


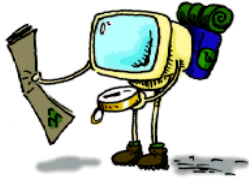
UAB
Universitat Autònoma de Barcelona

Tema 4: Estructures de control




EE
Enginyeria Electrònica

- ◆ **4.1 Estructura general d'un programa: estructures de control**
- ◆ **4.2 Estructures seqüència**
- ◆ **4.3 Estructures de selecció**
 - La bifurcació simple: `if`.
 - Bifurcació doble: `if ... else`.
 - Bifurcació doble: l'operador condicional `?:`
 - Bifurcació múltiple: `if ... else if ... else`.
 - Selecció: `switch`
- ◆ **4.4 Estructures d'iteració.**
 - Bucle condicional o sentinella: `while`.
 - Bucle condicional o sentinella: `do - while`.
 - Repetir n cops: `for`.
- ◆ **4.5 Les instruccions `break` i `continue`**
- ◆ **4.6 Més sobre l'operador increment/decrement**
- ◆ **4.7 Exemples**




1



UAB
Universitat Autònoma de Barcelona

4.1 Estructura general d'un programa: estructures de control




EE
Enginyeria Electrònica

- **Programa** →
 - És un conjunt d'instruccions (ordres donades a la màquina) que produeixen l'execució d'una determinada tasca
 - Davant d'entrades determinades produeix unes sortides específiques → Accions de *llegir* i *escriure*
 - L'algorisme establirà el conjunt d'accions (instruccions) que han de resoldre el problema
- **Acció primitiva** →
 - Quan no es pot descompondre en accions més simples. Exemple: assignar, llegir, escriure, ...
- **Bloc** →
 - Conjunt d'accions que s'executen consecutivament sense interrupcions
- El 1966, Böhm i Jacopini demostren que qualsevol programa (algorisme) es pot expressar amb només tres estructures de control que estableixen el flux de l'algorisme: **seqüència**, **selecció** i **iteració**.


Microelectrònica i Sistemes Electrònics

2



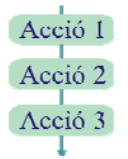
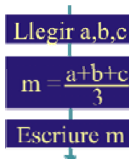
UAB
Universitat Autònoma de Barcelona

4.2 Estructures seqüència




EE
Enginyeria Elèctrica

- Cada bloc s'executa correlativament. es compleix que:
 - Cada cop s'executa només una instrucció (acció)
 - Cada acció s'executa només un cop
 - L'ordre d'execució és el mateix en què estan escrites
 - La finalització del darrer pas és la finalització de l'algorisme
- La representació de l'estructura seqüencial (i de les estructures selecció i iteració, com veurem) es pot fer en pseudocodi, diagrama de flux i, després, es transcriu al C emprant les instruccions del llenguatge.
- Format:


Pseudocodi	Diagrama de flux	Exemple: calcular la mitjana de 3 nombres		
Inici Acció 1 Acció 2 ... Acció n Fi		Inici Llegir a Llegir b Llegir c $m \leftarrow (a+b+c)/3$ Escriure (m) Fi		<pre>void main() { float a, b, c, m; scanf("%f%f%f", &a, &b, &c); m=(a+b+c)/3; printf("Mitjana m=%5.2f.\n", m); }</pre>

Microelectrónica i Sistemes Electrònics
3



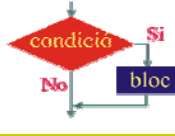

UAB
Universitat Autònoma de Barcelona

4.3 Estructures de selecció (I)




EE
Enginyeria Elèctrica

- Permeten variar la seqüència d'execució d'un algorisme en funció del resultat d'una expressió lògica o bé d'una altra expressió que pot prendre més de dos valors.
- La bifurcació pot ser **simple** (oberta), **doble** (tancada) o **múltiple** (tancada múltiple).
- La bifurcació simple: *if*


Pseudocodi:	Diagrama de flux:	Transcripció a C:
Si condició Llavors bloc FiSi		i) if (expressió) sentència ii) if (expressió) { bloc de sentències }
● Exemple: controlar si un número pot ser la referència d'un mes (format: dd/mm/aa)		
Llegir mes Si (mes > 12) Llavors Escriure ("Núm no vàlid") FiSi		<pre>void main () { int mes; printf("Entra numero de mes: "); scanf("%u", &mes); if (mes>12) printf("Mes no valid\n"); }</pre>

Microelectrónica i Sistemes Electrònics
4



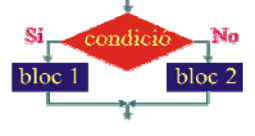
UAB
Universitat Autònoma de Barcelona

4.3 Estructures de selecció (II)

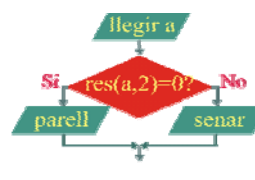


EE
Enginyeria Elèctrica i Electrònica

La bifurcació doble: **if ... else**


<p>Pseudocodi</p> <p>Si (condició) Llavors bloc1</p> <p>SiNo Llavors bloc2</p> <p>FiSi</p>	<p>Diagrama de flux</p> 	<p>Transcripció a C</p> <pre>if (condició) bloc1 else bloc2</pre> <p>Amb l'operador ?: Condició ? Expressió1 : Expressió2</p>
---	--	---

- Exemple: controlar si un número pot ser la referència d'un mes (format: dd/mm/aa)

<p>Pseudocodi:</p> <p>Entrar a;</p> <p>Si (residu(a,2) = 0) Llavors Escriure ("Parell.")</p> <p>SiNo Llavors Escriure ("Senar.")</p> <p>FiSi</p>	<p>Diagrama de flux:</p> 	<p>Transcripció a C:</p> <pre>void main() { int a, res; printf("Entra numero: "); scanf("%d", &a); res = a%2; if (res==0) printf("%i es parell.\n", a); else printf("%i es senar.\n", a); }</pre>
---	---	--


Microelectrónica i Sistemes Electrònics

5



UAB
Universitat Autònoma de Barcelona

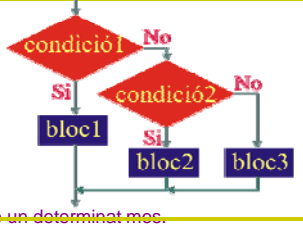
4.3 Estructures de selecció (III)



EE
Enginyeria Elèctrica i Electrònica

La bifurcació múltiple: **if ... else if ... else**

- La selecció múltiple es basa en una variable que pot prendre diferents valors.
- La instrucció de la selecció múltiple permet triar el camí correcte des de la primera comparació


<p>Pseudocodi</p> <p>Si condició1 Llavors bloc1</p> <p>SiNo Si condició2 Llavors bloc2</p> <p>...</p> <p>FiSi</p>	<p>Diagrama de flux</p> 	<p>Transcripció a C</p> <pre>if (condició 1) { bloc 1 } else if (condició 2) { bloc 2 } else { bloc 3 }</pre>
--	--	--

- Exemple: Dir quants dies té un determinat mes.

```
...
int any_de_traspas, mes, dies;
...
if (mes==4 || mes==6 || mes==9 || mes==11) dies = 30;
else if (mes==2)
  if (any_de_traspas) dies = 29;
  else dies = 28;
else dies = 31;
```


Microelectrónica i Sistemes Electrònics

6



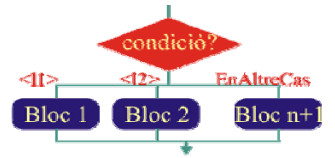
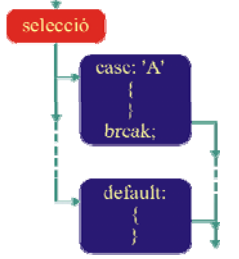
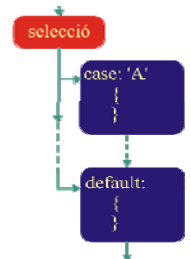
UAB
Universitat Autònoma de Barcelona

4.3 Estructures de selecció (IV)




EE
Enginyeria Elèctrica

- La selecció: **switch ... case**
 - La selecció múltiple es basa en una variable que pot prendre diferents valors.


<p>EnCasDe expressió</p> <pre> l1: bloc 1 Fi l1 ... ln: bloc n Fi ln EnAltreCas: bloc n+1 FiCas </pre>		<p>Transcripció a C:</p> <p>En C es transcriu amb la instrucció switch - case. L'expressió pot ser int o char, però no float.</p>
<pre> switch (condició) { case Expr.Const.1 bloc 1; break; //segons cas ... default: bloc n+1 } → dues possibilitats </pre>		

Microelectrónica i Sistemes Electrònics 7



UAB
Universitat Autònoma de Barcelona

4.3 Estructures de selecció (V)



EE
Enginyeria Elèctrica


- Exemple: Analitzar les (4) possibilitats del següent programa.

```

void main()
{
  char tria;
  printf("Aquest programa calcula l'energia d'una font (on hi ha 3 resistencies) coneguent:\n");
  printf(" A) El valor de cada resistencia.\n");
  printf(" B) El valor resistiu total.\n");
  printf(" C) El valor del corrent.\n");
  printf("      → Què tries? ");
  scanf("%c", &tria);
  switch (tria)
  {
    case 'A': printf("Rt = R1+R2+R3 \n");
    case 'B': printf("It = Vt / Rt \n");
    case 'C': printf("Pt = Vt * It \n");
              break;
    default: printf("      → Tria incorrecta!\n");
  }
}
                    
```


Tria A	Tria B	Tria C	Altre tria
Rt = R1+R2+R3 It = Vt / Rt Pt = Vt * It	It = Vt / Rt Pt = Vt * It	Pt = Vt * It	Tria incorrecta!

Microelectrónica i Sistemes Electrònics 8



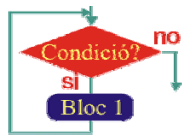
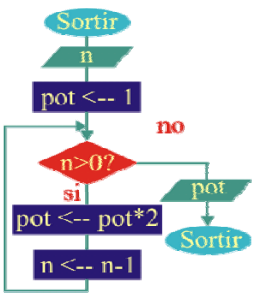
UAB
Universitat Autònoma de Barcelona

4.4 Estructures d'iteració (I)




EE
Enginyeria Elèctrica i Electrònica

- Permeten repetir un bloc de codi un nombre (determinat o indeterminat) de cops.
- **Bucle condicional o sentinella: Repetir mentre es compleixi: while**
 - Executa un conjunt d'instruccions mentre es compleixi una condició
 - Pot ser que no s'executi cap cop


<p>Pseudocodi Mentre condició Fer bloc</p> <p>Diagrama de flux</p>  <p>Característiques</p> <ul style="list-style-type: none"> - Número d'iteracions variable. - Finalitza quan deixa de complir-se la condició. - La condició s'ha de variar dins el bloc. <p>Codificació en C while (condició) bloc;</p>	<p>Exemple: Donar les n primeres potències de 2.</p> <pre>void main() { int pot, n; printf("→ A quina potència? "); scanf("%u", &n); pot = 1; printf(" → 2^%i = ", n); while (n > 0) { pot = pot * 2; n--; } printf("%i \n", pot); }</pre>	
--	---	---

Microelectrònica i Sistemes Electrònics 9




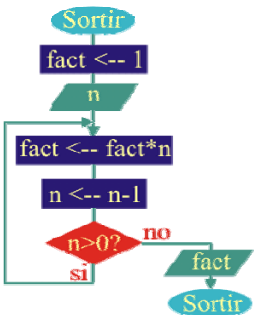
UAB
Universitat Autònoma de Barcelona

4.4 Estructures d'iteració (II)




EE
Enginyeria Elèctrica i Electrònica

- **Bucle condicional o sentinella: Repetir mentre es compleixi: do ... while**
 - Executa un conjunt d'instruccions mentre es compleixi una condició


<p>Pseudocodi Fer bloc Mentre condició</p> <p>Diagrama de flux</p>  <p>Característiques</p> <ul style="list-style-type: none"> - Una o més iteracions. - Nombre variable d'iteracions. - Finalitza quan deixa de complir-se la condició - La condició varia dins el bloc. <p>Codificació en C do bloc while (condició);</p>	<p>Exemple: Càlcul de n!.</p> <pre>void main() { int n, fact = 1; printf(" Entra n "); scanf("%u", &n); pot = 1; printf(" Factorial de %u = ", n); do { fact = fact*n; n--; } while (n > 0); printf("%u \n", fact); }</pre>	
--	--	---

Microelectrònica i Sistemes Electrònics 10



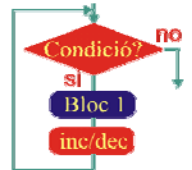
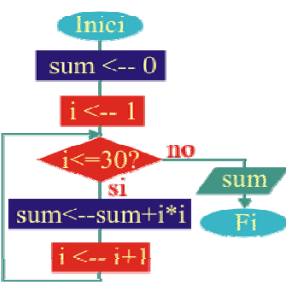
UAB
Universitat Autònoma de Barcelona

4.4 Estructures d'iteració (III)




EE
Enginyeria Elèctrica i Electrònica

- Bucle condicional o sentinella: Repetir n cops: **for**
 - Fa un número concret d'iteracions
 - La condició no es modifica dins el bloc

<p>Pseudocodi</p> <p>Per index // (inici, fi) Fer bloc inc/dec index</p> <p>Diagrama de flux</p>  <p>Codificació en C</p> <pre>for (inici; condició; pas) bloc</pre>	<p>Exemple:</p> <p>Escriure els quadrats dels 30 primers nombres.</p> <pre>void main() { int i, suma = 0; for (i=1; i<=30; i++) printf("%u^2 = %u\n", i, i*i); }</pre>	
--	--	--


Microelectrónica i Sistemes Electrònics

11



UAB
Universitat Autònoma de Barcelona

4.5 Instruccions **break** i **continue**



EE
Enginyeria Elèctrica i Electrònica


- **break** i **continue** són dues instruccions que ens permeten fer un salt en el bucle que s'està executant
- **break**, de forma similar a com ja s'ha vist en el case, obliga a una terminació immediata de les instruccions while i do-while
- Quan es troba **continue** en un bucle obliga a parar l'avaluació del cycle actual del bucle i a saltar al principi del bucle avaluant la propera condició que ve. S'aplica a les instruccions while, do-while i for. Se sol utilitzar per eviat l'avaluació de dades que no interessin. Per exemple, posant notes, si hi ha alguna nota negativa es pot saltar la seva avaluació fent servir continue en aquest cas particular.

4.6 Més sobre l'operador increment/decrement


- X++(--) → Incrementa/decrementa X després de fer l'operació (post-inc./dec.)
- ++(--X) → Incrementa/decrementa X abans de fer l'operació (pre-inc./dec.)
- Exemples:
 - Siguin i = 3, c = 10.
 - X = i++ → X = 3++ → Increment després de fer l'assignació: X = 3.
 - X = --c + 2 → X = --10 + 2 = 9 + 2 = 11
 - X = i-- + ++c → X = 3-- + ++10 = 3 + 11 = 14

Microelectrónica i Sistemes Electrònics

12

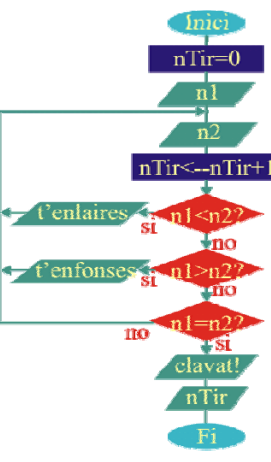


4.6 Exemple (I)




■ Exemple 1: joc d'endevinar el número tirat


- Un jugador diu un número i l'altre l'ha d'endevinar. Al final es diu el nombre de tirades fetes.

Pseudocodi	Diagrama de flux	Programa en C.
<pre> Enter n1, n2, nTir Inici Escriure ("Dir numero") nTir=0 Llegir (n1) Fer Escriure("Fes tirada "); Llegir (n2) nTir=nTir+1 Si (n1<n2) Llavors Escriure("t'enlaires") SiNo Si (n1>n2) Llavors Escriure("t'enfonses") Mentre (n1<>n2) Escriure("clavat!") Escriure("Tirades=", nTir) Fi </pre>		<pre> /* Joc endevinar numero*/ void main() { int n1, n2, nTir=0; printf("Entra numero: "); scanf("%i", &n1); do { printf("Tira: "); scanf("%i",&n2); nTir++; if (n1<n2) printf("t'enlaires!\n"); else if (n1>n2) printf("t'enfonses!\n"); }while (n1!=n2); printf(" ...clavat!\n"); printf("Tirades = %i\n",nTir); } </pre>

Microelectrónica i Sistemes Electrònics 13

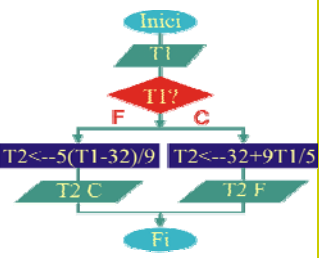


4.6 Exemple (II)



■ Realitzar un programa de conversió de graus Celsius a Fahrenheit i a l'inrevés.

- L'entrada vindrà donada en format 30.5C o 40F, per exemple.

Pseudocodi	Diagrama de flux	Programa en C.
<pre> Caracter car Real T1, T2 Inici Escriure ("Temperatura?") Llegir (T1, car) EnCasDe (car) 'F':T2= 5*(T1-32)/9 Escriure("T2C") Fi F 'C':T1= 32+9*T1/5 Escriure("T2F") Fi C FiEnCasDe Fi </pre>		<pre> /* Conversió Celsius-Fahrenheit*/ void main() { char car; float T1, T2; printf("Dona la temperatura: "); scanf("%f%c", &T1, &car); switch (car) { case 'F':T2=5.0*(T1-32)/9.0; printf("%2.1fC\n",T2); break; case 'C': T2 = 32+9.0*T1/5.0; printf("%2.1fF\n",T2); } } </pre>

Microelectrónica i Sistemes Electrònics 14

UFAB		4.6 Exemple (III)	
<p>Problema:</p> <p>Es té un conjunt desconegut de resistències de valors 15 i 47 ohms. Se sap que el seu valor total quant estan connectades en sèrie val <i>tot</i>. Realitzar el programa que troba totes les combinacions de resistències R1 i R2 possibles.</p>			
	<p>Pseudocodi</p> <pre> Enter Const R1=15, R2=47 Enter tot, num=0, i, j, nR1, nR2 Inici Escriure ("tot?") Llegir (tot) nR1=tot/R1 nR2=tot/R2 Per (i=0; i<=nR1) Fer Per (j=0; j<=nR2) Fer Si ((i*R1+j*R2) == tot) Escriure(i*R1, j*R2) num=num+1 FISI j<-j+1; FIPer i<-i+1; FIPer Escriure("Combin.=",num) Fi </pre>	<p>Programa en C</p> <pre> void main() { const int R1=15; const int R2=47; int tot, num=0, i, j, nR1, nR2; printf("Que val 'tot'? "); scanf("%i", &tot); nR1 = tot/R1; nR2 = tot/R2; for (i=0; i<=nR1; i++) for (j=0; j<=nR2; j++) if ((i*R1 + j*R2) == tot) { printf("%d'R1,%d'R2\n", i, j); num++; } printf("#combinacions=%d\n", num); } </pre>	
Microelectrònica i Sistemes Electrònics		15	