Alfredo Vellido : www.lsi.upc.edu/~avellido

# Fonaments d'Informàtica

## Semana 5. Estructuras iterativas

## RECAP: Estructura iterativa ‘while’

- La composición iterativa *while* tiene la siguiente **sintaxis**:

```
while (<expresion E>)
  {
    <bloque de instrucciones B>
  }
```

# Estructuras alternativas e iterativas

## Estructura iterativa 'while' (ej3b)

- **Ejemplo**: Dado un entero por teclado, **invertir el orden de sus cifras y sacar el número invertido por pantalla, pero sólo una vez invertido**

```
int n;
cin >> n;
int s=0; //Observar que int s=0 tiene que hacerse fuera del bucle

while(n!=0)
{
 s = 10*s + n%10; //en s los digitos de n, ya invertidos
 n=n/10;
}
cout << s << endl;
```

# Estructuras alternativas e iterativas

## La estructura iterativa ‘while’ se puede *anidar* ...

- Con la siguiente **sintaxis**:

```
while (<expresion E1>)
  {
   <bloque de instrucciones B1>
    while (<expresion E2>)
     {
      <bloque de instrucciones B2>
     }
    ...
  }
```

# Estructuras alternativas e iterativas

## Estructura iterativa 'while' (4b)

- Ej. de `while` anidado: tablas de multiplicar:

```
int main(void){
int num1 = 1, num2;
while (num1 <= 10)
{
 num2 = 1;
 while (num2 <= 10)
 {
  cout << num1 << "*" << num2 << "=" << num1*num2<<endl;
  num2++;
 }
 num1++;
 cout << endl;
}
system ("pause");return 0;
}
```

## Estructura iterativa ‘for’

- La composición iterativa *for* tiene la siguiente **sintaxis**:

```
for (<inicializ. I>; <expresion E>; <finaliz. M>)
{
  <bloque de instrucciones B>
}
```

**I** son las inicializaciones, **E** es la condición, **M** son las instrucciones a ejecutar al final del bucle, **B** es un bloque de instrucciones)

# Estructuras iterativas

## Estructura iterativa ‘for’

- La composición iterativa *for* tiene la siguiente **sintaxis**:

```
for (<inicializ. I>; <expresion E>; <finaliz. M>)
{
  <bloque de instrucciones B>
}
```

- **Efecto**:

```
I;
if (E) {B; M;}
if (E) {B; M;}
...
if (E) {B; M;}
(hasta que !E)
```

# Estructura iterativa ‘for’

- **Observaciones**:
  - El `for` no es estrictamente necesario.
  - La condición es una expresión booleana.
  - Las llaves no son necesarias si sólo hay una instrucción (y entonces se puede escribir en la misma línea:

    ```
    for (I; E; M) B;
    ```
  - Suele tener una variable asociada (se inicializa en **`I`** y se modifica en **`M`**)

## Estructura iterativa ‘for’ (2)

- **Ejemplo trivial: dado entero positivo por teclado, escribir en pantalla todos los enteros hasta el mismo**

```
int i=1,n;
cin >> n;
while (i<=n)
 {cout << i << " ";
  i++;
 }
```

## Estructura iterativa ‘for’ (2)

- **Ejemplo**: **Cálculo de factorial de un entero positivo (papel y lápiz)**

```
int n;
cin >> n;
int f = 1;
for (int i=2; i<=n; i++) f = f*i;
cout << f << endl;
                        // Observad el caso n=0, n=1
```

## Estructura iterativa ‘for’ (2b)

- **Ejemplo**: **Sacar por pantalla los divisores de un número entero positivo entrado por teclado**

```
int n;
cin >> n;   // n estríctamente positivo
for (int i=2; i<n; i++)
{
  if (n%i == 0) cout << i << endl;
}
```

## Estructura iterativa ‘for’ (2c)

- **Ejemplo**: **Calcular y sacar por pantalla la potencia de un número entero cualquiera, entrando por pantalla base y exponente (prohibido <cmath>)**

```
int b,e;
cin >> b >> e; // b estrictamente positivo, e positivo
int pot = 1;

for (int i=0; i<e; i++) pot = pot*b;
cout << pot << endl;
```

## Estructura iterativa ‘for’ (3)

- La composición iterativa **for** también se puede anidar. Ej: tablas de multiplicar:

```
int main(void){
for (int num1 = 1; num1<= 10; num1++)
 {
  for (int num2 = 1; num2<= 10; num2++)
   {
    cout << num1 << " * " << num2 << " = " <<
       num1*num2 << endl;
   }
 }
system("pause“); return 0;
}
```