

Referència ràpida Pseudocodi ⇔ C

Pseudocodi	C
Definicions de tipus de dades	
var nom_var : tipus;	tipus nom_var;
tipus simples:	
a: enter;	int a;
b: real;	
trobat: booleà;	
lletra: caràcter;	
f: fitxer;	
tipus estructurats:	
t : taula[10] de enters;	
c : taula[3] de enters := {0,1,2}	
a:=c[0]; \$ a := ____ c[1]:=3; \$ c := { }	
m : taula[FIL][COL] de enters;	
n: taula[2][3] de entres := {{1,2,3},{4,5,6}}	
suma:=n[0][1]+n[0][2]+n[0][3]; \$ suma := _____	
Constants:	
constant tipus NOM=valor;	
constant real PI=3,14;	

Operadors Aritmètics	
a:=b+c; a:=b-c; a:=b*c; a:=b/c; a:=b%c;	
Asignació a:=b;	
Operadors Relacionals	
a≠b a<b a≤b a>b a≥b	
Comparació a=b	
Operadors Lògics	
(a>b) i (b<c) (a=b) o (a≠c) no (a<b)	
Entrades sortides estàndar	
escriure("El doble de",x,"es",y);	
llegir(x);	
Entrades sortides amb fitxers de text	
var f:fitxer;	
f:=obrir("nom.txt", "mode"); f:=obrir("dades.txt", "r");	
llegirf (f, dada);	
escriuref (f, dada);	
tancar(f);	

Entrades sortides amb fitxers binaris	
var f:fitxer;	
f:=obrir("nom.txt", "mode"); f:=obrir("dades.txt", "rb");	
llegirf (f, dada);	
escriuref (f, dada);	
tancar(f);	
Estructures algorismiques	
algorisme Nom és var a:enter inici \$ codi a:=3; escriure("a es: ",a); falgorisme	
si (condició) llavors \$ instruccions sino \$ instruccions fsi	
opcio (x) \$ x tipus enter o caràc. 1: escriure ("Uno"); 2: escriure("Dos"); en altre cas: escriure("Altre"); fopcio	
mentre (condició) fer \$ instruccions fmentre	
per(i:=val_ini; condició; progressió) fer \$ instruccions fper	

Tractament de seqüències

Recorregut:

```
(...)  
$ inicialitzar_var  
suma:=0;  
$ primer element  
llegir (x);  
mentre (x≠-1) fer    $no_fin: centin.  
    $ tractar  
    suma:=suma+x;  
    $ llegir següent  
    llegir(x);  
fmentre  
$tractament final  
(...)
```

Cerca:

```
(...)  
$ inicialitzar_var  
hi_ha_neg:=fals;    $ és booleà  
$ primer element  
llegir (x);  
mentre (x≠0 i no hi_ha_neg) fer  
    $ avaluar i tractar  
    si (x < 0) llavors  
        trobat:=cert;  
        $tractar si cal  
    sino  
        $ llegir següent  
        llegir(x);  
    fsi  
fmentre  
$tractament final  
(...)
```

Funcions i accions	
Funcions:	
\$ Funció 1 funció Dividir (var a,b:enter) retorna real és var divisio:real; inici <div> <div>si (b>0) llavors</div> <div>divisio:=((real)a) / b;</div> <div>sino</div> <div>divisio:= 0;</div> <div>fisi</div> </div> retorna (divisio); ffunció	
\$ Funció 2 funció Suma (t:taula de real,n:enter) <div>retorna real és</div> var sum:real; inici <div> <div>\$ codi</div> <div>retorna (suma);</div> </div> ffunció	
Accions:	
\$ Acció acció Mostra (var a:real) és var inici <div> <div>\$ mostra amb 2 decimals</div> <div>escriu("El num és", a);</div> </div> facció	

Crides:	
<pre> algorisme Prova és var x, y: enters; z:real inici llegir (x); llegir (y); z:=Dividir(x,y); Mostra(z); falgorisme </pre>	
Paràmetres per valor i per referència	
<pre> \$ a per valor \$ b per referència Func Max (var a:enter, b:enter) retorna enter és var m:enter; inici m=b; si (a>b) llavors m=a; fisi retorna (m); ffunció </pre>	
<pre> Algorisme Prova és var res, x, y: enters; z: enters; \$z és referència inici z := y; \$referència de y; res := Max (x, y); res := Max (x, z); escriure ("Màxim: ", res); falgorisme </pre> <p>¿Sòn idèntiques les crides? _____</p>	

\$ segons la definició de la Funció 2
\$ feta més amunt.
(...)
constant DIM:=20;
var suma: real;
 valors: taula[50] de real;
inici
 suma:=Suma(valors, DIM);
(...)

Altres