



Estrutura de Dados

Diógenes Carvalho Matias

Introdução



Linguagem C

Laços em C:

É muito comum, em programas computacionais, termos procedimentos iterativos. ou seja procedimentos que devem ser executados em vários passos. Vamos analisar como exemplo abaixo o cálculo do valor do fatorial de um número inteiro não negativo.

$$z! = z \times (z - 1) \times (z - 2) \dots 3 \times 2 \times 1, \text{ onde } 0! = 1$$

Introdução

Laços em C

Mas antes disso vamos ver que a linguagem C nos ajuda nesta interação por isso podemos fazer essas interações através de laços de repetição.

Laço de repetição: while

Exemplo:

```
int i = 1;
while (i <= 10) {
    printf("%d", i++);
}
```

ou

```
int i = 10;
while (i <= 10) {
    printf("%d", i--);
}
```

Estrutura de Dados

Laços em C

Vamos ver a solução do exemplo do fatorial com o laço While:

```
#include <stdio.h>

int main (void)
{
    int i;
    int n;
    int fatorial = 1;
    printf("Digite um número inteiro nao negativo:");
    scanf("%d", &n);
        // calcula fatorial
    i = 1;
    while (i <= n)
    {
        fatorial *= i;
        i++;
    }
    printf("O Fatorial = %d \n", fatorial);
    return 0;
}
```

Estrutura de Dados

Laços em C

Laço de repetição: Do

```
int i = 1;  
do {  
    printf("%d", i++);  
} while (i <= 10);
```

Estrutura de Dados

Laços em C

Laço de repetição: Do

```
int i = 1;  
do {  
    printf("%d", i++);  
} while (i <= 10);
```

Obs: com base nesta estrutura faça a adaptação do algoritmo anterior utilizando o laço do While.

Estrutura de Dados

Laços em C

Laço de repetição: For

Exemplo:

```
for (i=1;i<=10;i++){  
    printf("\n %d", i);  
}
```



Estrutura de Dados

Laços em C

Como ficaria o algoritmo do Fatorial já mostrado utilizando o For?

Estrutura de Dados

Laços em C

```
#include <stdio.h>
int main (void)
{
    int i;
    int n;
    int fatorial = 1;
    printf("Digite um número inteiro nao negativo:");
    scanf("%d", &n);

    // calculo do fatorial
    for (i = 1; i <= n; i++)
    {
        fatorial *= i;
    }
    printf("O Fatorial = %d \n", fatorial);
    return 0;
}
```



Estrutura de Dados

Laços em C

Como exemplo podemos incrementar o algoritmo acima já analisado criando uma função em C e mostrando seu fatorial:



Estrutura de Dados

Laços em C

```
#include <stdio.h>
int fat (int n);
int main (void)
{
    int n = 0;
    int r;
    printf("Digite um número inteiro nao negativo");
    scanf("%d",&n);
    r = fat ( n );
    printf("Fatorial de %d = %d \n", n, r);
    return 0;
}

//criando uma função de para calcular o fatorial
int fat (int n)
{
    int fatorial = 1.0;
    while (n != 0)
    {
        fatorial *= n;
        n--;
    }
    return fatorial;
}
```



Estrutura de Dados

Vamos falar de vetores em C

O vetor é uma estrutura de dados linear que necessita de somente um índice para que seus elementos sejam endereçados. É utilizado para armazenar uma lista de valores do mesmo tipo, ou seja, o tipo vetor permite armazenar mais de um valor em uma mesma variável.

Estrutura de Dados

Vamos falar de vetores em C

O vetor é uma estrutura de dados linear que necessita de somente um índice para que seus elementos sejam endereçados. É utilizado para armazenar uma lista de valores do mesmo tipo, ou seja, o tipo vetor permite armazenar mais de um valor em uma mesma variável.

Exemplo de vetores:

```
int vetorX[10]; //Vetor de inteiros
```

obs: vetorX[0] é o primeiro elemento e vetorX[9] o último.

Estrutura de Dados

Vamos falar de vetores em C

Um outros exemplos:

```
int x[5];
```

```
    x[0]=12;
```

```
    x[1]=13;
```

```
    x[2]=34;
```

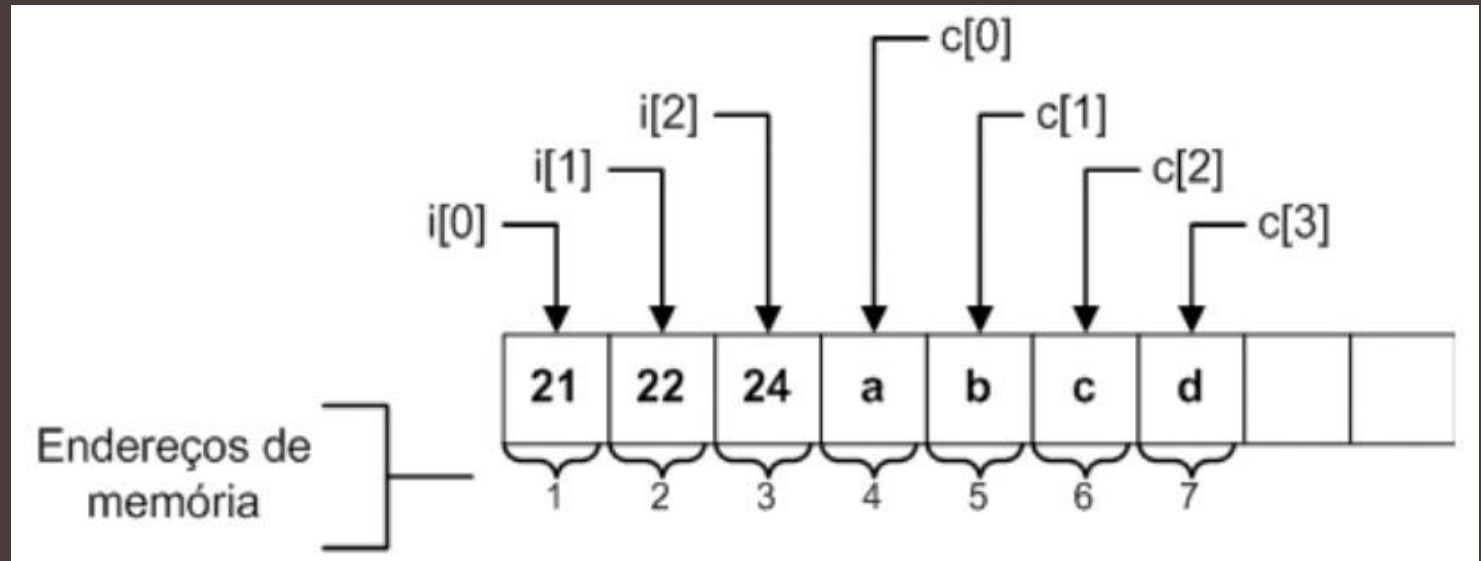
```
    x[3]=72;
```

```
    x[4]=01;
```

Estrutura de Dados

Vamos falar de vetores em C

Um exemplo o que esta acontecendo na memoria do computador:





Estrutura de Dados

Vamos falar de vetores em C

Algumas observações:

características de um vetor:

- Alocação estática (deve-se conhecer as dimensões da estrutura no momento da declaração em C)
- Estrutura homogênea
- Alocação seqüencial (bytes contíguos)
- Inserção/Exclusão
 - Realocação dos elementos;
 - Posição de memória não liberada;

Vamos ver como funciona um vetor simples:

Estrutura de Dados

Vamos falar de vetores em C

Exemplo pratico:

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

```
#include <stdlib.h>
```

```
int main(){
```

```
    float notas[2];
```

```
    printf("Digite a primeira nota:");
```

```
    scanf("%f",&notas[0]);
```

```
    printf("Digite a segunda nota:");
```

```
    scanf("%f",&notas[1]);
```

```
    printf("O aluno tirou as notas %.1f e %.1f.",notas[0],notas[1]);
```

```
    return 0;
```

```
}
```

Estrutura de Dados

Vamos falar de vetores em C

Algumas observações:

características de um vetor:

- Alocação estática (deve-se conhecer as dimensões da estrutura no momento da declaração em C)
- Estrutura homogênea
- Alocação seqüencial (bytes contíguos)
- Inserção/Exclusão
 - Realocação dos elementos;
 - Posição de memória não liberada;



Estrutura de Dados

Vetores em C

Exercicio 01:

Faça um algoritmo que receba valores inteiros de em um vetor de tamanho 3x2 e preencha um vetor inteiro de tamanho 6. Imprima o vetor preenchido.

Estrutura de Dados

Vetores em C

Exercicio 01:

Faça um algoritmo que receba valores inteiros de em um vetor de tamanho 3x2 e preencha um vetor inteiro de tamanho 6. Imprima o vetor preenchido.

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

int main(){

    int vetor[6], i;

    for (i = 0; i < 6; i++){

        scanf("%d",&vetor[i]);
        printf("%d \n",vetor[i]);
    }

    return 0;
}
```

Estrutura de Dados

Vetores em C

Exercicio 02:

Faça um cadastro de varias matriculas de alunos da faculdade e armazene-os em um vetor até o vetor ser preenchido por 18 matrículas. Esses números são distintos, ou seja, o vetor não armazenará valores repetidos.

Estrutura de Dados

Vetores em C

Exercicio 02:

Faça um cadastro de varias matriculas de alunos da faculdade e armazene-os em um vetor até o vetor ser preenchido por 18 matrículas. Esses números são distintos, ou seja, o vetor não armazenará valores repetidos.

```
#include<math.h>
#include<stdio.h>
#include<string.h>
int main(){

    int vetor[18], numero, cont, posicao = 0 ;

    while (posicao < 18){

        scanf("%d",&numero);

        if (posicao == 0){
            vetor[posicao] = numero;
            printf("%d\n",vetor[posicao]);
            posicao++;
        }
    }
```

Estrutura de Dados

Vetores em C

Exercicio 02:

Faça um cadastro de varias matriculas de alunos da faculdade e armazene-os em um vetor até o vetor ser preenchido por 18 matrículas. Esses números são distintos, ou seja, o vetor não armazenará valores repetidos.

```
//continuacao
```

```
else{
```

```
for(cont = 0; (cont < posicao)&&(vetor[cont]!= numero); cont++);
```

```
if (cont >= posicao){
```

```
    vetor[posicao] = numero;  
    printf("%d\n",vetor[posicao]);  
    posicao++;  
}
```

```
}
```

```
return 0;
```

```
}
```

Estrutura de Dados

Vetores em C

Exercicio 03:

Criar 4 vetores, o primeiro com a nota da primeira prova, o segundo com a nota da segunda prova e o terceiro, no quarto vetor com a média das 3 primeiras notas, e imprima o resultado “APROVADO” para aqueles que obtiverem uma média igual ou acima de 7, e “REPROVADO” para quem obtiverem uma média abaixo de 7.

OBS.: Saia do laço quando a primeira nota for igual a -7.

Estrutura de Dados

Vetores em C

Exercicio 03:

Criar 4 vetores, o primeiro com a nota da primeira prova, o segundo com a nota da segunda prova e o terceiro, no quarto vetor com a média das 3 primeiras notas, e imprima o resultado “APROVADO” para aqueles que obtiverem uma média igual ou acima de 7, e “REPROVADO” para quem obtiverem uma média abaixo de 7.

OBS.: Saia do laço quando a primeira nota for igual a -7.

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

```
int main(void){

    int contador = 0;
    float vetor[4],nota = 0 ;

    scanf("%f",&nota);
```

Estrutura de Dados

Vetores em C

Exercicio 03:

Criar 4 vetores, o primeiro com a nota da primeira prova, o segundo com a nota da segunda prova e o terceiro, no quarto vetor com a média das 3 primeiras notas, e imprima o resultado “APROVADO” para aqueles que obtiverem uma média igual ou acima de 7, e “REPROVADO” para quem obtiverem uma média abaixo de 7.

OBS.: Saia do laço quando a primeira nota for igual a -7.

```
while ( nota != -7){
```

```
    vetor[contador] = nota;  
    contador++;
```

```
    if (contador == 3){
```

```
        vetor[contador] = (vetor[0] + vetor[1] + vetor[2]) / 3;  
        printf("%.2f\n",vetor[contador]);
```

Estrutura de Dados

Vetores em C

Exercicio 03:

Criar 4 vetores, o primeiro com a nota da primeira prova, o segundo com a nota da segunda prova e o terceiro, no quarto vetor com a média das 3 primeiras notas, e imprima o resultado “APROVADO” para aqueles que obtiverem uma média igual ou acima de 7, e “REPROVADO” para quem obtiverem uma média abaixo de 7.

OBS.: Saia do laço quando a primeira nota for igual a -7.

```
        if(vetor[contador] >= 6){
            printf("APROVADO\n");
        }
        else{
            printf("REPROVADO\n");
        }
        contador = 0;
    }
    scanf("%f",&nota);

}
return 0;
}
```

Estrutura de Dados

Vetores em C

Exercicio 04:

Faça um algoritmo que leia um vetor de 10 posições, e preencha um segundo vetor, sendo que cada posição do segundo vetor receberá o valor do primeiro vetor na mesma posição multiplicado pelo maior valor dentro do primeiro vetor.

Estrutura de Dados

Vetores em C

Exercicio 04:

Faça um algoritmo que leia um vetor de 10 posições, e preencha um segundo vetor, sendo que cada posição do segundo vetor receberá o valor do primeiro vetor na mesma posição multiplicado pelo maior valor dentro do primeiro vetor.

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

```
#include <math.h>
```

```
#include <string.h>
```

```
int main(void)
```

```
{
```

```
int vetorA[10], vetorB[10], contador, maiornumero = 0;
```

```
for(contador = 0; contador < 10; contador++){
```

```
scanf("%d",&vetor1[contador]);
```

Estrutura de Dados

Vetores em C

Exercicio 04:

Faça um algoritmo que leia um vetor de 10 posições, e preencha um segundo vetor, sendo que cada posição do segundo vetor receberá o valor do primeiro vetor na mesma posição multiplicado pelo maior valor dentro do primeiro vetor.

```
if (maiornumero == 0){
    maiornumero = vetorA[contador];
}
else if( vetorA[contador] > maiornumero){
    maiornumero = vetorA[contador];
}
}

for( contador = 0; contador < 10; contador++){
    vetorB[contador] = vetorA[contador] * maiornumero;
    printf("%d\n",vetorB[contador]);
}

return 0;
}
```



Estrutura de Dados

Vamos falar sobre matrizes em C

Também chamado de vetores multidimensionais, esses tipos de vetores são divididos em linhas e colunas de dados.

Como por exemplo nesta declaração:

Estrutura de Dados

Vamos falar sobre matrizes em C

Também chamado de vetores multidimensionais, esses tipos de vetores são divididos em linhas e colunas de dados.

```
float mat[2][2]; //Tabela com 2 linhas e 2 colunas
```

Matriz M

i/j

	0	1
0		
1		

Estrutura de Dados

Vamos falar sobre matrizes em C

Também chamado de vetores multidimensionais, esses tipos de vetores são divididos em linhas e colunas de dados.

```
int x[2][5]; //Tabela com 2 linhas e 5 colunas
```

Uma melhor visualização seria :

<Variável(x) >	Coluna 0	Coluna 1	Coluna 2	Coluna 3	Coluna 4
Linha 0	2	54	60	23	44
Linha 1	4	63	7	11	55

Estrutura de Dados

Vamos falar sobre matrizes em C

Um exemplo de utilização dessa matriz seria para um programa para uma loja de departamentos com 2 filiais, cada uma vendendo 3 itens, poderia incluir um vetor declarado como abaixo:

```
int vendas [2][3];
```

Estrutura de Dados

Vamos falar sobre matrizes em C

Cada elemento vendas [i] [j] representa a quantidade do item j vendida na filial i, declarando como demonstrado abaixo:

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

int main(){
    int vendas [2][3]; int i,j;

    for(i=0;i<=1;i++){
        printf("Filial 0-: %d \n",i+1);
        for(j=0;j<=2;j++){
            printf("Digite o item 0-: %d:",j+1);
            scanf("%d",vendas[i,j]);
        }
        system("cls");
    }
    printf("Itens cadastrados com sucesso!!!"); return
    0;
}
```

Estrutura de Dados

Exercicio para ser feliz:

Construa um sistema de controle de estoque de uma pequena padaria. Seu programa deve oferecer um menu para :

- (a) cadastrar um novo produto;
- (b) aumentar o estoque de um produto cadastrado (quando há compras);
- (c) diminuir o estoque de um produto cadastrado (quando há vendas);
- e (d) imprimir os produtos cadastrados e suas características.

De cada produto deseja-se cadastrar: Nome (caracteres) e Preço (número real) além, da quantidade em estoque. Assuma que a padaria venderá no máximo, 100 produtos diferentes.

Estrutura de Dados

Exercicio para ser feliz:

OBS:

Para facilitar a localização de um produto (para as funções (b) e (c)), use a posição dos mesmos no vetor que representa o estoque como código do produto, informando-a ao fim do cadastro. Então, peça ao usuário para digitar o código do produto que deseja aumentar/diminuir o estoque.