.flex Vectores.cup Vectores.java`
- La corrección de este ejercicio se hace en función de varios casos de prueba independientes, similares, pero NO IGUALES, a los que se incluyen en el fichero comprimido. La correcta compilación de los casos de prueba que se dan en el fichero **vectores-init.zip** es condición necesaria, pero no suficiente para superar las pruebas.
- El programa se deberá poder ejecutar con uno o dos parámetros. El primer parámetro corresponderá al fichero con el programa a interpretar y el segundo parámetro que será opcional corresponderá al fichero donde se guardará la salida del programa. Si solo se ejecutase con un parámetro entonces mostraría la salida por consola. Por tanto, en Unix, cualquiera de las siguientes ejecuciones del programa serían válidas:

```
java -cp /usr/share/java/cup/cup_runtime.jar:. Vectores entrada.vec
java -cp /usr/share/java/cup/cup_runtime.jar:. Vectores entrada.vec salida.res
```