

Curso Profissional: Programador/a de Informática
PSD – 10.º ano: UFCD 0810 - Programação em C/C++ - avançada
Ficha de Trabalho 6

Ano letivo 21/22

Vetores e apontadores

- O nome de um vetor é igual ao endereço do 1º elemento (o seu menor endereço), isto é, se v for um vetor $v == \&v[0]$;
- Se o nome de um vetor é um endereço então o nome de um vetor é um apontador para o 1º elemento desse vetor (no entanto, não pode ser alterado durante a execução do programa a que pertence, apontando sempre para o 1.º elemento);
- Tendo em conta **a)** e **b)** concluímos que se i é o i -ésimo elemento do vetor v então:
 - $\&v[i]$ é equivalente a $v + i$, para obter o endereço do i -ésimo elemento;
 - $v[i]$ é equivalente a $*(v + i)$, para obter o conteúdo armazenado na i -ésima posição.

Apontadores para vetores

Suponhamos que tínhamos as seguintes declarações:

```
int v[3]={10,20,30};
```

```
int *ptr;
```

para colocar o ponteiro ptr a apontar para o 1.º elemento de v fazemos:

```
ptr=&v[0]; ou ptr=v;
```

```
printf ("%d %d\n",v[0],*ptr); /* output: 10 10 */
```

Modos de aceder ao valor 30

$v[2]$	Valor existente na posição de índice 2 do vetor
$*(ptr+2)$	ptr contém o endereço do 1º valor, se adicionarmos 2, obtemos o endereço do valor 30. Para obter o valor basta usar o operador $*$ (valor apontado por)
$*(v+2)$	Se $v == \&v[0]$ usa-se a mesma estratégia do exemplo anterior
$ptr[2]$	O endereçamento de elementos através de parêntesis retos pode ser realizado também por apontadores, como se de um vetor se tratasse

EXERCÍCIOS:

1- Supõe o seguinte esquema que corresponde à seguinte declaração

```
char s[]="Maria", *ptr=&s[0]; /* podia ser *ptr=s;
```

s						ptr				
M	a	r	i	a	\0			100		
100	101	102	103	104	105	500	501	...

Completa o seguinte quadro colocando as expressões que dão como resultado o conteúdo da coluna valor

Usando s	Usando ptr	Valor
		'r'
		100
		101
		500

2- Escreve um programa que mostre uma string no ecrã pela ordem que foi introduzida e pela ordem contrária. Deves usar um apontador para a string para percorrer a mesma.

Apontadores de apontadores

Uma vez que os apontadores ocupam espaço em memória, é possível obter a sua posição através do operador &

Sintaxe:

tipo **pr;

Sendo:

ptr o nome da variável do tipo apontador de apontador

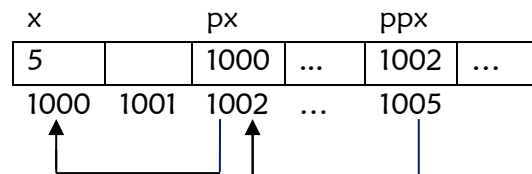
tipo* o tipo de variável para a qual apontará (um apontador)

***** o operador conteúdo indica que ptr é uma variável do tipo apontador

(...e assim sucessivamente, sem qualquer limitação de asteriscos)

Exemplificação (resumo):

a)



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

```
main()
```

```
{
```

```
int x=5, *px=&x, **ppx=&px; /*px aponta para x e ppx aponta para px*/
```

```
printf("x= %d &x= %d\n", x, &x);
```

```
printf("x= %d &x= %d\n", *px, px);
```

```
printf("x= %d &x= %d\n", **ppx, *ppx);
```

Expressão	Tipo	Valor	Descrição
x	int	5	Valor de x
px	int*	1000	Endereço de x
*px	int	5	Valor apontado por px
ppx	int**	1002	Endereço de px
ppx	int	1000	Valor apontado por ppx
**ppx	int	5	Valor apontado pelo endereço apontado por ppx