

### 3.4 Interrupciones de bucles y programas

- **La sentencia break**

La sentencia `break` se puede utilizar únicamente en el interior de cualquiera de los bucles del lenguaje M. Permite interrumpir el bucle desde cualquier punto de su cuerpo. Cuando se ejecute el comando `break`, el bucle termina inmediatamente y el programa continúa en la sentencia que sigue a éste. Por ejemplo, este programa imprime los números del 1 al 100.

```
for i=1:100
```

```
    if (i==11),break,end
```

```
    disp(i);
```

```
end
```

Cuando `i` toma el valor 11, la condición asociada a `if` es cierta y se ejecuta `break`; en ese momento se interrumpe el bucle (no se imprime el valor 11 ni los siguientes). Si no se hubiera escrito la línea de comando `break`, se imprimirían los números del 1 al 100.

Si `break` pertenece a un bucle que es interior a otro, sólo produce la interrupción del bucle al que pertenece, siguiendo el programa en el bucle exterior.

- **La sentencia continue**

Esta sentencia también se utiliza únicamente dentro de cualquiera de los dos bucles del lenguaje M.

La sentencia `continue` envía el flujo del programa a la cabecera del bucle en el que se encuentre, dejándose de ejecutar, en esa iteración, las líneas de comando que se encuentren entre `continue` y el final del bucle. No se interrumpe un bucle sino una iteración.

En el siguiente ejemplo, se ha realizado un programa semejante al del apartado anterior empleando el comando `continue` en lugar de `break`.

```
for i=1:100
```

```
    if (i==11),continue,end
```

```
    disp(i);
```

```
end
```

El bucle comienza con el valor 1 de la variable de control, que se va incrementando en una unidad en cada iteración imprimiéndose en pantalla. Cuando *i* vale 11 se cumple la condición asociada a *if* y, por tanto, se ejecuta *continue*, que envía el flujo del programa a la cabecera del bucle sin ejecutarse la iteración actual (el *disp* correspondiente) pero sí las siguientes. Debido a esto, el valor 11 no aparece en pantalla. El siguiente valor que toma *i* es 12, que se imprimirá, y así sigue el proceso hasta el valor 100.

En el siguiente programa se introducen las notas de 10 alumnos, si alguna no es correcta (menor que 0 o mayor que 10) se volverá a pedir esa nota (*continue* evita ejecutar el incremento de *i*, con lo cual se mantiene en el mismo alumno). Además se calcula la nota media.

```
i=1;media=0;
while(i<=10)
    fprintf('Alumno %d\n', i);
    notas(i)=input('Introduce nota');
    if (notas(i)>10 | notas(i)<0) continue;end;
    media=media+notas(i);
    i=i+1;
end
fprintf('La nota media es %f\n',media/10)
```

#### – La sentencia *return*

Esta sentencia se puede utilizar en cualquier lugar de un programa o función M. Provoca la finalización anticipada de la función o programa.

Por ejemplo, el siguiente programa termina anticipadamente cuando el dato introducido se considera no válido.

```
notas=input('Introduce las tres notas del alumno\n');
if (max(notas)>10 || min(notas)<0 )
    disp('Datos erróneos');
    return;
end
media=sum(notas)/length(notas);
fprintf('La nota media es: %f\n', media)
```