



INSTITUTO DE
INFORMÁTICA
UFG

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C
Inicialização

Vetores

Thierson Couto Rosa

Instituto de Informática
Universidade Federal de Goiás
Introdução à Programação

Primeiro Semestre de 2017



Problemas com Várias Variáveis de Mesma Natureza

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C

Inicialização

Notas Moires que a Média

Dada uma turma com 50 alunos, deseja-se computar a média das notas da turma em uma prova e deseja-se imprimir a matrícula dos alunos que tiveram nota superior à média até o momento.

Soluções com os recursos estudados

- Declarar 50 variáveis para armazenar as notas. Computar a média, usar 50 comandos de seleção para comparar cada variável com a média.



Problemas com Várias Variáveis de Mesma Natureza

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C

Inicialização

Notas Moires que a Média

Dada uma turma com 50 alunos, deseja-se computar a média das notas da turma em uma prova e deseja-se imprimir a matrícula dos alunos que tiveram nota superior à média até o momento.

Soluções com os recursos estudados

- Declarar 50 variáveis para armazenar as notas. Computar a média, usar 50 comandos de seleção para comparar cada variável com a média.
- Declarar uma variável nota. Ler cada nota e acumular seu valor e computar a média. Ler novamente todas as 50 notas comparando com a média.



Sequência de Várias de Mesmo Tipo

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores

na
Linguagem C

Inicialização

Melhoria na Primeira Solução

- A primeira solução anterior é a melhor do ponto de vista do usuário do programa, mas não do programador.
- Ela seria a ideal se pudéssemos declarar as 50 variáveis de outra forma.
- O ideal seria declarar uma sequência de 50 espaços do tipo float na memória, usando um único nome (por exemplo *notas*).
- A sintaxe da linguagem deveria permitir ter a acesso individual a cada um dos 50 espaços tipo-float através de um índice. (ex. `notas[i]`)



Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C
Inicialização

Variáveis Compostas Homogêneas Unidimensionais

- Um vetor (ou *array*) é uma coleção com n variáveis de um mesmo tipo (composição homogênea), que compartilham o mesmo nome e que ocupam posições consecutivas de memória.
- Cada uma variável-componente de um vetor denomina-se elemento e é identificada por um índice.
- Na linguagem C, se v é um vetor com n elementos, seu i -ésimo elemento é acessado através da expressão: $v[i]$, onde $0 \leq i < n$.



Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C

Inicialização

Exemplo

$valor_1$	$valor_2$	$valor_3$	\dots	$valor_n$
$v[0]$	$v[1]$	$v[2]$	\dots	$v[n - 1]$

- Cada $(valor_i)$, $0 \leq i < n$ é do mesmo tipo.
- O primeiro índice é zero. O último, é $n - 1$.
- Tentar acessar um elemento fora desses índices causa um erro de invasão de memória.



Declaração de vetores

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C
Inicialização

Na linguagem C

- De acordo com a definição de vetor, quais informações o compilador precisa saber para reservar um vetor na memória?



Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C
Inicialização

Na linguagem C

- De acordo com a definição de vetor, quais informações o compilador precisa saber para reservar um vetor na memória?
 - O nome do vetor.



Declaração de vetores

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C
Inicialização

Na linguagem C

- De acordo com a definição de vetor, quais informações o compilador precisa saber para reservar um vetor na memória?
 - O nome do vetor.
 - O tipo de dados comum a cada elemento do vetor.



Declaração de vetores

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C
Inicialização

Na linguagem C

- De acordo com a definição de vetor, quais informações o compilador precisa saber para reservar um vetor na memória?
 - O nome do vetor.
 - O tipo de dados comum a cada elemento do vetor.
 - O número de elementos do vetor



Formato Geral

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores

na
Linguagem C

Inicialização

```
tipo nome_vetor [tamanho];
```

- tipo pode ser qualquer tipo da linguagem (int, float, etc), ou pode ser um tipo criado pelo programador (visto mais tarde no curso)
- Cada elemento i do vetor é tratado com uma variável comum do tipo definido na declaração (mesmas operações), mas em vez de ter um nome, é acessada por: `nome_vetor[i]`



INSTITUTO DE
INFORMÁTICA
UFG

Problema Usado na Motivação

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C
Inicialização

Esboço da solução com vetores



Problema Usado na Motivação

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores

na
Linguagem C

Inicialização

Esboço da solução com vetores

- Declaração de dois vetores com 50 elementos: um vetor de int (matriculas) e um vetor de float (notas da prova).
- Uso de uma variável inteira para servir de índice nos dois vetores.
- Leitura do vetor de matrículas e do vetor de notas.
- Percorrer o vetor de notas, acumulando o valor das notas.
- Computar a média das notas
- Percorrer o vetor de notas testando para cada elemento i , se ele é maior ou igual á media. Se for, imprimir a matrícula e a nota do aluno i .



Solução do Problema Usando Vetores

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação

de Vetores

na
Linguagem C

Inicialização

```
1 #include <stdio.h>
2 #define TAM_VET 5
3 int main(){
4     int matricula[TAM_VET], i=0;
5     float nota[TAM_VET], media=0.0;
6     while (i < TAM_VET) {
7         //Ler a matricula e a nota do aluno i
8         scanf("%d %f",&matricula[i], &nota[i]);
9         media+=nota[i];
10        i++;
11    }
12    media /= (TAM_VET * 1.0);
13    printf("Alunos com nota acima da media (%.2f):\n",media);
14    //Verificar quais alunos superam a media
15    i=0;
16    while (i < TAM_VET) {
17        if (nota[i] >= media) {
18            printf("Matricula: %d Nota: %.2f\n", matricula[i],
19                nota[i]);
20        }
21        i++;
22    }
23 }
```



Inicializando Vetores

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C

Inicialização

É possível inicializar todos os elementos de um vetor durante sua declaração, com a seguinte sintaxe:

```
tipo nome_vet[n]= {valor0, valor1, ... valor(n-1) };
```

Exemplo

- **char** vogal[5]={ 'a', 'e', 'i', 'o', 'u' } em vez de
- **char** vogal[5]; vogal[0]='a'; vogal[1]='e'; vogal[2]='i' ; vogal[3]='o'; vogal[4]='u';



Lembretes importantes

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores

na
Linguagem C

Inicialização

- Enquanto os elementos do vetor não receberem valores no programa, seus valores são imprevisíveis (alguns compiladores inicializam os elementos com zero, mas não são todos).
- O primeiro elemento de um vetor tem índice zero e o último tem como índice um valor a menos que o tamanho do vetor.
- Erro comum:
 - a) Declaração: **char** vogal[5];
 - b) acesso: `printf ("%c", vogal[5]);`



Implementação de Vetores na Linguagem C

Implementação Interna

- Em C os vetores de um tipo T são implementados usando-se um ponteiro para o tipo.

- A declaração:

```
1 int vet[10]={9,1,2,5,0,30,10,11,12};
```

- Faz com que internamente o compilador:
 - a) declare *vet* como um ponteiro para int: **int*** *vet*;
 - b) aboque um espaço contíguo para 10 variáveis int;
 - c) armazene o endereço do primeiro byte que compõe o primeiro int na variável *vet*;



INSTITUTO DE
INFORMÁTICA
UFG

Vetores e Ponteiros

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

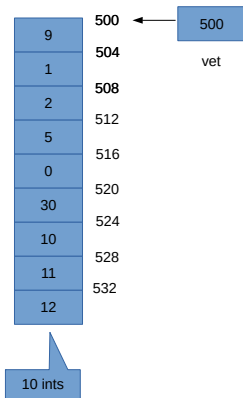
Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C

Inicialização

Ilustração do Exemplo Anterior





Recordando Expressões com Ponteiros

Operações Aritméticas

- São permitidas apenas duas operações aritméticas com ponteiros: adição e subtração. Os operadores `--` e `++` também podem ser utilizados.
- Cuidado: as operações aritméticas com ponteiros têm um significado diferente dos tradicionais. Se uma variável p é um ponteiro para inteiro, então, $p + 1$ não soma 1 no endereço armazenado em p , mas sim 4.
- Uma soma de um valor x a uma variável ponteiro faz com que a variável passe a apontar para o x -ésimo elemento daquele tipo, a partir do endereço que estava armazenado inicialmente na variável.



Vetores e Ponteiros

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores
na
Linguagem C

Inicialização

Acesso a elementos de um vetor

- Se vetores são implementados como ponteiros, então há duas formas de acesso aos elementos de um vetor:
 - indexação;
 - aritmética de ponteiros.
- A expressão $*(vet + 4)$ acessa o quinto elemento do vetor (valor zero) e é equivalente a $vet[4]$.
- Podemos rescrever o programa anterior que computa a média de modos que todos os acessos aos elementos do vetor sejam via ponteiros.



Solução do Problema Inicial Usando Ponteiro para Acesso

Vetores

Thierson
Couto Rosa

Motivação

Definição

Declaração

Exemplo

Inicialização

Lembretes

Implementação
de Vetores

na
Linguagem C

Inicialização

```
1 #include <stdio.h>
2 #define TAM_VET 5
3 int main(){
4     int matricula[TAM_VET], i=0;
5     float nota[TAM_VET], media=0.0;
6     while (i < TAM_VET) {
7         //Ler a matricula e a nota do aluno i
8         scanf("%d %f",matricula+i, nota+i); //Nao usa "&" na leitura
9         //!!!
10        media += *(nota+i);
11        i++;
12    }
13    media /= (TAM_VET*1.0);
14    printf("Alunos com nota acima da media (%.2f):\n",media);
15    //Verificar quais alunos superam a media
16    i=0;
17    while (i < TAM_VET) {
18        if ( *(nota+i) >= media) {
19            printf("Matricula: %d Nota: %.2f\n", *(matricula+i),
20                *(nota+i));
21        }
22        i++;
23    }
24    return(0);
25 }
```



Passagem de Vetor como Parâmetro de Função

Declaração do Parâmetro Vetor

- Declara-se um parâmetro da função cujo tipo é um ponteiro para o tipo que forma o vetor.
- Três formas de declarações são equivalentes:
 - $f(\text{Tipo } v[], \dots)$
 - $f(\text{Tipo } v[n], \dots)$, onde n é uma constante inteira;
 - $f(\text{Tipo}^* v, \dots)$
- Os três tipos de declaração são equivalentes e correspondem a um ponteiro para "Tipo".



Passagem de Vetor como Parâmetro de Função

Chamada de Função Passando Vetor Como Parâmetro

- Passa-se apenas o nome da variável do tipo ponteiro que corresponde ao vetor declarado. Por exemplo:

```
1 int f(tipo v[], ...){  
2     ...  
3 }  
4  
5 int main(){  
6     tipo vet[40];  
7     ...  
8     f(vet, ...);  
9  
10 }
```

- o endereço de início dos 40 elementos de vet para o parâmetro v durante a chamada.



Passagem de Vetor como Parâmetro de Função

Informações Adicionais

- Importante passar o tamanho do vetor como parâmetro, para qua a função tenha como percorrer o vetor.
- A implementação de vetores como um ponteiros para uma área contígua com vários elementos de um mesmo tipo, permite uma rápida passagem do vetor como parâmetro:
 - os elementos do vetor não precisam ser copiados para a função;
 - apenas o endereço do vetor é copiado, evitando várias cópias na memória.
- Porém há desvantagens: uma alteração de um elemento do vetor pela função altera diretamente o elemento do vetor passado como parâmetro, pois na verdade a função opera diretamente no vetor através do ponteiro passado a ela.



Passagem de Vetor como Parâmetro de Função

Solução para o Problema Inicial Usando Funções

```
1 float LeMatriculasENotas(int m[], int tam , float n[] ){
2     int i=0; float media=0.0;
3     for(i=0;i < TAM_VET;i++) {
4         scanf("%d %f",&m[i], &n[i]);
5         media += n[i];
6     }
7     return(media/n)
8 }
9 int main(){
10     int matricula[TAM_VET], i=0;
11     float nota[TAM_VET], media=;
12     media =LeMatriculasENotas(matricula, TAM_VET, nota);
13     for(i=0;i < TAM_VET;i++) {
14         if (nota[i] > media) {
15             printf("Matricula: %d Nota: %.2f\n", matricula[i],
16                 nota[i]);
17         }
18     }
19     return(0);
20 }
```