

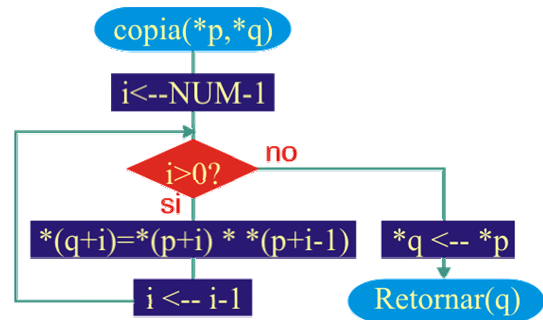
8.8 Exemples (II)



- Escriure una funció que, emprant apuntadors, copii un array A en un altre B.

```

/*
Copiar l'array A en B de forma que cada B[i]=A[i-1]*A[i], B[0]=A[0].
*/
#include <stdio.h>
#define NUM 10 //Nº elements dels arrays
void copia (int *, int *); //Declaració còpia
void main()
{
    int A[]={1, 2, 3, 4, 5, 6, 7, 8, 9 ,10};
    int B[NUM];
    copia(A, B); //Crida còpia
    for (int i=0; i<NUM; i++) printf("%i ", B[i]);
    printf("\n");
}
void copia (int *p, int *q) //Definició còpia
{
    for (int i=NUM-1; i>0; i--) *(q+i)=*(p+i) * *(p+i-1);
    *q = *p;
}
    
```



L'apuntador

15

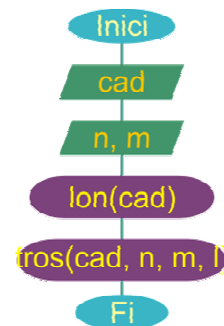
8.8 Exemples (III)



Fer una funció que imprimeixi **n** caràcters a partir de la posició **m** d'una cadena de **l** caràcters. Fer també la rutina de detecció del número de caràcters del string.

```

//Rutina principal
#include <stdio.h>
#define N 80
int lon(char*); //Detecció de longitud
void tros (char*, int, int, int); //Impressió
void main()
{
    int l, n, m;
    char cad[N];
    printf("Entra la cadena: \n\t\t");
    fflush(stdin);
    gets(cad);
    printf("Quants caracters vols i a partir de quin? \n\t\t");
    fflush(stdin);
    scanf("%i%i", &n, &m);
    l=lon(cad);
    tros(cad, n, m, l);
}
    
```



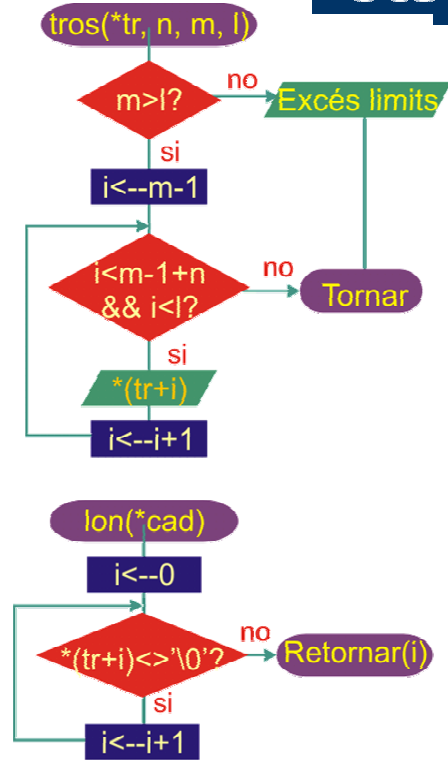
L'apuntador

16

8.8 Exemples (IV)

```
//Procediment d'impressió
//Paràmetres: cadena, n, m i l
void tros (char *tr, int n, int m, int l)
{
    printf("\n\t\t");
    if (m>l) printf ("Limits excedits!");
    else
        for (int i=(m-1); i<(m-1+n) && i<l; i++)
            printf("%c", *(tr+i));
    printf("\n\n");
}
```

```
//Procediment detecció de longitud
//Paràmetres: cadena,
int lon (char *tr)
{
    int i=0;
    while (*(tr+i)!='\0') i++;
    return(i);
}
```



8.8 Exemples (V)

Calcular els salts del cavall donada una posició inicial en un tauler d'escacs.

Per a la seva realització es tindrà present:

- Cada posició del tauler es representarà per una estructura coordenada de dos eixos.
- Les entrades i sortides vindran donades en el format que va d'a1 a h8.
- El resultat s'emmagatzemarà en un array de 8 punts coordinats (nombremàxim de salts).
- El programa constarà de les funcions: entrada posició inicial, càlcul salts i impressió resultat.
- En les funcions s'emplearà, sempre que calgui, el pas de paràmetres per adreça.

Nomenclatura: El tauler d'escacs es defineix com una matriu de dues dimensions, on les cel·les en l'eix X s'etiqueten per lletres que van de la a a la h i en l'eix de les Y pels nombres 1 a 8.

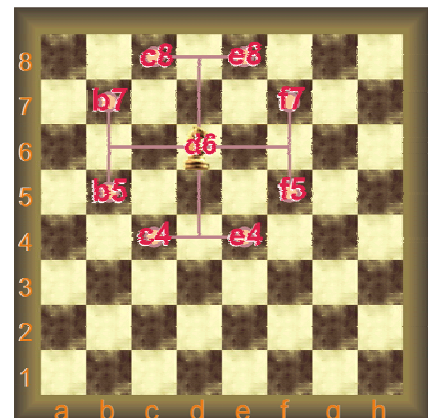
El cavall se situarà en una cel·la (d6, per exemple). Aleshores, el cavall pot saltar fins a 8 llocs diferents, tal com mostra la figura.

Així, l'estructura que es crea per a tractar el problema és:

```
struct coord
{
    char x;
    int y;
};
```

Així, la posició inicial s'entra directament amb, per exemple, d6.

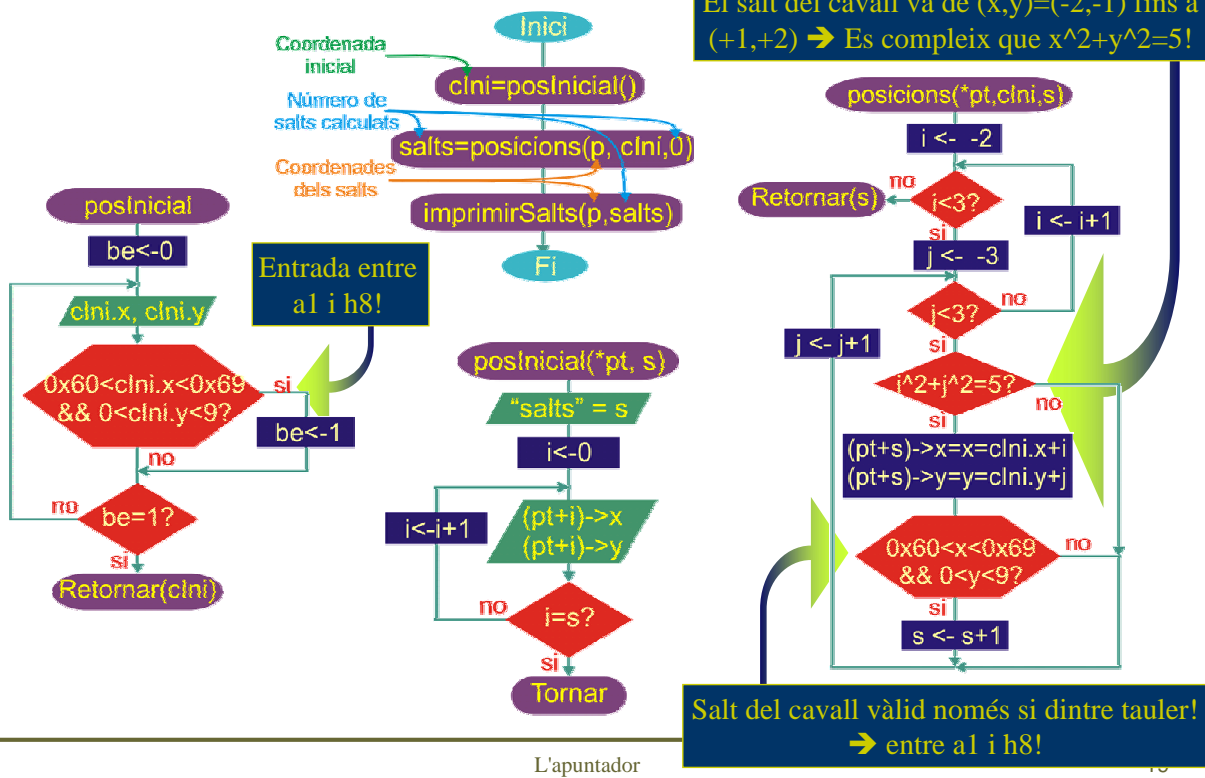
El tractament del caràcter de l'abscisa es fa directament a partir del seu cosí ASCII.



8.8 Exemples (VI)



//Diagrama de flux del programa: es dóna el programa principal i les tres funcions.



8.8 Exemples (VII)



//CODIFICACIÓ DEL PROGRAMA

//La codificació de la funció main és immediata: només distribueix tasques!

#include <stdio.h>

#define DIM 8 //Dimensió del tauler

struct coord //Estructura coordenada

```
{
    char x;
    int y;
};
```

Estructura coord

coord posnicial(void); //Posició inicial

int posicions(coord *, coord, int); //Càlculs

void imprimirSalts(coord *, int); //Impressió resultat

Declaració funcions

void main () //Programa principal

```
{
    int salts; //Variable compteig salts vàlids
    coord p[DIM]; //Array on es posen els salts
    coord clni; //Posició inicial del cavall
    printf("SALTS DEL CAVALL DONADA POSICIO INICIAL\n");
```

```
    clni=posnicial(); //Entrada cel.la inicial
    salts=posicions(p, clni, 0); //Càlcul salts possibles
    imprimirSalts(p, salts); //Impressió salts possibles
```

Programa principal

```
}
```

8.8 Exemples (VII)



```
//Entrada de la coordenada inicial
coord posInicial(void)
{
    int be=0; //S'activa en donar posició correcta
    coord clni; //Guarda coordenada inicial

    do
    {
        printf("Posicions del cavall possibles d'a1 a h8\n");
        printf("\nEntra posicio inicial: ");
        fflush(stdin);
        scanf("%c%i", &clni.x, &clni.y);
        if (clni.x>0x60 && clni.x<0x69 && clni.y>0 && clni.y<9) be=1; //X entre a i h, Y entre 1 i 8
    }while(!be);
    return (clni); //La funció retorna la posició inicial
}

//Procediment d'impressió
void imprimirSalts(coord *pt, int s) // *pt recull les posicions dels s salts possibles
{
    printf("\n--> %i salts possibles del cavall:", s);
    for (int i=0;i<s && i<DIM;i++)
        printf(" %c%i ", (pt+i)->x, (pt+i)->y); //S'empra notació -> d'estructures
    printf("\n\n");
}
}
```

Entrada punt inicial

Impressió salts

L'apuntador

21

8.8 Exemples (VII)



```
//Funció de càlcul de les posicions de salt
int posicions(coord *pt, coord clni, int s) //Es passa: adreça de l'array de les posicions de salt
{ // coordenada inicial i variable de compteig dels salts bons
    int i,j;
    char x; //Coordenada x calcul intermig
    int y; //Coordenada y calcul intermig

    for (i=-2;i<3;i++)
        for (j=-2;j<3;j++)
            if (i*i+j*j==5) //Condicció a complir-se en un salt
            {
                (pt+s)->x = x = clni.x+i;
                (pt+s)->y = y = clni.y+j;
                if (x>0x60 && x<0x69 && y>0 && y<9) s++; //Condicció de salt dintre del tauler
            }
    return(s); //Es retornen el número de salts trobats correctes
}
}
```

Càlcul dels salts

L'apuntador

22