

Exercício: Um professor quer armazenar os dados referentes ao gabarito de provas aplicadas a seus alunos em uma matriz. O professor possui 2 turmas, sendo que para cada turma serão aplicadas 3 provas, de 5 questões de escolha simples (respostas a,b,c,d,e) cada uma. Para cada prova, são armazenados os resultados de cada questão correspondente ao gabarito desta prova. Pede-se:

- declarar um **arranjo tridimensional** para armazenar as respostas de todas as provas de cada turma, isto é, os gabaritos;
- preencher** por leitura este arranjo;
- imprimir**, para cada turma a matriz de respostas de cada prova (colocando na tela os cabeçalhos explicativos dos conteúdos apresentados);
- imprimir** a(s) resposta(s) mais comum(ns) para as provas da segunda turma (a,b,c,d ou e).

Saídas:

- GABARITOS formatados (qual turma, qual prova)
 - Resposta com maior incidência (a,b,c,d ou e)
- Vetor contador de resposta** Como declarar?

```
//Programa gabarito
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#define TTURMA 2
#define TPROVA 3
#define TQUESTAO 5
int main( )
{
    char gabarito[TTURMA][TPROVA][TQUESTAO]; // armazena gabaritos
    int t,p,q; //índices para percorrer a matriz - turma prova questão
    char resp; // resposta
    char letra[5] = { 'a', 'b', 'c', 'd', 'e' }; // para impressão da opção com mais uso
    int cont [5]={0}; // vetor contador de cada resposta
    int i_cont; // índice do contador
    int maior; // maior ocorrência de respostas idênticas
    system("color f1");
}
```

```
Prova 1:
Questao 1: a
Questao 2: B
Questao 3: c
Questao 4: b
Questao 5: A
```

```
Prova 2:
Questao 1: -
```

```
// obtendo dados:
for (t = 0; t < TTURMA; t++) // turma
{
    printf("\nTurma %i: ",t+1);
    for (p=0;p < TPROVA; p++) // prova
    {
        printf("\n Prova %i: ",p+1);
        for (q=0;q < TQUESTAO; q++) // questões
            do // repete, até resposta estar entre caracteres a e e
            { // lê, consistindo e convertendo para minúscula
                printf("\n Questao %i: ",q+1);
                // precisa 1 espaço antes do %c ou fflush(stdin) antes do scanf:
                scanf(" %c", &gabarito[t][p][q]);
                gabarito[t][p][q] = tolower(gabarito[t][p][q]);
                while (gabarito[t][p][q] < 'a' || gabarito[t][p][q] > 'e');
            }
    }
}

// para ler com getchar, teria que usar o fflush(stdin) antes da leitura:
fflush(stdin);
gabarito[t][p][q] = tolower(getchar()); // lê e já converte
```

```
//imprimir as respostas para cada turma
for (t = 0; t < TTURMA; t++) // para cada turma
{
    printf("\nTurma %i: ",t+1);
    for (p=0;p < TPROVA; p++)
    {
        printf("\n Prova %i: \n ",p+1);
        for (q=0;q < TQUESTAO; q++)
            printf("\n Questao %i: %c ", q + 1, gabarito[t][p][q]);
        printf("\n\n");
    }
}

//conta as respostas das provas da 2a turma:
for (p=0;p < TPROVA; p++)
    for (q=0;q < TQUESTAO; q++)
        { //associa resposta à índice
            switch (gabarito[1][p][q])
            {
                case 'a': i_cont = 0; // poderia somar direto em cont[0]...
                    break;
                case 'b': i_cont = 1;
                    break;
                case 'c': i_cont = 2;
                    break;
                case 'd': i_cont = 3;
                    break;
                case 'e': i_cont = 4; // aqui não precisa break
            }
            cont[i_cont] = cont[i_cont] + 1; // soma aqui
        }
```

```
//Programa gabarito
#include <stdlib.h>
int main( )
{
    char gabarito[TTURMA][TPROVA][TQUESTAO];
    int t,p,q; //índices da matriz - turma prova questão
    char resp; // resposta
    char letra[5] = { 'a', 'b', 'c', 'd', 'e' };
    int cont [5]={0}; // vetor contador de cada resposta
    int i_cont; // índice do contador
    int maior; //respostas com opção mais comum

    // inicializa maior ocorrência de resposta (a, b, c, d ou e) :
    maior=0;
    //determina a opção mais respondida:
    for (i_cont=0; i_cont < 5; i_cont++)
        if (cont[i_cont] > maior)
            maior = cont[i_cont];

    //imprime a(s) mais respondida:
    printf("\n\nResposta(s) mais comum para segunda turma: ");
    printf("%d ocoerencias \n ---", maior);
    for (i_cont=0; i_cont < 5; i_cont++)
        if (cont[i_cont] == maior)
            // usa vetor letra[i_cont], que contém a, b, c, d ou e:
            printf("%c ", letra[i_cont]); //
    printf("\n\nFinal de programa \n\n");
    system("pause");
    return 0;
}
```

```
Prova 2:
Questao 1: a
Questao 2: b
Questao 3: d
Questao 4: b
Questao 5: a

Prova 3:
Questao 1: a
Questao 2: b
Questao 3: a
Questao 4: b
Questao 5: e

Resposta(s) mais comum para segunda turma: 6 ocoerencias
--> a b
```

Exercício

1. Uma **matriz esparsa** é uma matriz que tem aproximadamente 2/3 de seus elementos iguais a zero. Fazer um programa que lê (linha à linha) uma matriz esparsa **me**(MAXL,MAXC) contendo valores inteiros e forma uma matriz condensada **mc**, de apenas três colunas, contendo os elementos não nulos de **M**, de forma que:

- a primeira coluna contém os valores não nulo de **me**;
- a segunda coluna contém a linha de **me** onde foi encontrado o valor armazenado na coluna 1;
- a terceira coluna contém a coluna de **me** onde foi encontrado o valor armazenado na coluna 1.

Imprimir as duas matrizes, APÓS o preenchimento da matriz condensada!

- Determinar o número de linhas de **mc** faz parte do problema.
- Testar o programa com os seguintes dados de entrada:

0	0	1	0	2	-3	0	0	4	0	0	5	0	0	0	0	0	0	6	0	0	0	0	0	7	0	0	0	8
(linha 1)	(linha 2)	(linha 3)	(linha 4)	(linha 5)																								

```
//le uma matriz esparsa e gera uma matriz condensada
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#define MAXL 3 // número de linhas
#define MAXC 3 // número de colunas
int main( )
{
    int lin, col, linha_cond = 0;
    int esparsa[MAXL][MAXC];
    int condensada[(MAXL * MAXC) / 3][3]; // linhas condensada: 1/3
    //leitura dos dados
    for (lin = 0; lin < MAXL; lin++)
        for (col = 0; col < MAXC; col++)
        {
            printf("Digite o elemento [%d] [%d]: ", lin, col);
            scanf("%d", &esparsa[lin][col]);
        }
    ...
}
```

Slide 7

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```
// escrita da matriz esparsa
printf("\nMatriz esparsa lida:\n");
for (lin = 0; lin < MAXL; lin++)
{
    for (col = 0; col < MAXC; col++)
        printf("%5d", esparsa[lin][col]);
    printf("\n\n");
}
...
}
```

Slide 8

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```
// geracao da matriz condensada
for (lin = 0; lin < MAXL; lin++)
    for (col = 0; col < MAXC; col++)
        if (esparsa[lin][col])
        {
            condensada[linha_cond][0] = esparsa[lin][col];
            condensada[linha_cond][1] = lin;
            condensada[linha_cond][2] = col;
            linha_cond++; // também informa número de linhas preenchidas
        }
// escrita da matriz condensada, que terá linha_cond linhas preenchidas
printf("\nMatriz condensada:\n");
for (lin = 0; lin < linha_cond; lin++)
    printf("\n %6d %6d %6d", condensada[lin][0],
           condensada[lin][1],
           condensada[lin][2]);
printf("\n\nFinal de programa!\n");
system("pause");
return 0;
}
```

Slide 9

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Exercício:

Palíndromos são seqüências de caracteres que não mudam, se lidos da esquerda para a direita ou vice-versa (2002, ANA, ACAIACA, RADAR, AMOR é ROMA, eva asse essa ave, ame o poema).

Faça um programa que verifique se uma variável *string* é um palíndromo. Não precisa prever brancos intermediários.

Algoritmo palindromos

1. Inicio
2. Ler (var_string_lido);
3. Se var_string = inverso(var_string) // inverte string então escreva (var_string_lido, 'é um palindromo');
4. Fim.

Slide 10

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```
// Palindroma, invertendo palavra:
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int main( )
{
    char linha[81], linhainv[81]; // conteúdo digitado + invertido
    int i;
    system("color f1");
    do // lê, consistindo tamanho, mas sem evitar preenchimento da memória
    {
        fflush(stdin); // esvazia buffer
        printf("Informe palavra (até 80 caracteres): \n");
        scanf("%s", linha);
    } while (strlen(linha) > 80);
    for (i=0; i < strlen(linha); i++) // inverte
        linhainv[i] = linha[strlen(linha)-i-1];
    linhainv[strlen(linha)] = '\0'; // inclui finalizador de string
    printf("Lida: %s, invertida: %s\n", linha, linhainv);
    if (strcmp(linha, linhainv) == 0)
        printf("\n%s eh uma palindroma!", linha);
    else
        printf("\n%s nao eh uma palindroma!", linha);
    system("pause");
    return 0;
}
```

```
// Palindroma, invertendo palavra, depois de eliminar brancos:
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int main( )
{
    char linha[81], linha[81], linhainv[81]; // conteúdo lido, convertido + invertido
    int i, j;
    system("color f1");
    do // lê, consistindo tamanho, mas sem evitar preenchimento da memória
    {
        fflush(stdin);
        printf("Informe palavra (até 80 caracteres): \n");
        gets(linha); // para ler brancos! O gets NÃO inclui o '\n' !!!!
    } while (strlen(linha) > 80);
    // excluindo brancos
    for (i=0; i < strlen(linha); i++)
        if (linha[i] != ' ') // se não for branco
        {
            linha[j] = linha[i];
            j++;
        }
    linha[j] = '\0'; // inclui finalizador de string
    for (i=0; i < strlen(linha); i++) // inverte
        linhainv[i] = linha[strlen(linha)-i-1];
    system("pause");
    return 0;
}
```

atenção!

// só no fgets é que precisa substituir enter - '\n' - por finalizador linha[strlen(linha)-1] = '\0';

Slide 16

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

// Palíndroma, comparando posições:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char linha[81]; // conteúdo digitado
    int i_inic, i_fim, eh_pal;
    system("color f");
    do // lê, consistindo tamanho
    {
        fflush(stdin); // esvazia buffer
        .....
    } while (strlen(linha) > 80);
    eh_pal=1; // até prova em contrário, considera como palíndromo
    i_inic=0;
    i_fim=strlen(linha) - 1;
    // repete enquanto i_inic < i_fim e eh_pal = 1
    while (i_inic < i_fim && eh_pal== 1)
    // compara extremos, até chegar no meio
    {
        if (linha[i_inic] != linha[i_fim])
            eh_pal = 0;
        else
        {
            i_inic++;
            i_fim--;
        }
    }
    if (eh_pal== 1)
        printf("\n%s eh uma palindroma!",linha);
    else
        printf("\n%s nao eh uma palindroma!",linha);
    .....
}
```

Outras funções de *string*

- Formato: `strncat(destino,origem,nro_de_caracteres);`
Anexa apenas `nro_de_caracteres`
- Formato: `struprt(variável);`
Converte todos os caracteres para maiúsculas.
- Formato: `strlwr(variável);`
Converte todos os caracteres para minúsculas.
- Formato: `strrev(variável);`
Inverte os caracteres de uma string (fica invertido!)

Slide 14

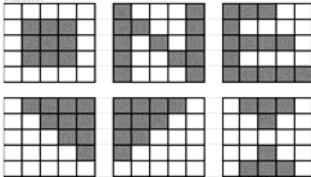
Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Exercício

2 - Preencher por leitura uma matriz M (5,5).

Em seguida, calcular e imprimir a média dos elementos das áreas assinaladas:



Slide 15

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Preencher por leitura uma matriz m (5,5) e escrevê-la

//Calcula media sobre elementos variados de
//uma matriz bidimensional

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#define MAX 5
int main()
{
    float m[MAX][MAX];
    //somatorios para cada caso
    float soma1, soma2, soma3, soma4, soma5, soma6;
    //medias para cada caso
    float media1, media2, media3, media4, media5, media6;
    int lin, col, contaelem, limite;
    //leitura dos valores da matriz m:
}
```

Slide 16

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Preencher por leitura uma matriz m (5,5) e escrevê-la

```
//leitura dos valores da matriz m
for (lin = 0; lin < MAX; lin++)
    for (col = 0; col < MAX; col++)
    {
        printf ("\nInforme o valor [%d] [%d]: ", lin, col);
        scanf ("%f", &m[lin][col]);
    }

//escrita da matriz m
printf ("\n\nMatriz lida: \n");
for (lin = 0; lin < MAX; lin++)
{
    for (col = 0; col < MAX; col++)
        printf ("%6.2f ", m[lin][col]);
    printf ("\n");
}
...
```

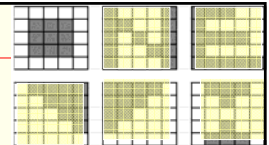
Slide 17

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Calcular e imprimir a média dos elementos da Figura 1:

```
m[2,2] m[2,3] m[2,4]
m[3,2] m[3,3] m[3,4]
m[4,2] m[4,3] m[4,4]
Linha: 2 a 4
Coluna: 2 a 4
```



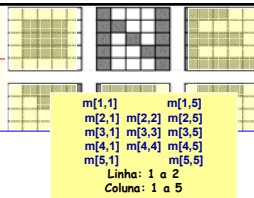
```
//primeira figura
soma1 = 0; contaelem = 0;
for (lin = 1; lin < 4; lin++)
    for (col = 1; col < 4; col++)
    {
        soma1 = soma1 + m[lin][col];
        contaelem++;
    }
media1 = soma1 / contaelem;
printf ("\nMedia da figura 1: %6.2f (%d elementos considerados) \n",
        media1, contaelem);
...
```

Slide 18

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Calcular e imprimir a média dos elementos da Figura 2:



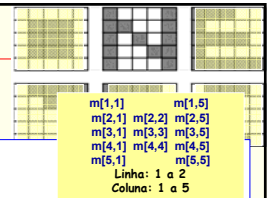
```
//segunda figura, opção 1:
soma2 = 0; contaelem = 0;
for (lin = 0; lin < MAX; lin++)
    switch (lin)
    {
        case 0: //linha 0 e 4
        case 4 : soma2 = soma2 + m[lin][0] + m[lin][MAX - 1];
                contaelem = contaelem + 2;
                break;
        case 1: //linhas 1, 2 e 3
        case 2:
        case 3: soma2=soma2 + m[lin][0] + m[lin][MAX-1] + m[lin][lin];
                contaelem = contaelem + 3;
    }
media2 = soma2 / contaelem;
printf("\nMedia da figura 2: %6.2f (%d elementos considerados) \n",
        media2, contaelem );
...
```

Slide 19

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Calcular e imprimir a média dos elementos da Figura 2:



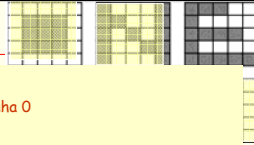
```
//segunda figura, opção 2:
soma2 = 0; contaelem = 0;
for (lin = 0; lin < MAX; lin++)
    {
        soma2 = soma2 + m[lin][0] + m[lin][5];
        contaelem = contaelem + 2;
        // testa se nesta linha a diagonal principal está fora das colunas 0 e 5:
        if (lin > 0 && lin < 4)
        {
            soma2 = soma2 + m[lin][lin]; // soma [1][1], [2][2] e [3][3]
            contaelem = contaelem + 1;
        }
        media2 = soma2 / contaelem;
        printf("\nMedia da figura 2: %6.2f (%d elementos considerados) \n",
                media2, contaelem );
    }
...
```

Slide 20

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Calcular e imprimir a média dos elementos da Figura 3:



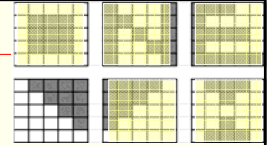
```
//terceira figura
soma3 = 0; contaelem = 0;
limite = 3; // última coluna considerada na linha 0
for (lin = 0; lin < MAX; lin++)
    switch (lin)
    {
        case 0: //linhas 0, 2 e 4
        case 2:
        case 4: for (col = 0; col < limite; col++)
                {
                    soma3 = soma3 + m[lin][col];
                    contaelem++;
                }
                limite++; //limite da coluna varia de 2 a 4
                break;
        case 1: //linhas 1 e 3
        case 3: soma3 = soma3 + m[lin][0];
                contaelem++;
    }
media3 = soma3 / contaelem;
printf("\nMedia da figura 3: %6.2f (%d elementos considerados) \n",
        media3, contaelem );
...
```

Slide 21

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Calcular e imprimir a média dos elementos da Figura 4:



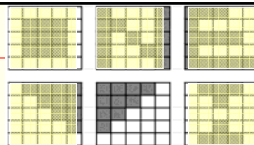
```
//quarta figura
soma4 = 0; contaelem = 0;
for (lin = 0; lin < MAX; lin++)
    for (col = lin + 1; col < MAX; col++)
        //em cada linha, valor para a coluna varia de linha + 1 até MAX - 1
        {
            soma4 = soma4 + m[lin][col];
            contaelem++;
        }
media4 = soma4 / contaelem;
printf("\nMedia da figura 4: %6.2f (%d elementos considerados) \n",
        media4, contaelem );
...
```

Slide 22

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Calcular e imprimir a média dos elementos da Figura 5: versão 1



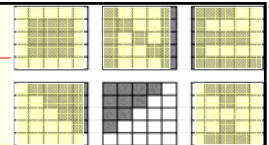
```
//quinta figura - versao 1
soma5 = 0; contaelem = 0;
// para cada linha, coluna varia de 0 a 4 - linha
for (lin = 0; lin < MAX - 1; lin++)
    for (col = 0; col < (MAX - 1) - lin; col++)
        //limite para a coluna: de 3 a 0
        {
            soma5 = soma5 + m[lin][col];
            contaelem++;
        }
media5 = soma5 / contaelem;
printf("\nMedia da figura 5: %6.2f (%d elementos considerados) - versao 1 \n",
        media5, contaelem );
...
```

Slide 23

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Calcular e imprimir a média dos elementos da Figura 5: versão 2



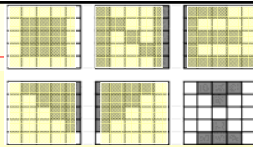
```
//quinta figura - versao 2
soma5 = 0; contaelem = 0;
limite = MAX - 1;
for (lin = 0; lin < MAX - 1; lin++)
    {
        for (col = 0; col < limite; col++)
            {
                soma5 = soma5 + m[lin][col];
                contaelem++;
            }
        limite--; //limite para a coluna varia de MAX - 1 a 0,
                // dependendo da linha
    }
media5 = soma5 / contaelem;
printf("\nMedia da figura 5: %6.2f (%d elementos considerados) - versao 2 \n",
        media5, contaelem );
...
```

Slide 24

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Calcular e imprimir a média dos
elementos da Figura 6:



```
//sexta figura
soma6 = 0; contaelem = 0;
for (col = 1; col < MAX - 1; col++)
    switch (col)
    {
        case 2: for (lin = 0; lin < MAX; lin++)
            //na coluna central, acessa as linhas 0, 1, 3 e 4
            if (lin != 2) //ignora linha 2
            {
                soma6 = soma6 + m[lin][col];
                contaelem++;
            }
            break;
        case 1:
        case 3: //colunas 1 e 3, acessa linhas 0 e MAX-1
            soma6 = soma6 + m[0][col] + m[MAX - 1][col];
            contaelem = contaelem + 2;
    }
}
media6 = soma6 / contaelem;
printf("\nMedia da figura 6: %6.2f (%d elementos considerados) \n",
    media6, contaelem);
system("pause");
return 0;
}
```