

MC-102 — Aula 15

Vetores

Instituto de Computação – Unicamp

Segundo Semestre de 2011

Roteiro

- 1 Introdução
- 2 Vetores
- 3 Exemplos
- 4 Vetor em função
- 5 Exercício

Como armazenar 3 notas?

```
float nota1, nota2, nota3;  
  
printf("Nota do aluno 1: ");  
scanf("%f", &nota1);  
printf("Nota do aluno 2: ");  
scanf("%f", &nota2);  
printf("Nota do aluno 3: ");  
scanf("%f", &nota3);
```

Como armazenar 100 notas?

```
float nota1, nota2, nota3, /* .... */ nota100;  
  
printf("Nota do aluno 1: ");  
scanf("%f", &nota1);  
printf("Nota do aluno 2: ");  
scanf("%f", &nota2);  
  
/* ... */  
  
printf("Nota do aluno 100: ");  
scanf("%f", &nota100);
```

Como armazenar n (≤ 100) notas?

```
float nota1, nota2, nota3, /* .... */ nota100;

if (n >= 1)  { printf("Nota do aluno 1: ");
               scanf("%f", &nota1); }
if (n >= 2)  { printf("Nota do aluno 2: ");
               scanf("%f", &nota2); }

/* ... */

if (n >= 100) { printf("Nota do aluno 100: ");
                 scanf("%f", &nota100); }
```

Vetores — Definição

Coleção de variáveis do mesmo tipo referenciada por um nome comum.

(Herbert Schildt)

- acesso por meio de índice
- posições contíguas na memória
- tamanho pré-definido
- índices fora dos limites podem causar comportamento anômalo do código

Declaração de um vetor

`<tipo> identificador [<número de posições>];`

- A primeira posição de um vetor tem índice 0.
- A última posição de um vetor tem índice *<número de posições>* - 1.

Exemplo

```
float notas[100];
```

Usando um vetor

```
a = identificador [<posição>];
```

- Pode-se substituir uma variável de um determinado tipo por **um único** elemento de um determinado vetor.
- Este elemento se comporta como uma variável: retorna o seu valor como uma expressão e pode ter valores atribuídos.

Exemplo

```
a = nota[10];  
nota[5] = 9.5;
```

Vetores

- Na memória: (Ex. `int d; int vetor[5]; int f;`)

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-

Vetores

- Ao executar `vetor[3]=10;`:

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-
					10		

Vetores

- O que ocorre se digitar os comandos:
`vetor[5]=5;`
`vetor[-1]=1;`

Vetores

- Ao executar
`vetor[3]=10;`
`vetor[5]=5;`
`vetor[-1]=1;`

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-
	1				10		5

Questões importantes sobre vetores

- O tamanho do vetor é pré-definido. (Ou seja, após a compilação o tamanho não pode ser mudado)
- Índices fora dos limites podem causar comportamento anômalo do código.
- Veja exemplo em `limites.c`

Como armazenar n (≤ 100) notas?

```
float nota[100];  
int n, i;  
  
printf("Número de alunos: ");  
scanf("%d", &n);  
  
for (i = 0; i < n; i++) {  
    printf("Nota do aluno %d: ", i+1);  
    scanf("%f", &nota[i]);  
}
```

- Veja o código: notas.c

Polinômios

```
float coef[26];  
int grau;  
  
for (i = grau; i >= 0; i--) {  
    printf("coeficiente de x^%d: ", i);  
    scanf("%f", &coef[i]);  
}
```

- Veja o código: poli.c

Derivada de um polinômio

```
float coef[26];  
int grau;  
  
for (i = 1; i <= grau; i++)  
    coef[i-1] = coef[i]* i;  
  
grau--;
```

- Veja o código: deriv.c

Cadeia de caracteres (string)

Vetor de caracteres terminado pelo caracter '`\0`'.

```
char str[30];  
printf("Digite uma cadeia de caracteres: ");  
scanf("%s", str);  
for (i = 0; str[i] != '\0'; i++)  
    printf("%c", str[i]);  
printf("\n");
```

- Veja o código: `str.c`

Busca por um elemento

```
for (i = 0; str[i] != c && str[i] != '\0'; i++);  
  
if (str[i] == c)  
    printf("%c está presente em %s\n", c, str);  
else  
    printf("%c não está presente em %s\n", c, str);
```

- Veja o código: busca.c

Balanceamento de parênteses

Como verificar se uma cadeia do tipo

`()()()()()())())((()())()())()()`

está balanceada?

- Veja o código: `balanc.c`

Vetor em função

- Vetores têm um comportamento diferente quando usados como parâmetros ou valores de retorno de funções.
- Por padrão, ao se indicar o tipo de um vetor, este sempre é interpretado pelo compilador como o endereço do primeiro elemento do vetor.
- Ao passar um vetor como parâmetro não é necessário fornecer o seu tamanho na declaração da função. Porém, é importante lembrar que o vetor tem um tamanho que deve ser considerado.
- Desta forma, sem precisarmos usar uma notação especial, os vetores são sempre passados por referência. Veja exemplos em `vetor parametro.c` e `vetor vs variavel.c`.

Vetor em função

- Devemos ficar atentos às implicações do fato dos vetores serem sempre passados por referência.
- Ao passar um vetor como parâmetro, se ele for alterado dentro da função, as alterações ocorrerão no próprio vetor e não em uma cópia.
- Ao retornar um vetor como valor de retorno, não é feita uma cópia deste vetor. Assim, o vetor retornado pode desaparecer se ele foi declarado no corpo da função.

Vetor em função

Faça um programa que receba dois vetores x e y de números reais, de tamanho n ($n \leq 20$), onde n é fornecido pelo usuário. A saída do programa deve ser o produto entre x e y , definido como

$$\sum_{i=0}^{n-1} x[i].y[i]$$