

**EXAMEN DE PROYECTO DE AUTÓMATAS Y LENGUAJES**  
**GRADO EN INGENIERÍA INFORMÁTICA**  
**CONVOCATORIA DE ENERO 2012-2013**

Fecha: 18 de enero de 2013

Nombre y apellidos:

---

**SE RECOMIENDA LA LECTURA COMPLETA DEL ENUNCIADO  
ANTES DE COMENZAR A RESOLVER EL PROBLEMA PROPUESTO.**

Ampliar el compilador desarrollado durante el curso para que incorpore una nueva **sentencia simple de incremento**. Esta sentencia puede tener dos formas diferentes:

- En la primera forma, la sentencia contiene una **variable escalar de tipo entero** y una **expresión de tipo entero**. El resultado de ejecutar la sentencia es que se incrementa la variable con el valor indicado por la expresión.
- En la segunda forma, la sentencia contiene una **variable de tipo array de enteros** y una **expresión de tipo entero**. El resultado de ejecutar la sentencia es que se incrementa cada uno de los elementos del array con el valor indicado por la expresión.

A continuación se muestran unos ejemplos de la nueva sentencia.

**Ejemplo 1** (*ej1.alf*)

```
main
{
    int x;
    x = 2;
    x += 8;
    printf x;
}
```

Programa correcto que imprime por pantalla el valor 10.

**Ejemplo 2** (*ej2.alf*)

```
main
{
    int x,y;
    x = 2;
    y = 20;
    x += y;
    printf x;
}
```

Programa correcto que imprime por pantalla el valor 22.

**Ejemplo 3** (ej3.alf)

```
main
{
    int x;
    x = 2;
    x += (3*4+12) - 20;
    printf x;
}
```

Programa correcto que imprime por pantalla el valor 6.

**Ejemplo 4** (ej4.alf)

```
main
{
    int y;

    function int prueba()
    {
        int x;
        x = 100;
        x += 1;
        printf x;
        return 0;
    }

    y = prueba();
}
```

Programa correcto que imprime por pantalla el valor 101.

**Ejemplo 5** (ej5.alf)

```
main
{
    int y;

    function int prueba(int x)
    {
        x += x;
        printf x;
        return 0;
    }

    y = prueba(4000);
}
```

Programa correcto que imprime por pantalla el valor 8000.

**Ejemplo 6** (*ej6.alf*)

```
main
{
    int y;

    x += 90;

}
```

Programa semánticamente incorrecto porque la variable x no ha sido declarada.

**Ejemplo 7** (*ej7.alf*)

```
main
{
    int y;

    function int cero()
    {
        int x;
        x = 0;
        return x;
    }

    cero += 88;
}
```

Programa semánticamente incorrecto porque “cero” es un identificador de función.

**Ejemplo 8** (*ej8.alf*)

```
main
{
    boolean y;

    y += 90;

}
```

Programa semánticamente incorrecto porque la variable y no es de tipo entero.

**Ejemplo 9** (*ej9.alf*)

```
main
{
    boolean y;
    int x;

    x += y;

}
```

Programa semánticamente incorrecto porque la expresión de incremento no es de tipo entero.

**Ejemplo 10** (*ej10.alf*)

```
main
{
    array int[5] v5;

    v5[0] = 10;
    v5[1] = 20;
    v5[2] = 30;
    v5[3] = 40;
    v5[4] = 50;

    v5 += 8;

    printf v5[0];
    printf v5[1];
    printf v5[2];
    printf v5[3];
    printf v5[4];

}
```

Programa correcto que imprime por pantalla los valores 18 28 38 48 58.

**Ejemplo 11** (*ej11.alf*)

```
main
{
    array int[5] v5;

    v5[0] = 10;
    v5[1] = 20;
    v5[2] = 30;
    v5[3] = 40;
    v5[4] = 50;

    v5 += -10;

    printf v5[0];
    printf v5[1];
    printf v5[2];
    printf v5[3];
    printf v5[4];

}
```

Programa correcto que imprime por pantalla los valores 0 10 20 30 40.

Se pide, a partir de la descripción de la nueva sentencia y de los ejemplos expuestos:

1. Deducir la sintaxis de la nueva sentencia y a partir de ella, realizar los cambios oportunos en el analizador morfológico y sintáctico. **(3 puntos)**
2. Deducir las restricciones semánticas de la nueva sentencia y a partir de ellas realizar los cambios oportunos en el analizador semántico. **(3,5 puntos)**
3. Deducir el código ensamblador que corresponde a la nueva sentencia y realizar los cambios oportunos en el generador de código. **(3,5 puntos)**

### ENSAMBLADOR

Si necesita incrementar en 1 el valor de un registro, puede utilizar la instrucción:

**inc <reg>**

### NOTA ACERCA DE LA CALIFICACIÓN

La nota final del examen se obtiene haciendo la suma de las puntuaciones obtenidas en los tres apartados anteriores siempre que en cada uno de los apartados el alumno obtenga al menos un 40% de la puntuación total del apartado. Es decir, en el apartado 1, 1.2 puntos, y 1.4 puntos en los apartados 2 y 3. Si un alumno no cumple esta restricción, el examen no estaría aprobado y se le asignaría una calificación estándar de 3,5 puntos.

### NORMAS DE ENTREGA

Se entregará un único fichero comprimido (.zip) que contenga todos los ficheros del nuevo compilador. El nombre del fichero comprimido deberá ser :

**Continua\_Apellido1\_Apellido2\_Nombre.zip.**