

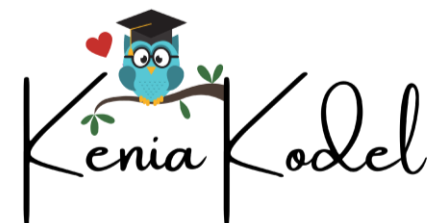


UNIVERSIDADE
FEDERAL DE
SERGIPE



M a t r i z e s

PROGRAMAÇÃO IMPERATIVA

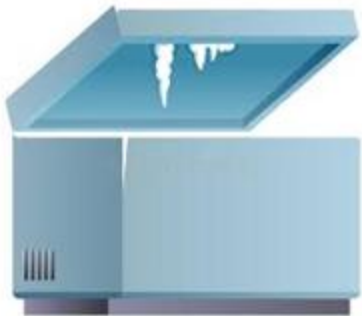


Matrizes (array bidimensional)

SITUAÇÃO PROBLEMA

Recebemos a tarefa de efetuar um levantamento de dados acerca de freezers.

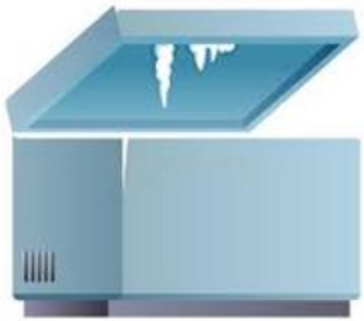
	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (KWh/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18



Matrizes (array bidimensional)

SITUAÇÃO PROBLEMA

Observar que temos dados em formato tabular - **matriz**.



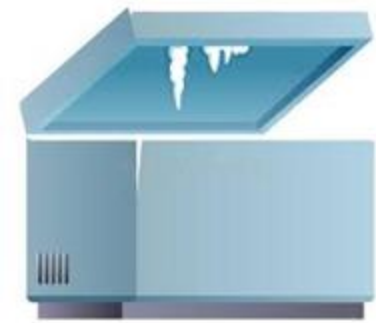
	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (Kwh/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18

Matrizes (array bidimensional)

SITUAÇÃO PROBLEMA



Quais as marcas dos freezers?



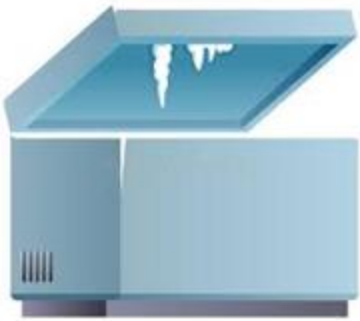
	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (Kwh/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18

Matrizes (array bidimensional)

SITUAÇÃO PROBLEMA



Quais os dados analisados?



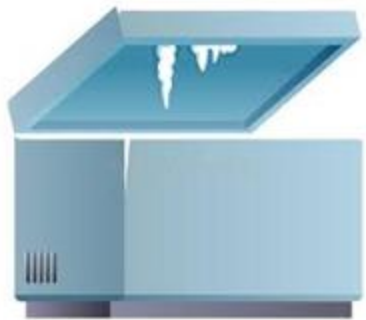
	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (Kwh/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18

Matrizes (array bidimensional)

SITUAÇÃO PROBLEMA



Como registrar a GARANTIA do freezer FRI?



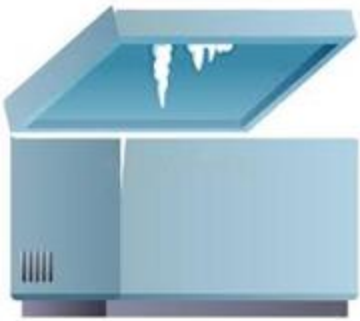
	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (Kwh/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18

Matrizes (array bidimensional)

SITUAÇÃO PROBLEMA



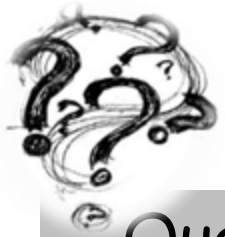
Quais os consumos de energia mensais que identificamos?



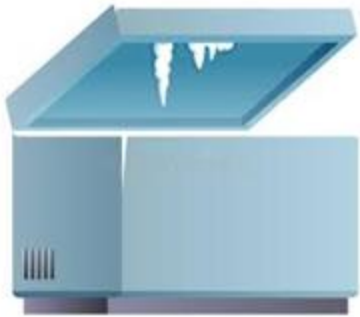
	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (Kwh/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18

Matrizes (array bidimensional)

SITUAÇÃO PROBLEMA



Quais os dados do freezer Lux?



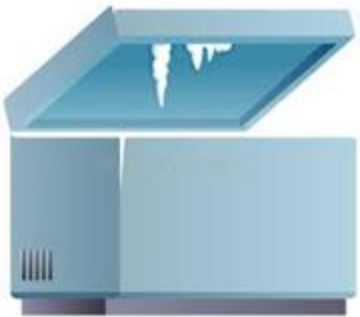
	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (Kwh/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18

Matrizes (array bidimensional)

SITUAÇÃO PROBLEMA



Qual a
ECONOMIA
do freezer
SUL?



	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (Kwh/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18

Matrizes (array bidimensional)

DEFINIÇÃO & SINTAXE

Uma matriz consiste num item de composição de programas de computador **útil ao armazenamento de dados encadeados**, dispostos em duas dimensões: linhas e colunas; ou mais.

<tipo> <Identificador> [N] [M] ;

Onde:

- <identificador> – nome da matriz
- N – quantidade máxima de elementos de composição da primeira dimensão (linhas) da estrutura
- M – quantidade máxima de elementos de composição da segunda dimensão (colunas) da estrutura
- <tipo> – tipo dos dados a serem mantidos na estrutura.

Matrizes (array bidimensional)

EXEMPLO


Uma matriz é útil ao armazenamento de dados encadeados, dispostos em duas dimensões: linhas e colunas; ou mais.

<tipo> <Identificador> [N] [M] ;

```
float M[][6]={ {385,534,309,546,503,477} ,  
               {12,121,12,3,24,12} ,  
               {35.9,72.1,46.2,74.3,78,96} ,  
               {-28,-18,-18,-18,-22,-18} } ;
```

Matrizes

SITUAÇÃO
PROBLEMA
INICIALIZAÇÃO



Recebemos a tarefa de efetuar um levantamento de dados acerca de freezers.

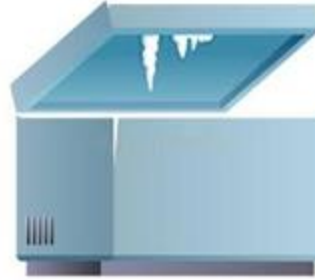
	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (kWh/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18

```
float M[][6]={ {385,534,309,546,503,477} ,  
                {12,121,12,3,24,12} ,  
                {35.9,72.1,46.2,74.3,78,96} ,  
                {-28,-18,-18,-18,-22,-18} };
```

Matrizes

(array bidimensional)

OPERAÇÕES



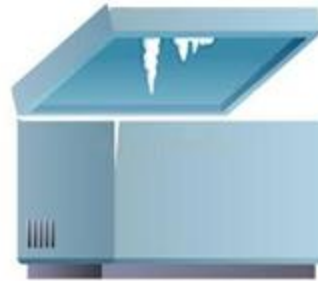
Quais os consumos de energia (mensais) identificados?



Matrizes

(array bidimensional)

OPERAÇÕES



Quais os consumos de energia (mensais) que identificamos?



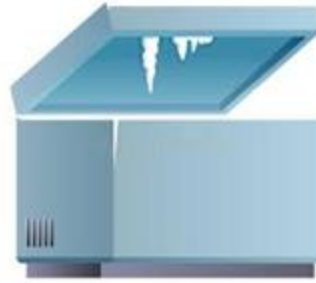
```
float M[][6]={ {385,534,309,546,503,477} ,  
                {12,121,12,3,24,12} ,  
                {35.9,72.1,46.2,74.3,78,96} ,  
                {-28,-18,-18,-18,-22,-18}};
```

```
void ExibeLinha (float Matriz[][6],int L,int C) {  
    for(int i=0;i<C;i++)  
        printf("%.1f ",Matriz[L][i]);}
```

Matrizes

(array bidimensional)

OPERAÇÕES



Quais os dados do freezer Lux?

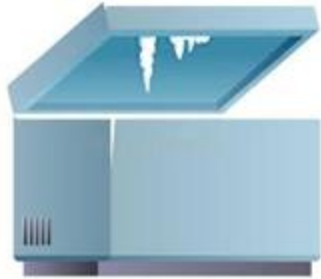


Matrizes

(array bidimensional)

OPERAÇÕES

	Deia	Sul	NSul	Frio	Fri	Lux
Capacidade (litros)	385	534	309	546	503	477
Garantia (meses)	12	12	12	3	24	12
Economia (KW/mês)	35,9	72,1	46,2	74,3	78	96
Temperatura Mínima °C	-28	-18	-18	-18	-22	-18



Quais os dados do freezer Lux?



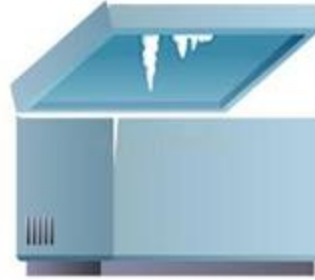
```
float M[][6]={ {385,534,309,546,503,477} ,  
                {12,121,12,3,24,12} ,  
                {35.9,72.1,46.2,74.3,78,96} ,  
                {-28,-18,-18,-18,-22,-18} };
```

```
void ExibeColuna(float Matriz[][6],int L,int C){  
    for(int i=0;i<L;i++)  
        printf("%.1f\n",Matriz[i][C]);}
```


Matrizes

(array bidimensional)

OPERAÇÕES



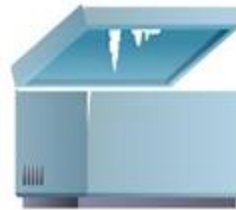
Quais os dados coletados?



Matrizes

(array bidimensional)

OPERAÇÕES



Quais os dados coletados?



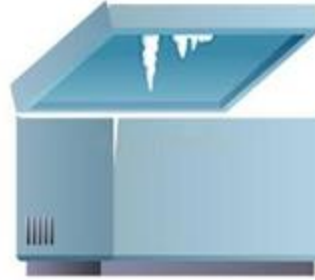
```
float M[][6]={ {385,534,309,546,503,477} ,  
                {12,121,12,3,24,12} ,  
                {35.9,72.1,46.2,74.3,78,96} ,  
                {-28,-18,-18,-18,-22,-18} };
```

```
void ExibeMatriz(float Matriz[][6],int L,int C){  
    for(int i=0;i<L;i++){  
        if (i==0) printf("Capacidade (litros) ");  
        if (i==1) printf("Garantia (meses) ");  
        if (i==2) printf("Economia (KWh/mês) ");  
        if (i==3) printf("Temperatura Min (C) ");  
        for (int j=0; j<C; j++)  
            printf("%6.1f",Matriz[i][j]);  
        printf("\n");  
    }  
}
```

Matrizes

(array bidimensional)

OPERAÇÕES



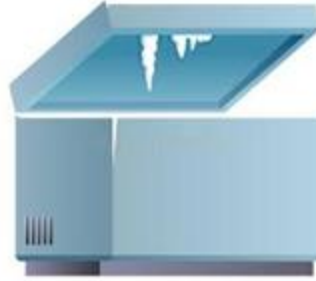
Como ajustar dados
da matriz?



Matrizes

(array bidimensional)

OPERAÇÕES



Como ajustar dados da matriz?



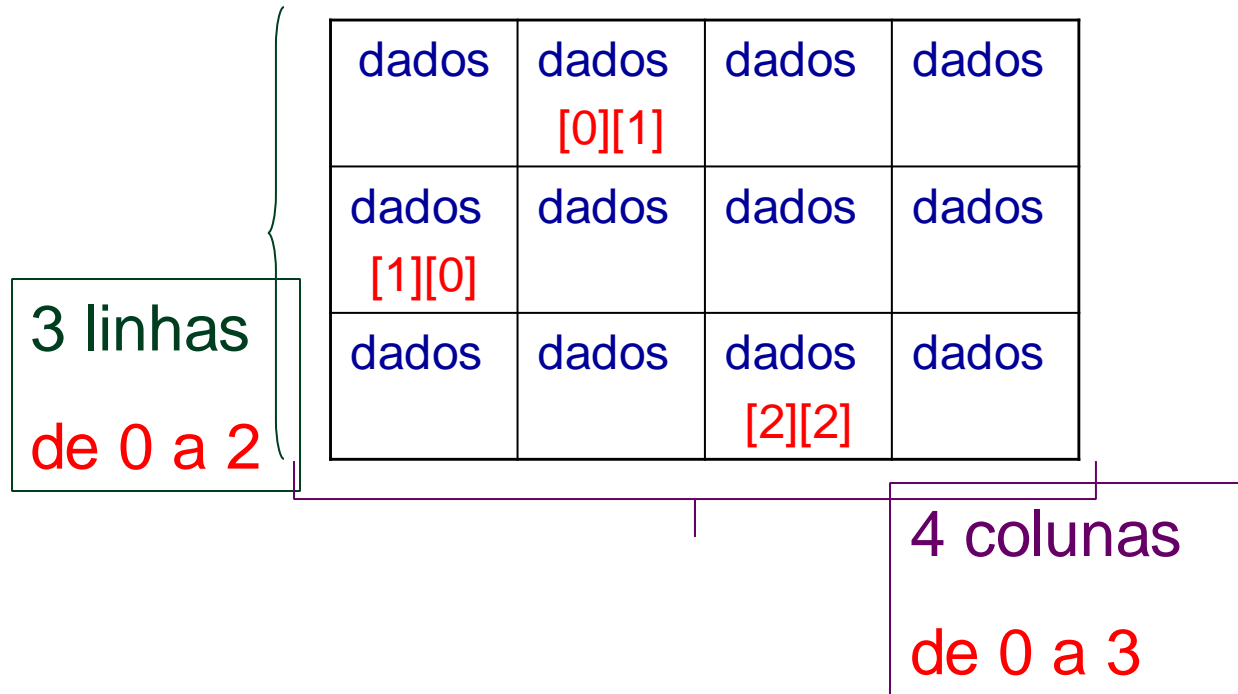
```
float M[][6]={ {385,534,309,546,503,477} ,  
                {12,121,12,3,24,12} ,  
                {35.9,72.1,46.2,74.3,78,96} ,  
                {-28,-18,-18,-18,-22,-18} };
```

```
printf("Freezer: ");  
int F;  
scanf("%d",&F);  
printf("Característica: ");  
int C;  
scanf("%d",&C);  
printf("\n\nQual o novo valor? ");  
scanf("%f",&M[C-1][F-1]);
```

Matrizes (array bidimensional)

OUTRA SITUAÇÃO PROBLEMA

- Esquemáticamente temos, por exemplo:



- Os dados são acessados por meio das coordenadas (linha e coluna) que ocupam na estrutura.

Matrizes

OUTRA SITUAÇÃO
PROBLEMA
INICIALIZAÇÃO

```
int Placar[4][6];  
correspondente a  
4 times e 6 jogos.
```

	J1	J2	J3	J4	J5	J6
T1	200	303	228	400	168	226
T2	252	314	115	142	172	215
T3	154	285	292	236	265	356
T4	327	336	346	365	243	401

`Placar[2][4]`
corresponde à
pontuação obtida
pelo time 3 no
jogo 5.

Matrizes (array bidimensional)

DEFINIÇÕES E OPERAÇÕES

- Cada célula de uma matriz bidimensional é identificada pelo nome da matriz seguido pelo par [linha][coluna] de localização desta (célula) na estrutura.

Tabuleiro[0][1]='X' ;

- Sendo *dados* valores de um tipo determinado qualquer.
- A estrutura completa é denominada matriz.
- **EXEMPLO:** Para composição do tabuleiro do jogo da velha poderia ser usada uma matriz com dimensão 3 x 3; como segue:

dados	X [0][1]	dados
dados [1][0]	dados	dados
dados	dados	dados [2][2]

```
char  Tabuleiro  [3][3];
```

Matrizes (array bidimensional)

OPERAÇÕES

- Os dados de tipo T mantidos em matriz são manipulados um-a-um.
- Também, estes são manipulados da mesma forma como variáveis não-compostas de tipo T são manipuladas.
- Assim: if (Tabuleiro[0][1]>Tabuleiro[1][2]) é uma operação possível. Sendo Tabuleiro uma matriz de inteiros, de um quebra-cabeça, por exemplo.
- Ou seja, cada dado da matriz se comporta tal qual uma variável de mesmo tipo de seus elementos.

06	05	08
01	04	03
	07	02

Matrices (array bidimensional)

EXEMPLOS

Inicializando matrices:

```
int matriz1[3][4]={ {1,2,3,4},  
                    {5,6,7,8},  
                    {9,10,11,12}};  
  
int matriz2[3][4]={ {11,21,13,41},  
                    {52,26,72,28},  
                    {39,13,31,13}};  
  
char matriz3[][3]={ {a,b,c},  
                    {d,e,f},  
                    {g,h,i},  
                    {j,k,l}};
```

a	b	c
d	e	f
g	h	i
j	k	l

matriz3

Matrizes

EXIBIÇÃO DE MATRIZ

```
void Exibir(int Matriz[][5], int L, int C)
{
    int i, j;
    for (i=0; i<L; i++)
        for (j=0; j<C; j++){
            printf(" %d ",Matriz[i][j]);
            if (j==C-1)
                printf("\n");}
    return;
}
```

Para exibir matriz A:

Exibir(A, 5, 4);

Para exibir matriz B:

Exibir(B, 5, 3);

Para exibir matriz C:

Exibir(C, 4, 4);

Para exibir matriz D:

Exibir(D, 5, 3);

MATRIZES

A	22	49	58	56
	52	54	14	15
	12	10	32	36
	69	68	65	62
	63	78	75	74
B	14	55	18	
	66	21	22	
	25	85	74	
	12	36	65	
	54	49	98	
C	78	89	94	45
	56	61	12	23
	32	21	16	65
	54	09	87	02
D	12	42	15	
	78	16	29	
	43	18	79	
	17	68	60	
	25	27	88	

Matrizes

SOMA MATRIZES

```
void SomarM(int M1[][3], int M2[][3],
            int L, int C, int MSoma[][3])
{
    int i, j;
    for (i=0; i<L; i++)
        for (j=0; j<C; j++)
            MSoma[i][j]=M1[i][j]+M2[i][j];
    return;
}
```

Para somar as matrizes B e D e obter a matriz E:

`SomarM(B, D, 5, 3, E);`

MATRIZES

22	49	58	56
52	54	14	15
12	10	32	36
69	68	65	62
63	78	75	74

A

14	55	18
66	21	22
25	85	74
12	36	65
54	49	98

B

78	89	94	45
56	61	12	23
32	21	16	65
54	09	87	02

C

12	42	15
78	16	29
43	18	79
17	68	60
25	27	88

D

Matrizes

SOMA MATRIZES

```
void SomarM(int M1[][5], int M2[][5],
            int L, int C, int MSoma[][5])
{
    int i, j;
    for (i=0; i<L; i++)
        for (j=0; j<C; j++)
            MSoma[i][j]=M1[i][j]+M2[i][j];
    return;
}
```

Para somar as matrizes B e D e obter a matriz E:

SomarM(B,D,5,3,E);



Se necessário o cálculo do total da matriz soma?

MATRIZES

22	49	58	56
52	54	14	15
12	10	32	36
69	68	65	62
63	78	75	74
14	55	18	
66	21	22	
25	85	74	
12	36	65	
54	49	98	
78	89	94	45
56	61	12	23
32	21	16	65
54	09	87	02
12	42	15	
78	16	29	
43	18	79	
17	68	60	
25	27	88	

A

B

C

D

Matrizes

SOMA MATRIZES

```
void SomarM(int M1[][3], int M2[][3],
            int L, int C,
            int MSoma[][3], int *T)
{
    int i, j;
    *T=0;
    for (i=0; i<L; i++)
        for (j=0; j<C; j++) {
            MSoma[i][j]=M1[i][j]+M2[i][j];
            *T=*T+MSoma[i][j];}
    return;
}
```

O cômputo do total deve ser sinalizado como parâmetro de saída, na definição e na invocação da função.

Para somar as matrizes B e D e obter a matriz E:

```
SomarM(B, D, 5, 3, E, &Total);
```

Matrizes

OPERAÇÃO SOBRE MATRIZES

```
int F(int M[][5], int L, int C)
{
    int Resultado;
    Resultado=M[0][0];
    int i, j;
    for (i=0; i<L; i++)
        for (j=0; j<C; j++)
            if (M[i][j] > Resultado)
                Resultado=M[i][j];
    return Resultado;
}
```

Qual o objetivo alcançado a partir da execução deste?



Matrizes

OPERAÇÃO SOBRE MATRIZES

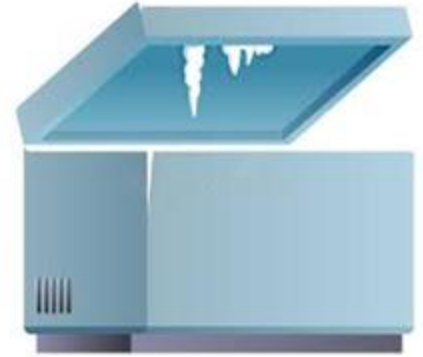
```
void Modulo(int M1[][3], int M2[][3],  
            int L, int C,  
            int *T)  
{  
    int i, j;  
    *T=0;  
    for (i=0; i<L; i++)  
        for (j=0; j<C; j++) {  
            *T=*T+M1[i][j]+M2[i][j];  
        }  
}
```



Qual o objetivo alcançado a partir da execução deste?

Matrizes

EXERCÍCIO



```
>>> Pesquisa Freezer <<<
```

```
Selecione a opção desejada:
```

- 1 - Exibir Levantamento
- 2 - Dados de um Freezer
- 3 - Dados de uma Característica
- 4 - Alterar Dado
- 5 - Melhor Freezer em Capacidade
- 6 - Melhor Freezer em Garantia
- 7 - Melhor Freezer em Economia
- 8 - Melhor Freezer em Temperatura
- 9 - Sair

```
█
```

...Freezer.c

```
switch (Op){  
    case 1: ExibeMatriz(M,4,6);  
            break;  
    ...  
    case 4: printf("\n_Altera Dado_");  
            break;  
    case 5: printf("\n_Melhor Freezer em  
Capacidade_");  
            break;  
    case 6: printf("\n_Melhor Freezer em Garantia_");  
            break;  
    case 7: printf("\n_Melhor Freezer em Economia_");  
            break;  
    case 8: printf("\n_Melhor Freezer em Temperatura_");  
            break;  
    case 9: printf("\n_Sair_");  
            break;}
```

Implementar
opções de 4 a 9.

Programação Imperativa

COMPLEMENTAR AULA...

Fundamentos da Programação de Computadores

Ana Fernanda Gomes Ascencio
Edilene Aparecida Veneruchi de Campos

Capítulos
Matriz



Programação Imperativa

COMPLEMENTAR AULA...

Curso de Linguagem C
UFMG

*linux.ime.usp.br/~lucas
mmg/livecd/documenta
cao/documentos/curso_
de_c/www.ppgia.pucpr.
br/_maziero/ensino/so/
projetos/curso-c/c.html*

Aula 5

Matrizes e Strings

[Aula 1: Introdução e Sumário](#)

[Aula 2 - Primeiros Passos](#)

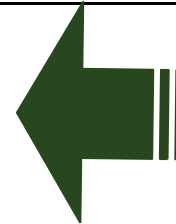
[Aula 3 - Variáveis, Constantes, Operadores e Expressões](#)

[Aula 4 - Estruturas de Controle de Fluxo](#)

[Aula 5 - Matrizes e Strings](#)

[Aula 6 - Ponteiros](#)

[Aula 7 - Funções](#)



Programação Imperativa

PRÓXIMO PASSO



Registros / Structs