INF-253 Lenguajes de Programación
Tarea 1: Python

Profesor: José Luis Martí Lara
Ayudante Cátedras: Sebastián Godínez San Martín
Ayudante Tareas: Gabriel Carmona Tabja

Martes 5 de Marzo de 2019

1. Translatext

Translatext es una idea para un traductor de pseudocódigo estándar.

El alumno aplicará conceptos de **Expresiones regulares** para realizar dicho programa, dado que deberá analizar gramaticalmente el código para su traducción.

2. Requerimientos

- El alumno debe utilizar Python 3 para el desarrollo de esta tarea.
- El programa debe utilizar expresiones regulares para el parseo del código. Pueden consultar la documentación del módulo **re** en el siguiente `https://docs.python.org/3/library/re.html`
- El programa recibirá un archivo llamado pseudocode.txt, que contiene el código a traducir.
- El programa debe entregar un archivo llamado pseudocodigo.txt, el cual contendrá el texto traducido.

3. ¿Que debe poder hacer Translatext?

Translatext debe poder traducir un pseudocódigo del ingles al español, donde si en alguna línea su sintaxis está incorrecta se deberá indicar al final con un #Error de esa misma línea que contiene un error. La línea que contenga error no debe ser traducida.

4. Introducción a Pseudocódigo

Se definirá una base de pseudocódigo estándar con su respectiva traducción, cualquier otra variación de un comando que signifique lo mismo es considerada mal escrita.

4.1. Estructura Básica

Comienzo y fin de un programa

```
Original:                              Translate:
PROCEDURE <Nombre Del Programa>         PROCEDIMIENTO <Nombre Del Programa>
```

```
<Codigo a ejecutar>                    <Codigo a ejecutar>
END PROCEDURE                          FIN PROCEDIMIENTO
```

Al ser *procedure\end procedure* comandos para indicar el inicio y el final de un programa, lo que haya antes y despues de esto serán consideradas mal escritas. Se puede contener más de un programa por archivo. Si no hay comienzo y final del programa o esta mal escrito el comienzo o final del programa deberá escribir en el archivo pseudocodigo.txt NO HAY PROGRAMA.

4.2. Declaración e Inicialización de Variables

- **SET**: asigna un valor a una variable
- **INITIALIZE**: inicializa una variable

```
Original:                              Translate:
SET <variable> TO <expresion>          FIJA <variable> EN <expresion>
INITIALIZE <variable>                  INICIALIZA <variable>
```

4.3. Operadores

4.3.1. Matemáticos

- **SUBSTRACT**: permite la resta entre dos expresiones
- **SUM**: permite la suma entre dos expresiones
- **MULTIPLY**: permite la multiplicación entre dos expresiones
- **DIVIDE**: permite la división entre dos expresiones

```
Original:                              Translate:
SUBTRACT <expresion> TO <expresion>    RESTA <expresion> A <expresion>
SUM <expresion> TO <expresion>         SUMA <expresion> A <expresion>
MULTIPLY <expresion> BY <expresion>    MUTIPLICA <expresion> POR <expresion>
DIVIDE <expresion> BY <expresion>      DIVIDE <expresion> POR <expresion>
```

4.3.2. Booleanos

- **AND**: verifica si las expresiones son todas verdaderas
- **OR**: verifica si alguna de las expresiones son verdaderas
- **NO**: niega una expresión
- **EQUAL TO**: si es que una expresión es igual a otra
- **DIFFERENT TO**: si es que un expresión es distinta a otra
- **GREATER**: decide cual es mayor entre dos valores
- **SMALLER**: decide cual es menor entre dos valores

Original:	Translate:
<expresion> AND <expresion>	<expresion> Y <expresion>
<expresion> OR <expresion>	<expresion> O <expresion>
NO <expresion>	NO <expresion>
<expresion> EQUAL TO <expresion>	<expresion> IGUAL A <expresion>
<expresion> DIFFERENT TO <expresion>	<expresion> DIFERENTE A <expresion>
GREATER <val/var> THAN <val/var>	MAYOR <val/var> QUE <val/var>
SMALLER <val/var> THAN <val/var>	MENOR <val/var> QUE <val/var>

OJO: *val/var* significa valor o variable.

4.4. Condicional

Original:	Translate:
IF <condicion> THEN	SI <condicion> ENTONCES
<codigo>	<codigo>
ELSE	SINO
<codigo>	<codigo>
END IF	FIN SI

OJO 1: puede haber condicionales anidados.
OJO 2: si es que falta la línea de apertura, el ELSE o la de cierre se considera mal escrito. Deben indicar el error en las otras líneas del condicional.

4.5. Loops

Original:	Translate:
WHILE <condicion>	MIENTRAS QUE <condicion>
<codigo>	<codigo>
END WHILE	FIN MIENTRAS

OJO: puede haber loops anidados.
OJO 2: si es que falta la línea de apertura o la de cierre se considera mal escrito. Deben indicar el error en la línea de apertura o cerradura, depende de cual es la que se encuentre presente.

4.6. Input y Output

- **READ**: entrada
- **PRINT**: salida

Original:	Translate:
READ <variable>	LEER <variable>
PRINT <expresion>	IMPRIME <expresion>

5. Ejemplo

5.1. Ejemplo 1

```
pseudocode.txt                                  pseudocodigo.txt
-------------------------------------------------------------------------------
PROCEDURE Tarea1                                PROCEDIMIENTO Tarea1
INITIALIZE dab                                  INICIALIZA dab
SET dab TO 10                                   FIJA dab EN 10
SET dab TO SUBTRACT SUM 10 TO 5 TO dab          FIJA dab EN RESTA SUMA 10 A 5 A dab
IF GREATER dab THAN 5 THEN                      SI MAYOR dab QUE 5 ENTONCES
WHILE NO 1 EQUAL TO dab                         MIENTRAS QUE NO 1 IGUAL A dab
READ XD                                         LEER XD
PRINT XD                                        IMPRIME XD
SUBTRACT 1 TO dab                               RESTA 1 A dab
END WHILE                                       FIN MIENTRAS
ELSE                                            SINO
PRINT "AYUDA"                                   IMPRIME "AYUDA"
END IF                                          FIN SI
END PROCEDURE                                   FIN PROCEDIMIENTO
```

5.2. Ejemplo 2

```
pseudocode.txt                                  pseudocodigo.txt
-------------------------------------------------------------------------------
PROCEDURE Tarea1                                PROCEDIMIENTO Tarea1
INITIALIZE dab                                  INICIALIZA dab
SET tomc TO FALSE                               FIJA tomc EN FALSE
SE dab TO SUM 10                                SE dab TO SUM 10 #Error
IF GREATER dab THAN 5 THEN                      IF GREATER dab THAN 5 THEN #Error
WHILE NO 1 EQUAL TO dab                         WHILE NO 1 EQUAL TO dab #Error
READ XD                                         LEER XD
PRINT XD                                        IMPRIME XD
SUBTRACT 1 TO dab                               RESTA 1 A dab
PRINT "AYUDA"                                   IMPRIME "AYUDA"
END IF                                          END IF #Error
END PROCEDURE                                   FIN PROCEDIMIENTO
```

6. Datos de Vital Importancia

- Las variables no pueden incluir números al principio, ni caracteres especiales en cualquier parte del nombre.
- Los valores pueden ser número, caracteres, strings o booleanos (True o False).
- Las expresiones de la sección 4.2 pueden ser variables, valores o cualquier operador de la sección 4.3.
- Las expresiones de la sección 4.3.1 solo pueden ser variables, operadores matematicos (4.3.1) o valores.

- Las expresiones de la sección 4.3.2 solo pueden ser variables, operadores booleanos (4.3.2) o valores.
- Las condiciones de la sección 4.4 y 4.5 solo pueden ser variables, operadores booleanos (4.3.2) o valores.
- Hay que notar que cualquier palabra clave de los comandos mostrados anteriormente no puede ser usado para las variables.

7. Sobre Entrega

- Se debera entregar un programa llamado translatext.py.
- Las funciones implementadas deben ser comentadas.
- Se debe trabajar en grupos de dos personas.
- **La entrega debe realizarse en zip y debe llevar el nombre: Tarea1LP_RolIntegrante-1_RolIntegrante-2.tar.gz**
- **El archivo README.txt debe contener nombre y rol de los integrantes del grupo e instrucciones detalladas para la correcta utilización de su programa.**
- **El no cumplir con las reglas de entrega conllevará un descuento maximo de 30 puntos en su tarea.**
- La entrega será vía moodle y el plazo máximo de entrega es hasta el **Martes 26 de Marzo a las 23:55 hora moodle**.
- Por cada día de atraso se descontarán 20 puntos (10 puntos la primera hora).
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.

8. Calificación

- Código (70 puntos)
  - Orden (5 puntos)
  - Expresiones Regulares (25 puntos)
  - Implementación de la revisión de la sintaxis del pseudocódigo (40 puntos)
- Output (30 puntos)
  - Correcta Traducción (15 puntos)
  - Correcta detección de errores (15 puntos)
- No usa expresiones regulares (-100 puntos)
- No respeta formato de entrega (-20 puntos)