Programación I

Instrucciones de control - bucles

Iván Cantador

Escuela Politécnica Superior Universidad Autónoma de Madrid

Contenidos

Escuela Politécnica Superior

- Bucles
 - La instrucción for
 - La instrucción while
 - La instrucción do while
- Bucles anidados



- Bucles
 - La instrucción for
 - La instrucción while
 - La instrucción do while
- Bucles anidados



Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid



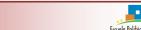
Bucles: la instrucción for (I)

• Un bucle for ejecuta un bloque de sentencias de forma iterativa mientras se cumpla una condición de fin, a partir de una sentencia de inicio y considerando una sentencia de paso de iteración

Sintaxis

```
for ( <inicio>; <condicion_fin>; <paso_iteracion> ) {
   <blow><br/>bloque_sentencias></br>
```

→ Las sentencias inicio, paso_iteracion y condicion_fin son optativas







• En general, el número de iteraciones de un bucle **for** viene determinado por valores inicial, de incremento y final de una variable entera (contador), que por motivos históricos suele nombrarse i, j, k, ...



Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid





• Ejemplo

#include <stdio.h>

suma += i;

void main() {
 int i, suma;

Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid

// Calculo de la suma de los numeros impares menores o iguales que 99



Bucles: la instrucción for (IV)

Ejemplo

```
#include <stdio.h>
#include <limits.h>

void main() {
   int lista[] = {4, 3, 7, 2, 9, 5};
   int longitudLista = 6;
   int i, valorMedio, valorMinimo, valorMaximo;

   // Calculo de los valores medio, minimo, y maximo de la lista
```

_

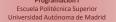
Bucles: la instrucción for (IV)

for (i = 1; $i \le 99$; i += 2) {

• Ejemplo

```
#include <stdio.h>
#include <limits.h>
void main() {
   int lista[] = {4, 3, 7, 2, 9, 5};
   int longitudLista = 6;
   int i, valorMedio, valorMinimo, valorMaximo;
   // Calculo de los valores medio, minimo, y maximo de la lista
   valorMedio = 0:
   valorMinimo = INT_MAX;
   valorMaximo = INT_MIN;
   for ( i = 0; i < longitudLista; i++ ) {</pre>
       valorMedio += lista[i];
       if ( lista[i] < valorMinimo ) {</pre>
          valorMinimo = lista[i];
       if ( lista[i] > valorMaximo ) {
          valorMaximo = lista[i];
   valorMedio /= longitudLista;
```







• Bucle for "infinito"

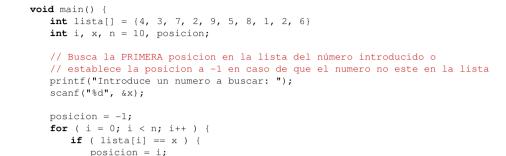
```
for (;;) {
      printf("Bucle infinito!\n");
```



Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid



10



• La sentencia break "rompe" la ejecución del bucle en el



Programación | Escuela Politécnica Superior Universidad Autónoma de Madrid

printf("La posicion de %d en la lista es %d.\n", x, posicion);



11

Bucles: la instrucción for (VII)

• La sentencia break "rompe" la ejecución del bucle en el que se encuentra

```
#include <stdio.h>
void main() {
   int lista[] = {4, 3, 7, 2, 9, 5, 8, 1, 2, 6}
   int i, x, n = 10, posicion;
   // Busca la ULTIMA posicion en la lista del número introducido o
   // establece la posicion a -1 en caso de que el numero no este en la lista
   printf("Introduce un numero a buscar: ");
   scanf("%d", &x);
   posicion = -1;
   for ( i = n-1; i >= 0; i-- ) {
      if ( lista[i] == x ) {
          posicion = i;
          break;
   printf("La posicion de %d en la lista es %d.\n", x, posicion);
```

Escuela Politécnica Superior

Universidad Autónoma de Madrid





Bucles: la instrucción for (VIII)

Bucles: la instrucción for (VI)

que se encuentra

break:

#include <stdio.h>

```
#include <stdio.h>
void main()
   int lista[] = {4, 3, 7, 2, 9, 5, 8, 1, 2, 6}
   int i, j, x, n = 10, posiciones[10];
   // Busca TODAS las posiciones en la lista del número introducido o
   // estableciendolas a -1 en caso de que el numero no este en la lista
   printf("Introduce un numero a buscar: ");
   scanf("%d", &x);
   posiciones[0] = -1;
   i = 0:
   for (i = 0; i < 10; i++) {
      if ( lista[i] == x ) {
          posiciones[j] = i;
          j++;
   // Imprimimos por pantalla todas las posiciones encontradas
   for (i = 0; i < j; i++) {
      printf("La posicion de %d en la lista es %d.\n", x, posiciones[i]);
```



12

• La sentencia continue "continua" (salta) a la siguiente iteración de un bucle sin ejecutar en la iteración actual aquellas sentencias posteriores al continue

```
#include <stdio.h>
void main() {
   int lista[] = {4, 3, 7, 2, 9, 5, 8, 1, 2, 6}
   int i, x, n = 10;
   // Imprime por pantalla los numeros de una lista que son mayores a uno dado
   printf("Introduce un numero a buscar: ");
   scanf("%d", &x);
   for (i = 0; i < n; i++) {
      // Si el numero i-esimo de la lista es menor o iqual que x se sique
      // la ejecucion del bucle con el siguiente numero
      if ( lista[i] <= x ) {
          continue;
      printf("Numero mayor que %d en la posicion %d: %d.\n", x, i, lista[i]);
```



Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid



14

Bucles: la instrucción while (I)

• Un bucle while ejecuta un bloque de sentencias de forma iterativa mientras se cumpla una condición de fin

Sintaxis

```
while ( <condicion_fin> ) {
   <blow><br/>bloque_sentencias></br>
```

→ Las sentencia condicion fin es <u>obligatoria</u>

Contenidos

Bucles

- La instrucción for
- · La instrucción while
- La instrucción do while
- Bucles anidados



Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid



15

Bucles: la instrucción while (II)

Eiemplo

```
#include <stdio.h>
void main() {
  char opcion = 'X';
  while ( opcion != 'A' && opcion != 'B' ) {
     printf("Elija una opcion, A o B: ");
     scanf("%c", &opcion);
  printf("Has elegido la opcion %c!\n", opcion);
```



printf("Bucle infinito!\n");

Bucle while "infinito"

while (1) {

 Al igual que en los bucles for, la instrucción break rompe la ejecución del bucle while en el que se encuentra

```
#include <stdio.h>

void main() {
    char opcion;

while ( 1 ) {
        printf("Elija una opcion, A o B: ");
        scanf("%c", &opcion);

        if ( opcion == 'A' || opcion == 'B' ) {
            break;
        }
    }

    printf("Has elegido la opcion %c!\n", opcion);
}
```



Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid



19

Bucles: la instrucción while (V)

Equivalencia entre bucles for y bucles while

18

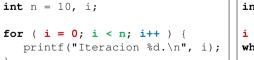
Bucles: la instrucción while (VI)

• Al igual que en los bucles for, la instrucción **continue** salta a la siguiente iteración del bucle while en el que se encuentra

```
#include <stdio.h>
#define AS
#define DOS
// El resto de macros irian aqui
#define SIETE
#define SOTA
#define CABALLO 0.5
#define REY
   double puntos=0, cartas[40] = {SOTA, AS, CUATRO, DOS, REY, /* resto de cartas */ DOS};
   int opcion, c=0;
   while ( puntos < 7.5 ) {
       printf("Quieres carta (s/n)? ");
        scanf("%c", &opcion);
        if ( opcion == 's' || opcion == 'S' ) {
            puntos += cartas[c];
            C++:
            continue;
       printf("Te has plantado!\n");
   printf("Tienes %f puntos!\n", puntos);
```

Escuela Politécnica Superior

Universidad Autónoma de Madrid



```
int n = 10, i;
i = 0;
while ( i < n ) {
   printf("Iteracion %d.\n", i);
   i++;
}</pre>
```





UNIVERSIDAD AUTONOMA

DE MADRID



Programación I

Universidad Autónoma de Madrid

Bucles

Contenidos

- La instrucción for
- La instrucción while
- · La instrucción do while
- Bucles anidados



Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid

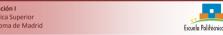


• Un bucle do while ejecuta un bloque de sentencias una vez y luego lo ejecuta de forma iterativa mientras se cumpla una condición de fin

Sintaxis

```
do {
   <blow>dlogue sentencias>
while ( <condicion_fin> );
```

→ Las sentencia condicion fin es obligatoria



UNIVERSIDAD AUTONOMA
DE MADRID

Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid



Bucles: la instrucción do while (II)

22

Bucles: la instrucción do while (III)

23

• Ejemplo:

```
#include <stdio.h>
void main() {
  char opcion;
     printf("Elija una opcion, A o B: ");
     scanf("%c", &opcion);
  while (opcion != 'A' && opcion != 'B' );
  printf("Has elegido la opcion %c!\n", opcion);
```

• Al igual que en los bucles for y while, en los bucles do while también se puede hacer uso de las instrucciones break y continue

Escuela Politécnica Superior

Universidad Autónoma de Madrid







25

La instrucción for

Bucles anidados

• La instrucción while

• La instrucción do while

Bucles

Bucles anidados (I)

- Un bucle anidado es un bucle que está dentro de otro
 - La anidación de bucles puede ser imprescindible para la realización de cálculos complejos
 - Ejemplo:

```
double A[2][3] = {{1, 2, 3}, {4, 5, 6}};
double media;
int i, j;

// 2 bucles anidados para calcular la media de los numeros
// de una matriz
media = 0;
for ( i=0; i<2; i++ ) {
   for ( j=0; j<3; j++ ) {
      media += A[i][j];
    }
}
media /= 2*3;</pre>
```



Programación I Escuela Politécnica Superior Universidad Autónoma de Madrid



27

Bucles anidados (II)

I)

• Un bucle anidado es un bucle que está dentro de otro

Programación I

Escuela Politécnica Superior

Universidad Autónoma de Madrid

- Puede existir más de un nivel de anidación
- Ejemplo:

```
double A[2][3] = {{1, 2, 3}, {4, 5, 6}};
double B[3][1] = {{1}, {2}, {3}};
double C[2][1];
int i, j;

// 3 bucles anidados para multiplicar las matrices A y B
for ( i=0; i<2; i++ ) {
   for ( j=0; j<1; j++ ) {
      C[i][j] = 0.0;
      for ( k=0; k<3; k++ ) {
        C[i][j] += A[i][k] * B[k][j];
      }
   }
}</pre>
```

26

Bucles anidados (III)

• En bucles anidados las instrucciones **break** y **continue** se aplican al <u>bucle más interno</u> al que pertenecen

```
double x, A[2][3] = \{\{1, 2, 3\}, \{4, 5, 6\}\};
int i, j, fila, columna;
// Busca la posicion de una matriz en la que se encuentra un numero dado
printf("Introduce el numero a buscar: ");
scanf("%f", &x);
fila = columna = -1;
for ( i=0; i<2; i++ ) {</pre>
   for ( j=0; j<3; j++ ) {</pre>
       if ( A[i][j] == x ) {
           fila = i;
           columna = j;
                             // Aquí se rompe el bucle j
           break;
   if (fila != -1) {
       break;
                             // Aquí se rompe el bucle i
printf("El numero %d esta en la posicion (%d, %d).\n", x, fila, columna);
```



