

## 8. LABORATEGIA

### TAULA ZENBERIKOAK / BEKTOREAK

#### 1 HELBURUAK

Praktika honekin bilatzen dena:

- Taula zenberikoen erabilera hobetzea.
- Taula zenberikoekin funtzioak erabiltzea: parametroen igarotzea, funtzioetara deiak, ...

#### 2 MOTIBAZIOA

- Demagun tamaina eta baldintza bereko taula zenberiko ezberdinak kargatu nahi ditugula, ez da beharrezkoa kodea dauden taula kopurua adina aldiz errepikatzea, dagokion funtzioa deituko da parametro bezala taulan kargatu nahi dugun parametroa pasatuz.
- Berdina aurkezteko orduan, edo taulekin burutzen dugun beste edozein ataza errepikakorrek, gero programa beran edo beste batetan erabili daitekeelarik.

##### 2.1 Funtzioak C-n

Taula zenberikoak parametro bezala erabiltzeko orduan, hiru gauza hartu behar dira kontutan: funtzioaren deklarazioa, funtzioaren definizioa eta funtzioaren erabilera.

Ad.: N tamainuko taula bat zerora hasieratzen duen funtzioa.

- Deklarazioa:

```
void bektorea_hasieratu (int limite, int bec[]);
```

- Definizioa:

```
void bektorea_hasieratu (int limite, int bec[]);
{
    int i;

    for (i=0;i<limite;i++){
        bec[i]=0;
    }
}
```

- Erabilera:

```
void main()
{
    /*taula baten elementu kopurua */
    int const NELE=10;

    /*Zenbaki osoen taula baten deklarazio edo definizioa */
    int notak_ikas[NELE];
```

```
        bektorea_hasieratu (NELE, notak_ikas);  
        ...  
    }
```

### 3 ARIKETAK

#### 3.1 1. ariketa

##### 3.1.1 Helburua:

Ariketaren helburua taulekin funtzioak erabiltzea, elementu ezberdinak identifikatzea (deklarazioa, definizioa eta erabilera) eta funtzionamendua konprobatzea da.

##### 3.1.2 1. enuntziaturako laguntza:

Enuntziatua ebatzita ematen da beste ariketen gida legezt hartzeko.

```
#include <stdio.h>  
#include <stdlib.h>  
#define N 10  
  
int taulan_posizioa_bilatu(int zenb, int taula[]);  
  
void main()  
{  
    int taula [N],i=0,posizioa,zenb;  
    printf("\nEman 10 zenbaki taula batetan gordetzeko.\n");  
  
    for (i=0;i<N;i++)  
    {  
        printf("\n%d-garrena:",i);  
        scanf("%d",&zenb);  
        taula[i]=zenb;  
    };  
    printf("\nEman zenbaki bat");  
    printf(" eta zenbaki horren lehenengo posizioa taulan esango dizut:");  
    scanf("%d",&zenb);  
    posizioa=taulan_posizioa_bilatu(zenb,taula);  
    printf("\n %d zenbakia %d posizioan dago.\n",zenb,posizioa);  
  
    system("PAUSE");  
}  
  
int taulan_posizioa_bilatu(int zenb, int taula[])  
{  
    int aurkituta=0,i=0;  
    while((i<9)&&(aurkituta==0))  
    {  
        if (taula[i]==zenb) aurkituta=1;  
        i++;  
    }  
    if (aurkituta==0) return (-1);  
    else return (i-1);  
}
```

**3.1.3 Enuntziatua:**

N tamainako taula (ez derrigorrez ordenatuta) baten elementuak teklatutik irakurtzen dituen programa bat egin. Ondoren, erabiltzaileari zenbaki bat eskatuko zaio eta zenbaki horren lehenengo posizioa taulan kalkulatu du. Zenbakia bektorean agertzen ez bada, -1 bat aurkeztuko da.

Adibidea (demagun N 10 dela):

|    |    |    |   |    |   |    |    |    |    |  |
|----|----|----|---|----|---|----|----|----|----|--|
| i: | 0  | 1  | 2 | 3  | 4 | 5  | 6  | 7  | 8  |  |
| B: | 12 | 14 | 3 | 11 | 7 | 17 | -4 | 23 | 31 |  |

Eman zenbaki bat: 17

17 zenbakia 5, posizioan dago

**3.2 2. ariketa****3.2.1 Helburua:**

2. ariketaren helburua aurreko ariketan ikasitako ezaguerak praktikan jartzea da.

**3.2.2 2. enuntziaturako laguntza:**

1. ariketa erabili.

**3.2.3 Enuntziatua:**

Aurreko ariketan bezala, baina bi ezberdintasun nagusiekin:

- Taula ez betetzea posible izango da, erabiltzaileak zenbaki negatibo bat teklatzen badu balio gehiago irakurtzen ez delako jarraituko.
- Erabiltzaileak elementua handienetik txikienera ordenatuta sartuko ditu (egiaztapena ez da beharrezkoa).

**3.3 3. ariketa****3.3.1 Helburua:**

Funtzio edo funtzio baten interfasea emanda, hau inplementatu eta erabiltzen duen programa bat idatzi.

**3.3.2 3. enuntziaturako laguntza:**

Programa hau burutzeko hurrengo funtzio interfasea erabili:

```
void taulan_satu(int posizioa, int dimensioa, int taula[])
```

non posizioa elementurik txikiena bilatzeko hartuko den lehenengo posizioa delarik eta behin aurkituta posizio horretan lagako da elementu guztiak eskuinerantz mugitu ondoren. Dimensioa taulan sartutako elementu kopurua da eta taula, modifikatuko den taula da. Funtzio honek x posiziotik azkenengorarte bilatuko du elementurik txikiena eta elementu guztiak eskuinerantz mugitu ondoren, elementurik txikiena x posizioan sartuko du.

### 3.3.3 Enuntziatua:

Zenbakiak eskatu eta N dimensioko taula baten sartzen dituen programa bat egin (demagun ariketa honetan N 10 dela); elementu sarrera erabiltzaileak -1 sartu edo taula betetzen denean amaitzen da. (KONTU!! Inoiz ez da guztiz betetzea utzi behar, sarketarako behintzat leku bat geratu behar da) Programak sasrtutako zenbaki kopurua zenbatzen joan beharko du (-1 kontutan hartu gabe) eta taularen x posizioa (erabiltzaileak adierazita) aldatu beharko du, x posiziotik aurrera aurkitutako zenbakirik txikiena sartuz. Txikiena xx posizioan sartu baino lehen beste elementu guztiak eskuinarantz mugituko ditu.

Adibidea:

EMAN GEHIENEZ 10 ZENBAKI (NEGATIBOA AMAITZEKO):

12 14 3 11 7 17 -1

|    |    |    |   |    |   |    |   |   |   |   |
|----|----|----|---|----|---|----|---|---|---|---|
| I: | 0  | 1  | 2 | 3  | 4 | 5  | 6 | 7 | 8 | 9 |
| B: | 12 | 14 | 3 | 11 | 7 | 17 |   |   |   |   |

EMAN X POSIZIOA: 3

Behin aldaketa eginda, arraya honela geratzen da:

12 14 3 **7** 11 7 17

## 3.4 4. ariketa

### 3.4.1 Helburua:

4. ariketaren helburua funtzio baten interfasea sortzea da eta gainera hau inplementatu eta erabiltzen duen programa ere burutzea.

### 3.4.2 4. enuntziaturako laguntza:

Programa hau burutzeko, kapikua\_da funtzioa erabili, zeinek taularen edukia kapikua den ala ez esaten duena.

### 3.4.3 Enuntziatua:

Gehienez N digitoko zenbaki osoa emanda (demagun N 5 dela), kapikua den ala ez erabakitzen duen programa idatzi.

Adibidea:

GEHIENEZ 5 DIGITOKO ZENBAKI BAT SARTU: 1231

1231 zenbakia ez da kapikua.