

**Programação Imperativa – Folha das aulas práticas nº 04 – Uma Resolução**

**Temas:** INTRODUÇÃO AOS VECTORES

Nome \_\_\_\_\_ Ncd \_\_\_\_\_

**Algumas notas sobre vectores...** Um vector de 20 inteiros com o nome *v* é *definido* da seguinte forma:

```
int v[20];
```

Os 20 elementos são  $v[0], v[1], \dots, v[19]$ . Colocar o dobro do valor de  $v[2]$  em  $v[5]$ :  $v[5]=2*v[2]$ ; escrever  $v[2]$ : `printf("v[2]=%d\n",v[2]);`.

Um vector pode ser definido e *inicializado*: `int w[5]={4,6,8,10,1};`.

Um índice de um vector pode ser qualquer expressão *E* com valor inteiro compreendido entre 0 e o número de elementos do vector - 1; assim acontece, por exemplo, na atribuição  $v[E]=F$  em que *F* tem o mesmo tipo que *v*. Evidentemente há também vectores de `char` de `float`, de `double`,... evidentemente, o nome de um vector (como todos os nomes definidos pelo utilizador) pode ser qualquer sequência de letras ou dígitos ou “.” começada por uma letra (com e excepção das palavras reservadas). Exemplo:

```
#define MAXTEN 10000
double tensoes_simetricas_superiores_22 [MAXTEN];
```

- 
1. O vector *v* foi definido com `int v[20];`. Diga qual é o o valor final dos elementos de *v* que estão definidos após a execução das seguintes instruções

```
int v[20];      /* Define v[] com 20 posicoes */
int i=3;        /* i <- 3 */
v[i]=2*i;       /* v[3] <- 6 */
v[v[i]]=1;      /* v[6] <- 1 */
v[i]*=5;        /* v[3] <- v[3] * 5 */
```

```
v[3] = 30;
v[6] = 1;
```

- 
2. Escreva um curto programa que coloque 20 em  $v[0]$ , 19 em  $v[1]$ ,..., 1 em  $v[19]$ .

```
#include <stdio.h>
main() {
    int v[20], i;
    for(i=0; i<20; i++)
        v[i] = 20 - i;
}
```

3. Considere o seguinte programa

```
#define MAX 10

main(){
    int s=0, i, v[MAX]={2,5,10,8,1,3,9,2,12,1};
    for(i=0;i<MAX;i++)
        if(v[i]>5) s++;
    printf("%d\n",s);
}
```

Diga o que escreve este programa no caso geral, isto é, para um vector  $v$  qualquer (com  $MAX$  elementos).

O programa conta o número de elementos do vector  $v[]$  que são maiores do que 5 (neste exemplo:  $v[2]$ ,  $v[3]$ ,  $v[6]$  e  $v[8]$ , imprimindo '4'). No caso geral, este programa imprime

$$\#\{i \mid i \in [0, MAX - 1], v[i] > 5\}$$

---

4. Escreva um programa que conte e imprima a soma dos elementos de um vector (deve defini-lo e inicializá-lo) que não excedam 2.

```
#include <stdio.h>

#define MAX 10

main() {
    int s=0, soma=0, i, v[MAX]={2,5,10,8,1,3,9,2,12,1};
    for(i=0; i<MAX; i++)
        if(v[i]<=2) {
            s++;
            soma += v[i];
        }
    printf("%d\n", soma);
}
```

---

5. Considere o seguinte programa

```
#define MAX 10

main(){

    int i, s[MAX]={2,5,10,8,1,3,9,2,12,1};

    for(i=1;i<MAX;i++)
        s[i]=s[i]+s[i-1];

    for(i=0;i<MAX;i++)
        printf("%3d\n",s[i]);
}
```

Indique o conteúdo dos elementos do vector após a execução do código indicado.

Conteúdo no final:

$s[0]=2$ ,  $s[1]=7$ ,  $s[2]=17$ ,  $s[3]=25$ ,  $s[4]=26$ ,  
 $s[5]=29$ ,  $s[6]=38$ ,  $s[7]=40$ ,  $s[8]=52$ ,  $s[9]=53$ .

6. Considere o seguinte programa

```
int lala(int x, int n,int v[]){
    int i;
    v[n]=x;
    for(i=0;i<=n;i++)
        if(v[i]==x)
            return(i!=n);
}

main(){
    int v[6]={2,3,4,5,6};
    printf("%d %d %d\n",lala(2,6,v),lala(1,6,v),lala(6,6,v));
}
```

O que escreve o programa? No caso geral, qual o valor retornado pela função `lala`?

Output: '1 0 1'

No caso geral, a função `lala(x,n,v[])` retorna 1 se existe um elemento  $v[i]$  tal que  $v[i] == x$  e  $i < n$ ; retorna 0 caso contrário.

---

7. Considere o seguinte programa

```
int lele(int n,int v[],int m,int w[]){
    int i,j,s=0;
    for(i=0;i<n;i++)
        for(j=0;j<m;j++)
            if(v[i]==w[j]){
                s++;
                break;
            }
    return s;
}

main(){
    int v[5]={2,3,4,5,1}, r[4]={10,2,8,1};
    printf("%d\n",lele(5,v,4,r));
}
```

O que escreve o programa? No caso geral, qual o valor retornado pela função `lala`?

Output: '2'

No caso geral, a função `lele(n,v[],m,w[])` retorna o número de elementos de  $v[]$  que existem em  $w[]$ .