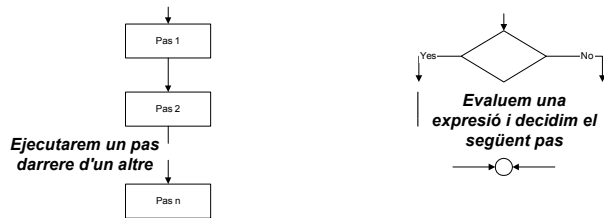


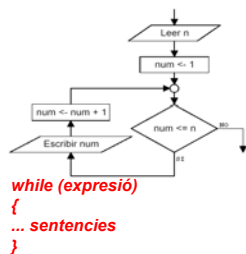
Estructures de repetició

1

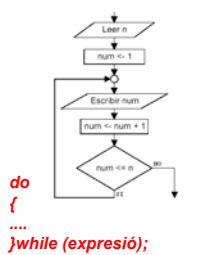


Entrada-Eixida

Estructures Iteratives (bucles)



Diagrams de flux de dades (DFD's)



2

Sentencia for

```
for(inic; cond; increm)
{
    sentencial;
    sentencia2;
}
```

Sentencia while

```
while( expresion logica)
{
    sentencial;
    sentencia2;
}
```

Sentencia do..while

```
do
{
    sentencial;
    sentencia2;
}
while (expresión lògica);
```

3

- **Exercici 1: Sumar n sencers introduïts pel teclat. Demanar primerament n, i després, la resta de nombres.**

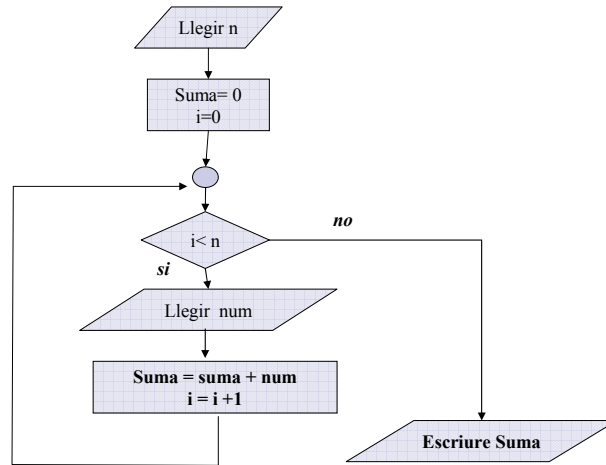
Entrades: n, num

Eixides: suma

Anàlisi: emprarem una estructura de repetició,però quina??

4

- **Exercici 1: Sumar n sencers introduïts pel teclat. Demanar primerament n, i després, la resta de nombres.**



5

```
//Programa que suma n numeros enteros
#include <iostream.h>

int main()
{
    //declaración de variables
    int n, num, suma, i;

    cout << "Introducir el valor de n:";
    cin >> n;

    suma = 0;
    cout << "Introducir " << n << " numeros enteros\n";
    for (i=0; i<n; i++)
    {
        cin >> num;
        suma = suma + num;//acumulo x en la suma
    }
    cout << "Resultado de la suma: " << suma;

    return 0;
}
```

6

- **Exercici 2: Sumar tots els numeros introduïts abans d'un 0.**

Entrades: Numeros que volem sumar...

Eixides: El resultat de la suma

Anàlisi: anem introduint sencers fins detectar el 0 ...

... però quina estructura de repetició ens convé utilitzar??

7

```
//Programa que suma numeros hasta introducir un 0=fin
#include <iostream.h>

int main()
{
    //declaración de variables
    int x;
    int suma;
    bool fin;

    //Inicializar variables
    fin = false;
    suma = 0;

    cout << "Introducir " << n << "numeros enteros (0 = fin)\n";
    do
    {
        cin >> x;
        if (x != 0)
            suma = suma + x;//acumulo x en la suma
        else
            fin = true;
    }while (fin != true);

    cout << "Resultado de la suma: " << suma;

    return 0;
}
```

8

- **Exercici 3: Trobar el mínim valor de n, tal que $2^n > 1000$**

Entrades:

Eixides: $n \rightarrow 2^n > 1000$

Anàlisi: Quina estructura de repetició empraré?

9

```
//Programa calcula el numero n tal que  $2^n > 1000$ 
#include <iostream.h>

int main()
{
    //declaración de variables
    int n; //lo que hay que buscar
    int pot_2; //para calcular la potencia de dos

    //Valores iniciales para empezar a buscar
    n = 0;
    pot_2 = 1;

    while (pot_2 <= 1000)
    {
        n = n + 1;
        //Calculo la potencia
        pot_2 = pot_2 * 2; //siguiente potencia
    }

    cout << "Resultado " << n << endl;

    return 0;
}
```

10

- **Exercici 4: Escriure un programa per a mostrar els valors de la recta $y = mx + b$ en el rang $[r0 \dots r1]$ amb el format d'eixida es el següent:**

```
Recta: y= 2x+1 Rango:[1..3]
Valor de x = 1, Valor de y = 3
Valor de x = 2, Valor de y = 5
Valor de x = 3, Valor de y = 7
```

Entradas:
Salidas:
Análisis :

11

```
#include <iostream.h>

int main()
{
    int m, b; //coeficientes de la ecuacion de la recta
    int r0,r1; //puntos extremos del intervalo
    int x, y; //para el bucle for

    cout << "Introducir coeficientes de la recta: y=mx+b\n";
    cin >> m >> b;

    cout << "Introducir r0 y r1, tal que (r0 < r1) \n";
    cin >> r0 >> r1;

    cout << "Recta: "<<< "y=" << m << "x + " << b ;
    cout << "\tRango: [" << r0 <<" " << r1 << "]"";
    //Calculo los puntos de la recta
    for (x = r0 ; x <= r1; x++)
    {
        //Calculo valores de y para cada valor de x
        y = m * x + b;

        //Muestro resultado
        cout << " Valor de x: " << x << ", Valor de y: " << y <<endl;
    }

    return 0;
}
```

12

■ Exercici 5: Calcular quantes vocals hem introduït pel teclat abans del caràcter '#'

Entrades: Seqüència de caràcters (inclòs el '#')

Eixides: La quantitat de vocals que conté la seqüència (total)

Anàlisi: Anem llegint caràcters i si detectem una vocal incrementarem un comptador (cont), como sabemos si es una vocal???

13

```
//Programa que cuenta el numero de vocales en una secuencia de caracteres
#include <iostream.h>

int main()
{
    //declaración de variables
    char ch;
    int cont; //numero de vocales

    //lectura de datos
    cout << "Introducir caracteres (# = fin)";
    cont = 0; //todavia no hay vocales
    do
    {
        cin >> ch; //leo carácter
        switch(ch)
        {
            case 'a': case 'A':
            case 'e': case 'E':
            case 'i': case 'I':
            case 'o': case 'O':
            case 'u': case 'U': cont = cont + 1;
                               break;

        }
    }while( ch != '#');

    cout << "Vocales introducidas: " << cont << endl;

    return 0;
}
```

14

- **Exercici 6:** Escriure un programa per a jugar al *master mind*. En primer lloc, el jugador 1 introduirà una combinació (secreta) de 4 numeros sense repeticions. Posteriorment, esborrem la pantalla, i el jugador 2 deurà de introduir combinacions fins encertar la combinació secreta. Tras cada cada combinació intrduida pel jugador 2, mostrarem el numero de encerts obtinguts (es a dir, quantes posicions son correctes). Finalment el programa mostrarà el numero d'intents empleats en obtindre la combinació secreta.

Entrades:

Eixides:

Anàlisi:

15

```
#include <iostream.h>

int main()
{
    int c1, c2, c3, c4;
    int n1, n2, n3, n4;
    int intentos, aciertos;
    bool terminar = false;

    cout << "Jugador 1: Introduce 4 numeros de la combinacion (c1 c2 c3 c4)\n";
    cin >> c1 >> c2 >> c3 >> c4;
    //limpio la pantalla
    system("CLS");
    intentos = 0;
    do{
        cout << "Jugador 2: introduce jugada ( c1 c2 c3 c4)\n";
        cin >> n1 >> n2 >> n3 >> n4;
        intentos = intentos + 1;
        aciertos = 0;
        if (c1 == n1) aciertos = aciertos + 1;
        if (c2 == n2) aciertos = aciertos + 1;
        if (c3 == n3) aciertos = aciertos + 1;
        if (c4 == n4) aciertos = aciertos + 1;
        //Compruebo si ya se termina el juego
        if (aciertos == 4)
            terminar = true;
        cout << "Numero de aciertos: " << aciertos;
    }while(terminar !=true);
    cout << "Numero de intentos: " << intentos;
    return 0;
}
```

16