

## Problemes capítol 8

Realitzar cada exercici donant el pseudocodi i/o el diagrama de flux i el codi C.

Considerar, en cada cas, els procediments/funcions que simplifiquen la solució del problema i la fan més intel·ligible.

**8.1** Amb un array de 10 valors,

Inicialitzar-lo

Recorrent-lo emprant un índex imprimir la taula (valor, adreça)

Recorrent-lo emprant amb un apuntador imprimir la taula (valor, adreça)

**8.2** Exercici de treball amb pas de paràmetres amb estructures.

Crear una estructura molt simple com, per exemple, una estructura de dades personals amb només el camp dni. Sobre aquesta estructura inicialitzada, realitzar tres funcions (que canvien el dni) que es criden, consecutivament, per valor, per adreça i per referència, imprimint, després de cada crida, el resultat obtingut.

**8.3** Escriu una funció que imprimeixi les adreces i continguts d'un array

**8.4.** Realitzar un programa que, davant una entrada d'un string, en retorni un altre amb un espai i la lletra en majúscula després de cada punt i deixi només un espai a per tot. Recórrer el string emprant apuntadors.

## Exercici 8.1

L'exercici consta de tres parts:

Creació i inicialització de l'array

Recorregut de l'array per índex

Recorregut de l'array per apuntador

La codificació no té cap problema destacable.

### Codificació en C

```
/*
Recorrent un array per índex i per apuntadors
*/
#include <stdio.h>
void main()
{
    int i, a[10];           //Índex, array
    int *ptr;              //Apuntador a un enter

    for (i=0; i<10; i++) a[i] = i*i;           //Inicialitzar l'array
    printf("Índex\tAdreça\t\tValor\n");

    printf("Recorrent l'array per índex\n");           // Recorrent l'array per índex
    for (i=0; i<10; i++) printf("%i\t%i\t\t%i\n", i, &a[i], a[i]);

    printf("Recorrent l'array per apuntador\n");           // Recorrent l'array per apuntador
    ptr=a;
    for (i=0; i<10; i++) printf("%i\t%i\t\t%i\n", i, (ptr+i), *(ptr+i));
}
```

## Exercici 8.2

És un exercici de pràctica del pas de paràmetres en estructures.

Es crea una estructura molt simple i es comproven els resultats després de realitzar les diferents crides.

### Codificació en C

```
/* PAS DE PARAMETRES */

#include <stdio.h>
#define N 5

struct estudiant
{
    char *nom;
    long dni;
};

void CridaPerValor (estudiant);
void CridaPerVariable (estudiant *);
void CridaPerReferencia (estudiant &);

void main(void)
{
    estudiant est = {"Pere", 11111111};
    printf("->Inicial: nom=%s dni=%i\n", est.nom, est.dni);
    CridaPerValor(est);
    printf("->DespresCridaValor: nom=%s dni=%i\n", est.nom, est.dni);
    CridaPerVariable(&est);
    printf("->DespresCridaVariable: nom=%s dni=%i\n", est.nom, est.dni);
    CridaPerReferencia(est);
    printf("->DespresCridaReferencia: nom=%s dni=%i\n", est.nom, est.dni);
}

void CridaPerValor (estudiant e)
{
    e.dni = 22222222;
    e.nom="PereDos";
}

void CridaPerVariable (estudiant* e)
{
    e->dni = 33333333;
    e->nom="PereTres";
}

void CridaPerReferencia (estudiant &e)
{
    e.dni = 44444444;
    e.nom="PereQuatre";
}
```

### Exercici 8.3

L'exercici és directe.

#### Codificació:

```
/*  
Escriu una funció que imprimeixi les adreces i continguts d'un array  
*/  
  
#include <stdio.h>  
#define NUM 6  
  
void imprimir (int *);           //Procediment d'impressió  
  
void main()  
{  
    int A[]={2, 4, 6, 8, 10, 12};  
    imprimir(A);  
}  
  
void imprimir (int *p)         //definició procediment imprimir  
{  
    for (int i=0; i<NUM; i++)  
        printf("\t&A[%i]=%i, <A[%i]>=%2i\n", i, (p+i), i, *(p+i));  
}
```

## Exercici 8.4

L'algorisme fa ús de dues variables booleanes: punt i espai. Ambdues s'activen davant un espai o punt i actuen de forma adequada per a deixar els espais, punts i majúscules adequats en la frase.

Per tant, l'algorisme actua de la forma següent:

- i) punt i espai s'activen en iniciar el programa per a donar a entendre que la lletra ha de començar en majúscula.
- ii) quan es detecta un espai s'ha de controlar que no n'hi hagi cap més.
- iii) quan es detecta un punt s'ha de deixar només un espai després i la lletra ha de començar en majúscules.

Per tant, l'algorisme llegirà una frase i la copiarà en una altra amb les modificacions que calguin. Els caràcters es comprovaran per mitjà d'una instrucció case.

### Codificació:

```
/*
Pas a majúscula després de punt i espaiat correcta.
*/
#include <stdio.h>

const int NUM = 80;

void main()
{
    char frase[NUM], frase1[NUM];           //String vell i nou
    char *fr, *fr1;                          //Apuntadors a frase vella i nova
    int i=0, j=0, punt=1, espai=1;
    printf(" Entra una frase: ");
    gets(frase);
    fr = frase;
    fr1 = frase1;
    while ((*fr+i) != '\0') && (i<NUM)
    {
        switch(*(fr+i))
        {
            case '!': punt=1;                //Tractament punt
                *(fr1+j)='!';
                espai=0;
                break;
            case ' ': if (!espai)            //Tractament espai
                {
                    espai=1;
                    *(fr1+j) = ' ';
                }
                else j--;                    //S'eliminen caràcters (espais) redundants!
                break;
            default: if (punt)                //Tractament caràcter normal
                {
                    if (!espai)
                    {
                        *(fr1+j)=' ';
                        j++;
                    }
                    *(fr1+j) = *(fr+i) & 0xdf; //A majúscula
                }
        }
    }
}
```

```
        else *(fr1+j) = *(fr+i);
        punt=0;
        espai=0;
    }
    j++;
    i++;
}
*(fr1+j)='\0';
printf("Frase nova: %s\n", fr1);
}
```

```

/*
Pas a majúscula després de punt i espaiat correcta.
*/
#include <stdio.h>

const int NUM = 80;

void main()
{
    char frase[NUM], frase1[NUM];
    char *fr, *fr1;
    int i=0, j=0, punt=1, espai=1;
    printf(" Entra una frase: ");
    gets(frase);
    fr = frase;
    fr1 = frase1;
    while ((*fr+i) != '\0') && (i<NUM)
    {
        switch(*(fr+i))
        {
            case '.': punt=1;
                    *(fr1+j)='.';
                    espai=0;
                    break;
            case ' ': if (!espai)
                    {
                        espai=1;
                        *(fr1+j) = ' ';
                    }
                    else j--; //Més d'un espai: trec per que al final es posa!
                    break;
            default:
                    if (punt)
                    {
                        if (!espai)
                        {
                            *(fr1+j)=' ';
                            j++;
                        }
                        *(fr1+j) = *(fr+i) & 0xdf; //A majúscula
                    }
                    else *(fr1+j) = *(fr+i);
                    punt=0;
                    espai=0;
                }
            j++;
            i++;
        }
    }
    *(fr1+j)='\0';
    printf("Frase nova: %s\n", fr1);
}

```