### **Bucles**

### Tipos de bucle

- Número de repeticiones no conocido a priori:
  - while
  - do-while

- Número de repeticiones conocido a priori:
  - for

### Variable de control del bucle

- 1. Debe ser inicializada antes de comenzar el bucle
- Su valor debe ser evaluado/comprobado en cada repetición del bucle
- Su valor debe ser actualizado en cada repetición del bucle

#### **Bucles** infinitos

- Cómo se detienen:
  - Ctrl-C
  - Linux: Matando el proceso (kill -9)

- Cómo se descubren:
  - Poniendo trazas en varios puntos del programa: cout << "he llegado hasta aquí (4)" << endl;</p>
  - Utilizando el depurador

#### **Bucle while**

Sintaxis:

while(expression)
 statement

```
while (7)
    cout << "hello" << endl;
    cout << "good-bye" << endl;</pre>
```

```
short input;
cout << "enter number between 5 and 89, inclusive: ";
cin >> input;
while (input < 5 \mid \mid input > 89)
    cout << "that value is unacceptable... try again: ";
    cin >> input;
cout << "the value " << input</pre>
     << "is in the interval [5, 89]" << endl;
```

```
long sum = 0;
short counter = 1;
while (counter <= 100)
    sum += counter;
    counter++;
cout << "sum of first 100 integers: " << sum;
```

```
const string QUESTION = "Can you guess my secret number? ";
const short ANSWER = 1234;
short ans;
cout << QUESTION;
cin >> ans;
while ((ans - ANSWER) != 0)
    cout << "Incorrect number. Please try again." << endl;</pre>
    cout << QUESTION;
cout << "You found it! Congratulations!" << endl;</pre>
```

### Bucle do-while

Sintaxis:

```
do
    statement
while(expression);
```

```
const short MAX AGE = 116, MAX ALLOWABLE TRIES = 3;
short age, count = 0;
do
    cout << "Please enter your age: ";</pre>
    cin >> age;
    count++;
    if (age \leq 0 || age > MAX AGE)
        cout << "This is not a valid value!" << endl;
        if (count == MAX ALLOWABLE TRIES)
            exit(1);
    while (age \leq 0 || age > MAX AGE);
```

### **Bucle for**

Sintaxis:

```
for (expression1; expression2; expression3)
    statement
```

```
short sum;
cout << "What is 2+2? ";
cin >> sum;
for (; sum != 4;)
    cout << "Incorrect!! What is 2+2? " << endl;
    cin >> sum;
```

```
float average;
long max, sum = 0;
cout << "enter a positive integer: ";</pre>
cin >> max;
for (int i = 0; i \le max; i++)
    sum += i;
average = static cast<float>(sum) / max;
cout << "average is " << average << endl;</pre>
```

# Ámbito (scope) de variable de bucle

```
int i = 20;
for (int i = 1; i \le 10; i++)
    cout << "hello" << endl;
cout << i << endl;
```

### Ejercicio

 Escribir un programa en C que imprima todos los años no bisiestos entre 1900 y 2000, incluyendo ambos en su caso.

Queremos imprimir lo siguiente:

```
* * * * *

* * * *

* * *
```

Imprimimos 5 líneas:

```
for (int i = 0; i < 5; i++)
{
    cout << endl;
}</pre>
```

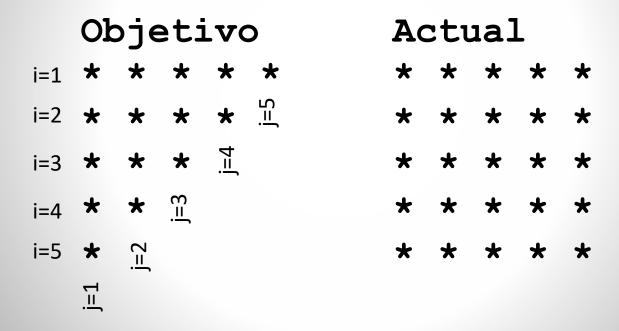
Imprimimos 5 asteriscos:

```
for (int j = 0; j < 5; j++)
{
    cout << "*";
}</pre>
```

Combinamos (anidamos):

```
for (int i = 0; i < 5; i++)
{
    for (int j = 0; j < 5; j++)
        {
        cout << "* ";
        }
        cout << endl;
}</pre>
```

Salida real vs. salida esperada:



#### Corregimos:

```
for (int i = 0; i < 5; i++)
{
    for (int j = 0; j < 5 - i; j++)
    {
        cout << "* ";
    }
    cout << endl;
}</pre>
```

**Ejercicio**: imprimir lo siguiente partiendo de lo anterior:

```
* * * * *

* * * *

* * *

* *
```