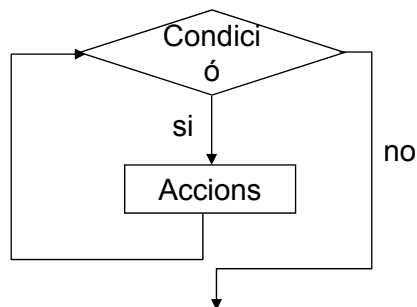


LLENGUATGES DE PROGRAMACIÓ

BLOC 1 – SEMINARI 2

ESTRUCTURES DE CONTROL ITERATIVES ARRAYS UNIDIMENSIONALS

Composició iterativa (estructura mentre): while



PSEUDOCODI

```
mentre <condició>  
  <accions>  
fi mentre
```

LLENGUATGE C

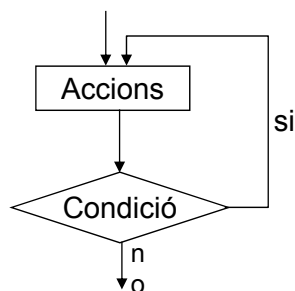
```
while (condició)  
  Acció_bucle;
```

```
while (condició)  
{  
  Acció_bucle_1;  
  ...  
  Acció_bucle_N;  
}
```

Composició iterativa (estructura mentre): while

```
/* Programa que calcula el factorial d'un nombre enter */  
  
#include <stdio.h>  
  
main()  
{  
    int n, factorial, i;  
  
    printf("Introdueix un nombre positiu:");  
    scanf("%d",&n);  
    factorial = 1;          /* Inicialització variables bucle */  
    i = 1;  
    while (i < n)          /* Condició de finalització */  
    {  
        i++;              /* Actualització variables bucle */  
        factorial = factorial * i; /* Accions del bucle */  
    }  
    printf("El factorial és: %d.",factorial);  
}
```

Composició iterativa (estructura repetir): do - while



PSEUDOCODI

```
repetir  
    <accions>  
mentre <condició>
```

LLENGUATGE C

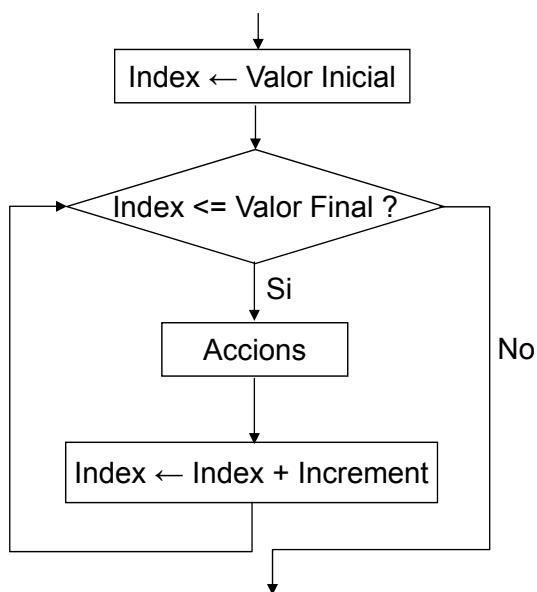
```
do  
    Acció_bucle;  
while (condició);  
  
do  
{  
    Acció_bucle_1;  
    ...  
    Acció_bucle_N;  
}  
while (condició)
```

Composició iterativa (estructura repetir): do - while

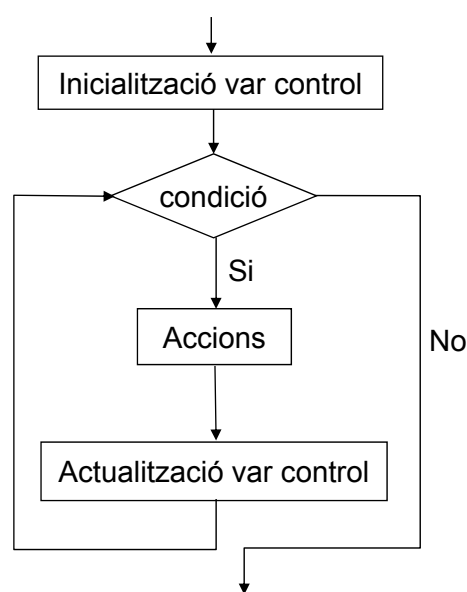
```
/* Programa que calcula el factorial d'un nombre enter */  
  
#include <stdio.h>  
  
main()  
{  
    int n, factorial, i;  
  
    printf ("Introdueix un nombre positiu:");  
    scanf ("%d",&n);  
    factorial = 1;  
    i = 1;          /* Inicialització variables bucle */  
    do  
    {  
        factorial = factorial * i;    /* Accions del bucle */  
        i++;                          /* Actualització variables bucle */  
    }  
    while (i <= n);                  /* Condició de finalització */  
    printf ("El factorial és: %d",factorial);  
}
```

Composició iterativa (estructura per): for

PSEUDOCODI



LLENGUATGE C



Composició iterativa (estructura per): for

PSEUDOCODI

```
per index  $\in$  [Valor Inicial .. Valor Final] pas n  
    <accions>  
fi per
```

LLENGUATGE C

```
for (var=var_inicial; condicio; var=expressió)  
    Acció_bucle;
```

```
for (var=var_inicial; condicio; var=expressió)  
{  
    Acció_bucle_1;  
    ...  
    Acció_bucle_N;  
}
```

Composició iterativa (estructura per): for

```
/* Programa que calcula el factorial d'un nombre enter */  
#include <stdio.h>  
  
main()  
{  
    int n, factorial, i;  
  
    printf ("Introdueix un numero positiu:");  
    scanf ("%d",&n);  
    factorial = 1;  
  
    /* Inicialització, actualització var control, condició final dins del for*/  
    for (i=2; i <= n; i++)  
        factorial = factorial * i;      /* Accions del bucle */  
  
    printf ("El factorial és: %d",factorial);  
}
```

Composició iterativa (estructura per): for

```
/* Programa d'exemple per la instrucció for */
#include <stdio.h>

main()
{
    int n, i;

    printf("Introdueix un numero positiu:");
    scanf("%d",&n);
    printf("\n");

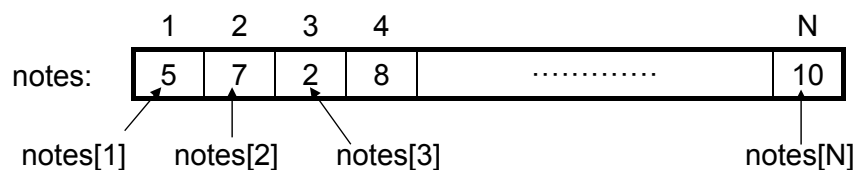
    /* Inicialització, actualització var control, condició final dins del for*/
    for (i=n; i<=20; i+=2)
    {
        printf("i=%d  ",i);          /* Escrivim per pantalla el valor de i */
        printf("i+5=%d \n",i+5);    /* Escrivim per pantalla el valor de i més 5*/
    }
}
```

```
Introdueix un número positiu: 9
i=9  i+5=14
i=11 i+5=16
i=13 i+5=18
i=15 i+5=20
i=17 i+5=22
i=19 i+5=24
```

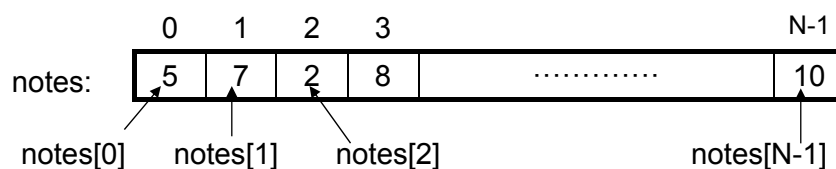
Arrays

Array: Conjunt ordenat i finit d'elements del mateix tipus (estructura homogènia)

PSEUDOCODI



LLENGUATGE C



PSEUDOCODI

Assignació

notes[2] ← 7

Entrada de dades

llegir (notes[2])

Sortida de dades

escriure (notes[2])

LLENGUATGE C

Assignació

notes[1] = 7;

Entrada de dades

scanf ("%d", ¬es[1]);

Sortida de dades

printf ("%d", notes[1]);

Declaració d'arrays

PSEUDOCODI

1.- Declaració de constants pels límits de la taula

```
const
  MIN ← <valor>
  MAX ← <valor>
ficonst
```

LLENGUATGE C

```
#define N_ELEMS <valor>
```

2.-Declaració del nou tipus de dades taula

```
tipus
  <nom_tipus_taula> : taula [MIN .. MAX] de <tipus>
fitipus
```

3.-Declaració de la variable de tipus taula

```
var
  <nom_taula> : <nom_tipus_taula>
fivar
```

```
<tipus> <nom_array> [N_ELEMS]
```

Declaració d'arrays

PSEUDOCODI

```
const
  MAX_ALUMNES←100
ficonst
tipus
  TaulaDeNotes : taula [1.. MAX_ALUMNES] de enter
fitipus
var
  notes: TaulaDeNotes
```

```
notes: [ ] [ ] [ ..... ] [ ]
```

LLENGUATGE C

```
#define MAX_ALUMNES 100
int notes[MAX_ALUMNES];
```

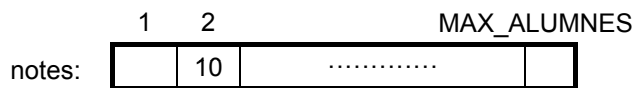
```
notes: [ ] [ ] [ ..... ] [ ]
```

Inicialització d'arrays

Accés individual als elements de l'array

<nom_taula> [<posició_element>]

<nom_array> [<posició_element>]



Exemple:

notes[2] ← 10

notes[1] = 10;

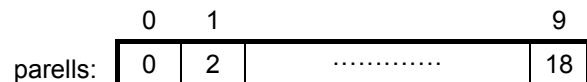
Inicialització en la declaració

```
int parells[10] = {0,2,4,6,8,10,12,14,16,18};
```

Inicialització element a element

```
int parells[10];  
int i;
```

```
for (i=0;i<9;i++)  
    parells[i]=2*i;
```



Exemple

Volem fer un programa que defineixi un vector de 10 enters i inicialitzi cada posició amb el triple del seu índex. A continuació volem que es mostri per pantalla

```
/* Programa d'exemple per arrays numèrics unidimensionals*/  
#include <stdio.h>  
  
#define N_ELEMS 10  
  
main()  
{  
    int vec[N_ELEMS];  
    int i;  
  
    for (i=0;i<N_ELEMS;i++)  
        vec[i]=3*i;  
  
    for (i=0;i<N_ELEMS;i++)  
        printf("%d",vec[i]);  
}
```