

ALGORISMES I PROGRAMACIÓ (GESTIÓ AERONÀUTICA)

SESSIÓ 28/02/2005

1. 'if – else' i 'if – else – if'

En el transcurs d'un programa normalment cal prendre decisions. La manera de fer-ho és avaluant alguna expressió de tipus lògic i, en funció del resultat, fer una cosa o altra. Això es fa amb la sentència 'if – else', que funciona de la següent manera:

```
if (condició) {  
    sentències  
}  
else {  
    sentències  
}
```

} Bloc if
} Bloc else

on 'condició' és alguna operació lògica, 'sentències' és una o varies sentències i la part de l'else és opcional. Si la condició és certa (val 1 o TRUE), s'executa el bloc pertanyent a l'if i sinó (val 0 o FALSE), el de l'else (si existeix)

Si en un dels dos blocs hi ha només una sentència, no cal posar {}, tot i que és recomanable per evitar possibles ambigüitats (en if – else aniuats).

Exemple>

```
#include "stdio.h"  
  
void main(void)  
{  
    int num;  
  
    printf("\nEntra un nombre del 1 al 10: ");  
    scanf("%d",&num);  
  
    if ( (num%2) == 0 ) //Resta de num entre 2. Només pot ser 0 o 1.  
        printf("El nombre que acabes d'entrar és parell\n");  
    else  
        printf("El nombre que acabes d'entrar és senar\n");  
}
```

Pot donar-se el cas en què necessitem actuar de manera diferent per a més de dos casos. Ara if – else se'ns queda curt. Necessitem quelcom més. L'escala if – else – if pretén modelar-ho:

```
if (condició1) {  
    sentències_ condició1  
}  
else if (condició2){  
    sentències_ condició2  
}  
.  
.  
.  
else if (condicióN){  
    sentències_ condicióN  
}  
else {  
    sentències_si_cap_condició_anterior_és_certa  
}
```

Tan aviat com es trobi una condició certa, s'executaran les sentències associades a ella. Si cap de les condicions és certa, s'executaran les de l'else final.

Exemple>

```
#include "stdio.h"

void main(void)
{
    int num;

    printf("\nEntra un numero entre 1 i 100: ");
    scanf("%d",&num);

    if ( ( num >= 1 ) && ( num < 25 ) )
        printf("\nEl nombre està entre 1 i 24\n");
    else if ( ( num >= 25 ) && ( num < 50 ) )
        printf("\nEl nombre està entre 25 i 49\n");
    else if ( ( num >= 50 ) && ( num < 75 ) )
        printf("\nEl nombre està entre 50 i 74\n");
    else
        printf("\nEl nombre està entre 75 i 100\n");
}
```

Exercici #1> Col·loqueu un breakpoint a la línia de l'if, debugueu i observeu com salteu cap a l'opció que correspon.

2. 'switch – case'

El llenguatge C incorpora també una sentència de selecció múltiple, 'switch-case', que compara successivament el valor d'una expressió amb una llista de constants enteres o de caràcters. Quan troba una correspondència, executa les sentències associades a la constant (entre case i break). Si no troba cap correspondència executa les sentències per defecte (default). La forma general del switch és:

```
switch (expressió){
    case constant_1:
        sentències
        break;
    case constant_1:
        sentències
        break;
    .
    .
    .
    default:
        sentències
} // Fi switch
```

Exemple>

```
#include "stdio.h"

void main(void)
{
    int num;

    printf("\nEntra un numero entre 1 i 7: ");
```

```

scanf("%d",&num);

switch (num){
    case 1 :
        printf("\nDillins\n");
        break;
    case 2 :
        printf("\nDimarts\n");
        break;

    case 3 :
        printf("\nDimecres\n");
        break;

    case 4 :
        printf("\nDijous\n");
        break;

    case 5 :
        printf("\nDivendres\n");
        break;

    case 6 :
        printf("\nDissabte\n");
        break;

    case 7 :
        printf("\nDiumenge\n");
        break;

    default :
        printf("\nHe dit un nombre entre 1 i 7!!!!\n");
} // fi switch
} // fi programa

```

Exercici #2> Copieu el codi anterior, col·loqueu un breakpoint a la línia del `switch`, debugueu i observeu com salteu cap a l'opció que correspon.

3. 'for'

En moltes ocasions necessitarem repetir una sèrie d'instruccions un número determinat de vegades. La sentència 'for' del llenguatge C es l'estructura de control específica per aquest tipus de bucles. La sintaxi del 'for' és:

```

for (inicialització ; condició ; actualització)
{
    sentències;
} //fi del bucle

```

Abans d'iniciar-se el bucle s'executa **inicialització**, que és una sentència que assigna un valor inicial a la variable que fa de comptador. A continuació s'avalua la **condició**. Si la condició no es compleix (és FALSE), el 'for' s'acaba. Si la condició es compleix (és TRUE) s'executen les **sentències** entre les claus i es fa l'**actualització** de la variable comptador. Per fer aquesta actualització, normalment s'utilitzen els operadors d'increment (ex.- `n++`) i decrement (ex.- `n--`), però també es pot actualitzar el comptador amb expressions del tipus '`n=n+2`'. Aquests passos es van repetint fins que la condició es deixa de complir.

Exemple>

```
#include "stdio.h"

void main(void)
{
    int n;

    for(n=0;n<3;n++)
    {
        printf("HOLA \n");
        printf("ADEU \n\n");
    } //fi del bucle
}
```

En aquest exemple, el 'for' s'executa 3 vegades (n=0, n=1, n=2) i el resultat per pantalla serà:

```
HOLA
ADEU
```

```
HOLA
ADEU
```

```
HOLA
ADEU
```

Exercici #3> Modifiqueu l'exemple anterior per a que el bucle s'executi 5 vegades. A més, afegiu el codi necessari per a que abans d'escriure "HOLA" surti per pantalla el valor del comptador (és el valor de la variable n).

4. Altres exercicis

Exercici #4> Escriviu un programa que realitzi (sense perdre decimals) i mostri per pantalla les següents operacions:

- a. $Y = 1 / ((x-1) (x+1))$
- b. $Y = x / ((x+1) (x+2))$

on x ha de ser un enter que es demana a l'usuari. Aquesta vegada però heu de tenir en compte si es produeix divisió per zero. En aquest cas, NO efectuareu l'operació i mostrareu per pantalla un missatge alertant del problema.

Exercici #5> Les funcions `getchar`, `getche` i `getch`, són funcions que serveixen per a entrar caràcters, per fer-vos una idea de la diferència que hi ha entre elles, executeu el següent programa per veure la diferència de les funcions anteriors:

```
#include "stdio.h"
#include "conio.h"

void main(void)
{
    char c1,c2,c3;

    printf("El següent caràcter l'entraràs usant getchar: ");
    c1=getchar();

    printf("El següent caràcter l'entraràs usant getche: ");
    c2=getche();

    printf("\nEl següent caràcter l'entraràs usant getch: ");
```

```

    c3=getch();
}
printf("\nEls characters que has entrat són: %c %c %c %n",c1,c2,c3);
}

```

Exercici #6> A mesura que aneu fent programes cada vegada més complicats, és molt probable que necessiteu alguna mena de menú per tal que l'usuari triï entre varies opcions. Creeu una petita aplicació que simuli un menú:

És demanarà a l'usuari que entri un caràcter de la 'a' a la 'f'. Cada lletra representarà una opció diferent. Si es tria 'a' es mostrarà per pantalla el missatge: "Efectuant opció a", si es tria 'b', "Efectuant opció b" etc. Useu la estructura switch/case.

NOTA #1> Considereu el cas en què no s'ha triat cap de les opcions vàlides (default).

NOTA #2> Heu de considerar majúscules i minúscules.

Exercici #7> Quan us porteu malament a classe, és probable que el/la professor/ra us faci quedar al final per copiar 500 vegades a la pissarra alguna cosa com ara: "Estaré atent a classe". El domini de la sentència iterativa 'for' us pot ajudar bastant en aquest cas. Feu una petita aplicació en què l'usuari entri un nombre enter positiu (i gran!!!!!!), que representarà el nombre de vegades que heu de copiar una certa frase, i usant el 'for', imprimiu-la les vegades que faci falta.

Exercici #8> Escriure un programa que, primer llegeixi quatre nombres enters i després, amb una combinació de comparacions escrigui el més gran dels 4. Intenteu-ho fer utilitzant el mínim nombre possible de comparacions.

Exercici #9> Feu un petit programa que us demani que entreu un nombre real que guardareu dins la variable R. Després, usant una de les funcions getchar, getch o getch, entrareu un dels caràcters 'a', 'b', 'c' i 'd' que equivaldran a efectuar les següents operacions (usant el switch):

- Imprimir per pantalla el perímetre d'una circumferència de radi R.
- Imprimir per pantalla l'àrea d'una circumferència de radi R.
- Dir si R és més gran o més petit que 67,43.
- Llegir un altre nombre real que guardarem dins la variable R1 i imprimir per pantalla el resultat de la seva suma R+R1.

Debugueu pas a pas el programa, observeu el flux d'execució i aneu mirant quin valor van prenent les variables que hi intervenen.

Exercici #10> Realitzeu un programa que mostri per pantalla el compte enrera del llançament d'un coet (a partir de 20) utilitzant la estructura de control 'for'.

Exercici #11> Usant la sentència iterativa 'for' i l'operador '+=', sumeu els N primers nombres naturals, on N és un enter positiu entrat per l'usuari.