

# EXERCICIS L2

**Operadors**

## Exercici 2. És digit?

Quina complicació!

```
es_digit = (lletra-'0'==0)|| (lletra-'0'==1)||  
(lletra-'0'==2)|| (lletra-'0'==3)|| (lletra-'0'==4)||  
(lletra-'0'==5)|| (lletra-'0'==6)|| (lletra-'0'==7)||  
(lletra-'0'==8)|| (lletra-'0'==9);
```



Per fer això, es podria haver fet directament això (però tampoc està bé!):

```
es_digit = (lletra=='0')||(lletra=='1')||  
(lletra=='2')||(lletra=='3')||(lletra=='4')||  
(lletra=='5')||(lletra=='6')||(lletra=='7')||  
(lletra=='8')||(lletra=='9');
```

OK, però no es demanava això

```
if((lletra >= '0') && (lletra <= '9')){  
    es_digit = true;  
}  
else{  
    es_digit = false;  
}
```



Correcte però es podria entendre millor

```
es_digit= 48<=lletra && lletra<=57 ;
```

Correcte:

```
es_digit = (lletra >= '0') && (lletra <= '9');
```



## Exercici 2. És digit?

Aquí me l'has colat

```
ch = lletra;  
num = ch - '0';
```

```
/* Escriu l'expressio que retorna cert si la  
lletra es un digit*/  
if (num <= 9){  
    es_digit = true;  
  
}  
else{  
    es_digit = false;  
}
```



No calia fer cap typecast:

```
/* Escriu l'expressio que retorna cert si la lletra es  
un digit*/  
es_digit = ((int)lletra >= 48 && (int)lletra <= 57);
```

Què passa si lletra = '!' (codi ASCII 33?)

Per altra banda, es demanava expressió.

## Exercici 3. És parell?

**OJO! Estàs barrejant tipus. `es_parell` és enter o booleà?**

```
int es_parell;  
es_parell = enter % 2;  
  
if(es_parell == 0){  
    es_parell = true;  
}  
else{  
    es_parell = false;  
}
```



**Correcte:**

```
es_parell = (enter % 2) == 0;
```



# Exercici 4. Àrea de la circumferència

Per què tants càstings?

```
float area = (float) radi * (float) radi * PI;
```



**Correcte:**

```
area = PI * radi * radi
```



**També correcte**

```
#include <math.h>
...
area = PI * pow(radi, 2)
```



# **EXERCICIS L3**

**Estructures de Control**

# Exercici 1. Són divisibles?

**Aquesta estructura del codi és inacceptable**

```
if(a%b==0){printf("%d es divisible entre %d\n", a, b); }  
    else{printf("%d NO es divisible entre %d\n", a, b);}
```



**El mateix codi ben estructurat:**

```
if (a%b == 0){  
    printf("%d es divisible entre %d\n", a, b);}  
else {  
    printf("%d NO es divisible entre %d\n", a, b);  
}
```

## Exercici 2. Valor absolut

Aquesta solució vol ser intel·ligent però no ho és tant  
(possible problema d'overflow en  $2 * \text{nombre}$ )

```
if(nombre<0){valor_abs=nombre-(2*nombre);}
else{valor_abs=nombre;}
```



La solució “normal” és més segura:

```
if(nombre<0){
    valor_abs=-nombre;
}
else{
    valor_abs=nombre;
}
```

Quin llenguatge de programació és aquest?

```
if (nombre < 0) do;
    valor_abs = -nombre;
else valor_abs = nombre;
```






# Exercici 3. Temperatura corporal

Aviam, per què he definit jo unes constants???

```
#define LIM_HIP 35.5
#define LIM_NORM 37.5
#define LIM_FEB 41
...
if(temp < 35.5){
    printf("HIPOTERMIA\n");
}
else if(temp >= 35.5 && temp < 37.5){
    printf("NORMAL\n");
}
else if(temp >= 37.5 && temp <= 41){
    printf("FEBRE\n");
}
else if(temp > 41){
    printf("HIPERPIREXIA\n");
}
```



Recordau

Hard-coding NO

```
...
a = b + 100;
c = 100 + a;
...
```



**const**

```
    MAX := 100;
```

**fconst**

```
...
a = b + MAX;
c = MAX + a;
```

...



# Exercici 4. Creixents, decreixents, desordenats

Vigileu amb els operadors que feu servir. La AND és &&, en canvi & és una altra cosa (bitwise and).

```
if((a < b) & (b < c)){
    printf("ORDENATS CREIXENT\n");
}
else{
    if((a > b) & (b >c)){
        printf("ORDENATS DECREIXENT\n");
    }
    else{
        if((a == b) & (b == a)){
            printf("ORDENATS CREIXENT");
        }
        else{
            printf("DESORDENATS\n");
        }
    }
}
```

# Exercici 5. Switch operació

Em sembla perfecte declarar una variable "res",  
però treu-li partit! Codi de printf duplicat!

Estem una mica distrets...

```
switch (c) {  
case '+':  
c = a + b ;  
break;  
case '*' :  
c = a * b;  
break;  
case '-' :  
c = a - b;  
break;  
case '/' :  
c = a / b;  
break;  
default:  
printf ("ERROR");  
}
```



... i feu el favor d'indentar

```
switch (c) {  
    case '+' :  
        res = a + b;  
        printf("%d\n", res);  
        break;  
    case '-' :  
        res = a - b;  
        printf("%d\n", res);  
        break;  
    case '*' :  
        res = a * b;  
        printf("%d\n", res);  
        break;  
    case '/' :  
        res = a / b;  
        printf("%d\n", res);  
        break;  
    default :  
        printf("ERROR\n");  
}
```

# Exercici 5. Switch operació

COMORRR???

```
switch (c) {  
    case '+': printf("15");  
               break;  
    case '-': printf("5");  
               break;  
    case '*': printf("50");  
               break;  
    case '/': printf("2");  
               break;  
    default:  
        printf("ERROR");  
}
```



# Exercici 5. Switch operació

Alumne que s'ha oblidat que existeix la clàusula "default" del switch

```
switch (c) {  
    case '+' :  
        res = a + b;  
        printf("%d\n", res);  
        break;  
    case '-' :  
        res = a - b;  
        printf("%d\n", res);  
        break;  
    case '*' :  
        res = a * b;  
        printf("%d\n", res);  
        break;  
    case '/' :  
        dec = a / b;  
        printf("%d\n", dec);  
        break;  
}  
if((c != '+') & (c != '-') & (c != '*') & (c != '/')){  
    printf("ERROR");  
}
```



# Exercici 6. Switch de la lletra del DNI

Seguim sense repassar com funciona el switch

```
switch (lletra_dni){  
    case 0: return 'T';  
    break;  
    case 1: return 'R';  
    break;  
    case 2: return 'W';  
    break;  
    case 3: return 'A';  
    break;  
    case 4: return 'G';  
    break;  
    case 5: return 'M';  
    break;  
    case 6: return 'Y';  
    break;
```

...





# Exercici 6. Switch de la lletra del DNI



**De veritat creieu que us posaria un programa on la solució fos aquesta???**

Com si la solució amb el switch no fos suficientment pesada...

```
if (dni%23==0){
    lletra_dni='T';
}
if (dni%23==1){
    lletra_dni='R';
}
if (dni%23==2){
    lletra_dni='W';
}
if (dni%23==3){
    lletra_dni='A';
}
if (dni%23==4){
    lletra_dni='G';
}
if (dni%23==5){
    lletra_dni='M';
}
```

```
if (dni%23==6){
    lletra_dni='Y';
}
if (dni%23==7){
    lletra_dni='F';
}
if (dni%23==8){
    lletra_dni='P';
}
if (dni%23==9){
    lletra_dni='D';
}
if (dni%23==10){
    lletra_dni='X';
}
if (dni%23==11){
    lletra_dni='B';
}
```

```
if (dni%23==12){
    lletra_dni='N';
}
if (dni%23==13){
    lletra_dni='J';
}
if (dni%23==14){
    lletra_dni='Z';
}
if (dni%23==15){
    lletra_dni='S';
}
if (dni%23==16){
    lletra_dni='Q';
}
if (dni%23==17){
    lletra_dni='V';
}
```

```
if (dni%23==18){
    lletra_dni='H';
}
if (dni%23==19){
    lletra_dni='L';
}
if (dni%23==20){
    lletra_dni='C';
}
if (dni%23==21){
    lletra_dni='K';
}
if (dni%23==22){
    lletra_dni='E';
}
```



## Exercici 6. Switch de la lletra del DNI

Altres cop repetint codi !!!

```
switch (res) {  
  
    case 0 :  
        lletra_dni = 'T';  
        printf("NIF: %d-%c\n", dni, lletra_dni);  
        break;  
  
    case 1 :  
        lletra_dni = 'R';  
        printf("NIF: %d-%c\n", dni, lletra_dni);  
        break;  
  
    case 2 :  
        lletra_dni = 'W';  
        printf("NIF: %d-%c\n", dni, lletra_dni);  
        break;
```

... i així fins a 23 vegades.

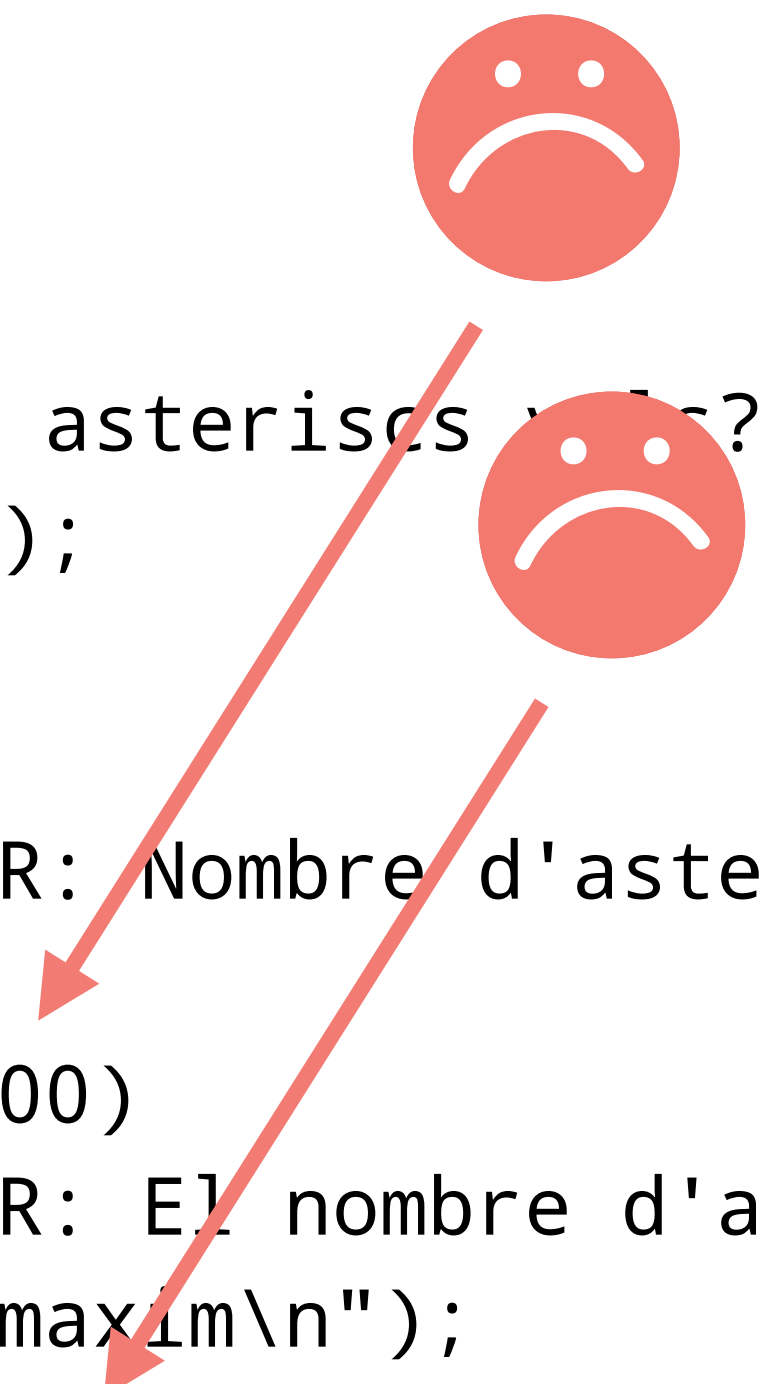




# Exercici 7. Bucle asteriscs

**NO HARD CODING!**

```
int n;
printf("Quants asteriscs vols?\n");
scanf("%d", &n);
int i;
if (n < 0)
    printf("ERROR: Nombre d'asteriscs no pot
ser negatiu\n");
else if (n > 100)
    printf("ERROR: El nombre d'asteriscs
demanat supera el maxim\n");
else if (n <= 100 && n >= 0)
    for (i = 0; i < n; i++)
        printf("*");
return 0;
```



```
#define MAX 100
```

```
int main(){
```

```
    int n,i;
```

```
    printf("Quants asteriscs vols?\n");
```

```
    scanf("%d", &n);
```

```
    if (n>100){
```

```
        printf("ERROR: El nombre d'asteriscs dema
supera el maxim\n");
```

```
    }
```

```
    else if (n<0){
```

```
        printf("ERROR: Nombre d'asteriscs no pot
negatiu\n");
```

```
    }
```

```
    else {
```

```
        for(i=1;i<=n;i++){
```

```
            printf("*");
```

```
        }
```

```
    }
```

# Exercici 7. Bucle asteriscs

I el bucle?

```
if (n<0){
    printf("ERROR: Nombre d'asteriscs no pot ser negatiu\n");
}
else if (n>MAX){
    printf("ERROR: El nombre d'asteriscs demanat supera el maxim\n");
}
else if (n<=100){
    printf("*\n");
}
```

# Exercici 8. Comptar al revés

Quina estructura té aquest for? Qui ho ha programat sap per què funciona?

```
int n,i;
printf("Introdueix un nombre\n");
scanf("%d", &n);
if (n<0){
    printf("ERROR: No s'accepten nombres negatius\n");
}
else if(n>150){
    printf("ERROR: Nombre massa gran\n");
}
else {
    for (i=n+1;i--;){
        printf ("%d\n",i);
    }
}
```

**PROHIBIDÍSSIM FER SERVIR EL FOR AIXÍ**



## Exercici 8. Comptar al revés

```
int main() {  
    OJO CUIDAO  
  
    int n;  
    printf("Introdueix un nombre\n");  
    scanf("%d", &n);  
  
    /* Descomenteu els següents printf i col·loqueu-los al codi on toqui */  
    if (n < 0) {  
        printf("ERROR: No s'accepten nombres negatius\n");  
    }  
  
    else if (n > MAX) {  
        printf("ERROR: Nombre massa gran\n");  
    }  
    else {  
        for (n = n; n >= 0; n--) {  
            printf("%d\n", n);  
        }  
    }  
    return 0;  
}
```

## Exercici 8. Comptar al revés

```
int n, i;
printf("Introdueix un nombre\n");
scanf("%d", &n);

if (n < 0){
printf("ERROR: No s'accepten nombres negatius\n");
}
else{
if (n > MAX){
printf("ERROR: Nombre massa gran\n");
}
else{
printf("%d\n",n);
for (i = n; i > 0; i--){
n=n-1;
printf("%d\n",n);
}
}
}
```

**Es pot simplificar:**

```
for (i = n; i >= 0; i--){
printf("%d\n",i);
}
```

## Exercici 8. Comptar al revés

**Exemple de codi que es complica la vida:** No calen tantes variables ni tampoc avaluar  $n \leq \text{MAX}$  cada cop. "n" canvia?

```
int n, i, m;
printf("Introdueix un nombre\n");
scanf("%d", &n);

if (n < 0) {
    printf("ERROR: No s'accepten nombres negatius\n");
}
if (n > MAX) {
    printf("ERROR: Nombre massa gran\n");
}

i = 0;
m = n - i;

for (i = 0; i <= n && n <= MAX; i++) {
    m = n - i;
    printf("%d\n", m);
}
```

# Exercici 8. Comptar al revés

Típic exemple de solució “amb parches” perquè funcionin tots els casos

```
if ((n <= MAX) & (n > 0)){  
    for (i = n; i >= 0; i--){  
        printf("%d\n", i);  
    }  
}
```

Només posant un `>=` aquí ja ens estalviàvem aquest cas particular

```
else{  
    if (n < 0){  
        printf("ERROR: No s'accepten nombres negatius\n");  
    }  
    else{  
        if (n == 0){  
            printf("0");  
        }  
        else{  
            printf("ERROR: Nombre massa gran\n");  
        }  
    }  
}
```

(I una indentació que dificulta molt la feina)

# Exercici 8. Comptar al revés

La solució era ben senzilla:

```
int n;
printf("Introdueix un nombre\n");
scanf("%d", &n);

if (n < 0){
    printf("ERROR: No s'accepten nombres negatius\n");
}
else if (n > MAX){
    printf("ERROR: Nombre massa gran\n");
}
else{
    for(int i=n; i>=0; i--){
        printf("%d\n",i);
    }
}
```



# Exercici 9. Rectangle d'asteriscs

Exemple de com la reutilització de variables crea codis més difícils d'interpretar

```
if(fil<0 || col<0){
    printf("ERROR: Nombre de files/columnes no pot ser negatiu\n");
}
else if(fil>MAX_FILES || col>MAX_COLS){
    printf("ERROR: Dimensions superen el maxim\n");
}
else{
    for (fil=fil; fil>0; fil--){
        i=col;
        for(i=i;i>0;i--){
            printf("*");
        }
        printf("\n");
    }
}
```

# Exercici 9. Rectangle d'asteriscs

**I aquest "else if" gratis?**

```
if (fil>MAX_FILES || col>MAX_COLS){
    printf("ERROR: Dimensions superen el maxim\n");
}else if (fil<0 || col<0){
    printf("ERROR: Nombre de files/columnes no pot ser negatiu\n");
}else if (fil==0 || col==0){
}else{
    for (i=0; i<fil; i++){
        for(j=0; j<col; j++){
            printf("*");
        }
        printf("\n");
    }
}
```

# Exercici 9. Rectangle d'asteriscs

Tornem-hi a no fer servir les constants

```
if (fil < 0 || col < 0){
    printf("ERROR: Nombre de files/columnes no pot ser negatiu\n");
}
else{
    if (fil > 20 || col > 20){
        printf("ERROR: Dimensions superen el maxim\n");
    }
    else{
        for (i = 0; i < fil; i++){

            for (n = 0; n < col; n++){
                printf("*");
            }
            printf ("\n");
        }
    }
}
```

## Exercici 9. Rectangle d'asteriscs

**Altra vegada: “fil” i “col” canviaran de valor durant l'execució del bucle? No, oi? Doncs per què hem de comprovar la condició cada cop?**

```
if (fil < 0 || col < 0) {
    printf("ERROR: Nombre de files/columnes no pot ser negatiu\n");
}
if (fil > MAX_FILES || col > MAX_COLS) {
    printf("ERROR: Dimensions superen el maxim\n");
}

i = 0;
m = 0;
for (m=0; m < fil && fil <= MAX_FILES; m++){
    for (i=0; i < col && col <= MAX_COLS; i++) {
        printf("*");
    }
    printf("\n");
}
```

# Exercici 10. Torre d'asteriscs

**Solució correcta:**

```
if(n < 0){
    printf("ERROR: Nombre de linies no pot ser negatiu\n");
} else if (n > MAX_LINIES){
    printf("ERROR: El nombre de linies demanat supera el maxim\n");
} else{
    for(int i=1; i<=n; i++){
        for(int j=0; j<i; j++){
            printf("*");
        }
        printf("\n");
    }
}
```

# Exercici 10. Factorial

OTRA VES???

```
if (n < 0) {  
    printf("ERROR: No es pot calcular el factorial d'un nombre negatiu\n");  
}  
for (i = 1; 0 < i && i <= n; i++){  
    fact = fact * i;  
}  
if ( n >= 0) {  
    printf("%d! = %d\n",n,fact);  
}
```

# Exercici 12. El Lloro

No usar bé el condicional booleà

```
do{
    printf("Introdueix un numero\n");
    scanf("%d", &numero);
    if (numero != -1){
        printf("LLORO: %d!\n", numero);
    }
}
while (numero != -1);
```



```
acabar = false;
```

```
do{
    printf("Introdueix un numero\n");
    scanf("%d", &numero);
    if (numero == -1){
        acabar = true;
    }
    else{
        printf("LLORO: %d!\n", numero);
    }
}
while (!acabar);
```



# Exercici 12. El Lloro

**Què fa aquesta variable C ?**

```
int numero,c=0;
bool acabar;
acabar = 0;

do{
    if(c==0){c++;}
    else{printf("LLORO: %d!\n", numero);}
    printf("Introdueix un numero\n");
    scanf("%i",&numero);
    acabar=(numero==-1);

}
while (!acabar);
```

**El proper dia penalitzaré els codis indentats així**



## Exercici 12. El Lloro

**El problema d'imprimir primer i comprovar després. Què imprimeix aquest codi quan l'usuari introdueix el codi "-1"?**

```
acabar = false;
do{
    scanf("%d",&numero);
    printf("LLORO: %d!\n", numero);
    if (numero==-1){
        acabar = true;
    }
}
while (!acabar);
```

## Exercici 12. El Lloro

Aquest és només un exemple de programes que no funcionen perquè no pareu atenció

```
acabar = 0;
do{

    printf("Introdueix un numero\n");
    scanf("%d", &numero);
    printf("LLORO: %d!\n", numero);
}
printf("LLORO: %d!\n", numero);
while (!acabar);
```

Syntax Error(s)

\_\_tester\_\_.c: In function 'main':

\_\_tester\_\_.c:20:5: error: expected 'while' before 'printf'

```
    printf("LLORO: %d!\n", numero);
```

^~~~~~

# Exercici 12. El Lloro

Per què no funciona aquest programa?

```
int numero;  
bool acabar;  
acabar = (-1);  
do{  
  
    printf("Digues un número:\n");  
    scanf("%d", &numero);  
    if(numero != -1){  
        printf("%d!", numero);  
    }  
  
}  
while (numero != acabar);
```

# Exercici 12. Màquines recreatives

Feu servir les variables de manera que s'entengui què fa

```
bool sortir;  
int n;  
// Inicialitza la variable "sortir". A quin valor???  
sortir = 1; <— Primer, hauria de ser "true". Segon, sortir=cert des de l'inici???  
/* Escriu la condicio del while de tal manera que nomes es pugui  
jugar si encara tenim prou saldo i no volem sortir*/  
while (saldo>=cost_partida && sortir == 1){  
    printf("Saldo: %d\n",saldo);  
    printf("Jugant...\n");  
    printf("Partida acabada. Vols una altra partida?\n");  
    printf("Introdueix 0 per acabar\n");  
    printf("Introdueix 1 o qualsevol altre nombre per una nova partida\n");  
    scanf("%d",&n);  
    if (n == 0){ // L'usuari vol sortir  
        sortir = 0;  
    }  
    // Actualitzem la variable saldo  
    saldo = saldo-cost_partida;  
}
```

**ELS BOOLEANS**, o els declareu com a "int" i assigneu valors 0 i 1 (com si féssim servir un estàndard de C que no permet booleans). O bé, si els declareu com a "bool", llavors heu de fer servir "true" i "false".

# Exercici 12. Màquines recreatives

## Aquest codi té diversos problemes

```
// Inicialitza la variable "sortir". A quin valor???
bool sortir;
int n;
sortir = 0;
while (n!=0 && saldo >= COST_PARTIDA)
{
    printf("Saldo: %d\n",saldo);
    printf("Jugant...\n");
    printf("Partida acabada. Vols una altra partida?\n");
    printf("Introdueix 0 per acabar\n");
    printf("Introdueix 1 o qualsevol altre nombre... partida\n");
    scanf("%d",&n);
    if (n == 0){
        sortir = 0;
    }
    saldo = saldo - COST_PARTIDA;
}

if (n == 0){
    printf("S'ha acabat perquè has volgut sortir\n");
}
if (saldo < cost_partida) {
    printf("S'ha acabat perquè no et queden diners\n");
}
if (sortir){ // Això no s'executa mai
    printf("S'ha acabat perquè has volgut sortir\n");
}
```

# Exercici 12. Màquines recreatives

Un altre exemple de codi que costa d'entendre perquè no es respecta què significa el nom de la variable

```
while ((saldo>=cost_partida) && sortir!=false)
{
    printf("Saldo: %d\n",saldo);
    printf("Jugant...\n");

    printf("Partida acabada. Vols una altra partida?\n");
    printf("Introdueix 0 per acabar\n");
    printf("Introdueix 1 o qualsevol altre nombre per una nova
partida\n");
    scanf("%d",&n);
    if (n == 0){
        //L'usuari ha triat que vol sortir. Actualitza la variable per
reflectir aquest canvi.
        sortir = false;
    }
    // Actualitzem la variable saldo
    saldo = saldo-cost_partida;
}
```

**sortir != false vol dir que farem aquest bucle mentre sortir sigui cert**

**sortir = false? Si t'acaba de dir que vol sortir!**

# Exercici 12. Màquines recreatives

Seguim sense estar atents...

```
while (saldo>SALDO);
```

**Un punt-i-coma després del while. Perfecte perquè s'executi infinitament**

```
{
    printf("Saldo: %d\n",saldo);
    printf("Jugant...\n");

    printf("Partida acabada. Vols una altra partida?\n");
    printf("Introdueix 0 per acabar\n");
    printf("Introdueix 1 o qualsevol altre nombre per una nova
partida\n");
    scanf("%d",&n);
    if (n == 0){
        //L'usuari ha triat que vol sortir. Actualitza la variable per
reflectir aquest canvi.
        sortir = true;
    }
    // Actualitzem la variable saldo
    saldo = SALDO-2;
}
```

**Quant creieu que dona "saldo"?**

```
#include <stdio.h>
```

```
int main(){
```

```
    int n;
```

```
    while(n < 10);
```

```
    printf("S'imprimira aquesta frase?? \n");
```

```
}
```

# Exercici 12. Màquines recreatives

**No a modificar variables per forçar la sortida**

```
while (saldo>0){
    printf("Saldo: %d\n",saldo);
    printf("Jugant...\n");
    printf("Partida acabada. Vols una altra partida?\n");
    printf("Introdueix 0 per acabar\n");
    printf("Introdueix 1 o qualsevol altre nombre per una nova partida\n");
    scanf("%d",&n);
    if (n == 0){
        //L'usuari ha triat que vol sortir. Actualitza la variable per reflectir aquest canvi.
        saldo = 0;
    }
    // Actualitzem la variable saldo
    saldo = saldo - cost_partida;

}
/* Un cop s'ha acabat el bucle, podem imprimir les variables de control per saber
per quin motiu hem sortit del bucle. */
if (n != 0) {
    printf("S'ha acabat porque no et queden diners\n");
}
if (n==0){
    printf("S'ha acabat porque has volgut sortir\n");
}
```



# Exercici 13. Múltiples de 3

Us heu complicat un munt la vida! MUAJAJJAA

```
if (n <= MAX){
    i = n;    /* Bucle que recorre tots els
enters des de i=n fins i=MAX i comprova si
cadascun dels valors es multiple de 3. Si ho
es, l'imprimeix.*/
    while(i <= MAX){
        if (i % 3 == 0){
            printf("%d\n",i);
        }
        i++;
    }
}
```

Però entenc la vostra intenció.

En aquest cas, la solució eficient seria:

```
int n;
int i;
printf("Introdueix un nombre menor que %d\n",MAX);
scanf("%d", &n);

if (n <= MAX){
    //Al sortir d'aquí, n serà múltiple de 3
    while(n%3 != 0){
        n++;
    }
    //Ara ja podem anar de 3 en 3
    for(int i=n; i<MAX; i=i+3){
        printf("%d\n",i);
    }
}
```