

Segundo examen parcial

Suponga que se desea agregar una nueva funcionalidad a un lenguaje de programación que ya existe. La funcionalidad consiste en que al invocar un método este pueda devolver varias variables (no uno como usualmente se usa).

Para simplificar la implementación, considere las siguientes restricciones:

1. Existe un único método que se invoca para solicitar al usuario los valores que se desean retornar.
2. El método sólo recibe un parámetro de entrada de tipo entero que indica la cantidad de valores que se retornarán.
3. El valor del parámetro de entrada deberá estar entre 1 y 5.
4. Los valores de retorno serán todos de tipo entero.
5. El método (qué más parece una macro), lo único que hará será pedir al usuario tantos números enteros como lo indique el parámetro de entrada.
6. Además de invocar el método, debe ser posible asignar cada uno de los valores de retorno a una variable local diferente (se puede dar por supuesto que las variables son enteras).
7. No es posible hacer uso de variables globales en su solución.
8. Deberá generar código MIPS que se pueda ejecutar sin errores en una máquina virtual (SPIM).

Por ejemplo, el siguiente es un código válido de entrada para el compilador:

```
ornitorrinco, orden_66, es42 = método(3)
print es42
print ornitorrinco
print orden_66
```

Las variables `ornitorrinco`, `orden_66` y `es42` se suponen enteras. “método” deberá solicitar al usuario 3 valores enteros y de alguna forma asignárselo a las variables enteras. El comando imprimir deberá mostrar en la salida estándar el valor ingresado por el usuario. Las variables donde se guardan los valores podrían tener cualquier nombre (por ejemplo, puede seguir el formato de los identificadores de su proyecto).

La impresión de valores de las variables puede realizarse en cualquier orden.

La sintaxis usada en el ejemplo anterior es para demostrar qué es lo que se espera. Tiene completa libertad para decidir cómo será su sintaxis (por ejemplo: usar punto y coma para delimitar instrucciones, asignar los valores de las variables de otra forma, usar otro método para imprimir los datos, etc).

Puede basarse en el proyecto del curso (modificando el proyecto e incorporando la nueva forma de hacer llamados) o bien, puede implementar desde cero su solución.

Ahora suponga el siguiente ejemplo:

```
pato, gat@ = método(4)
print pato
print gat@
print perro
```

Un error léxico en el ejemplo anterior sería el uso de @ en el nombre de identificadores (suponiendo que se definió que sólo caracteres alfanuméricos son válidos).

Errores sintácticos deberán ser mostrados en caso de que no cumpla con la sintaxis que usted definió. Por ejemplo, si en el ejemplo anterior usa [] en lugar de (), deberá lanzar un mensaje indicando que se esperaba un (y se encontró un [.

El código anterior debería indicar al menos dos errores semánticos:

- La cantidad de variables de retorno (4) no coincide con el número de variables a asignar (2)
- La variable perro no existe

Evaluación

Para calificar su solución tome en cuenta los siguientes rubros:

- 10% Análisis léxico
 - Funciona
 - Indica claramente cuál es el error léxico
 - Defensa
- 25% Análisis sintáctico
 - Funciona
 - Indica claramente cuál es el error sintáctico
 - Defensa
- 20% Estructura de datos
 - Defensa
- 20% Análisis semántico
 - Considera al menos lo siguiente: Un llamado al macro debe guardar los valores en algún lado obligatoriamente. La cantidad de variables que se guardan debe coincidir con el parámetro que se envía al macro. Además, las variables a imprimir deben tener un valor asignado, sino debería indicarse el error.
 - Indica claramente cuál es el error semántico
 - Defensa
- 25% Generación de código
 - La salida del programa es un archivo que puede ser ejecutado en un simulador de MIPS
 - Funcionalidad
 - Defensa

Entrega

Este examen es estrictamente individual.

La fecha máxima de entrega es el **jueves 5 de julio a la 2pm** en Mediación Virtual. Deberá subir un archivo comprimido con la solución a su examen, además de un ejecutable que reciba archivos con el código de entrada y que la salida sea el archivo con código MIPS correspondiente. Este archivo debe poderse ejecutar en SPIM sin errores. Incluya un documento con su nombre que muestre la estructura de datos que utilizó.

Deberá defender su solución el jueves 5 entre las 10am y las 12md. o entre la 1pm y las 6pm ante el profesor. Se atenderá FIFO. En caso de que alguien no se presente en ese lapso a defender su solución del examen, perderá los puntos correspondientes a la defensa.