# Vetores Multidimensionais

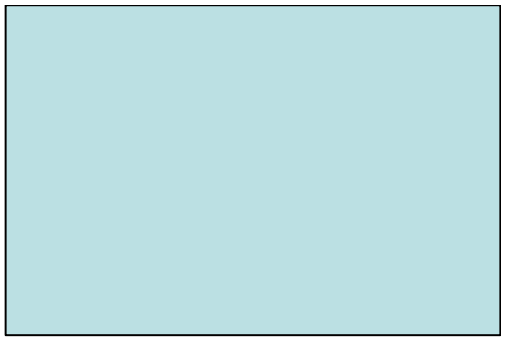
Exercício: Construa um programa, na linguagem C, que declare uma matriz 7x4 de números em ponto flutuante, a inicialize com valores fornecidos pelo usuário, através da entrada padrão, e a retorne na saída padrão com o layout a seguir:

| | | | |
|---|---|---|---|
| x.xx | x.xx | ... | x.xx |
| x.xx | x.xx | ... | x.xx |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |

10 10 ...

```c
#include <stdio.h>
#define nl 7
#define nc 4
int main() {
    float matriz[nl][nc];
    int i, j;
    for (i=0;i<7;i++)
        for (j=0;j<nc;j++) {
            printf ("\nEntre com matriz[%d][%d]=",i+1,j+1);
            scanf ("%f",&matriz[i][j]);
         }
    for (i=0;i<nl;i++) {
        printf("\n|");
        for (j=0;j<nc;j++)
            printf ("%10.2f",matriz[i][j]);
        printf(" |");
     }
}
```

# Vetores Multidimensionais

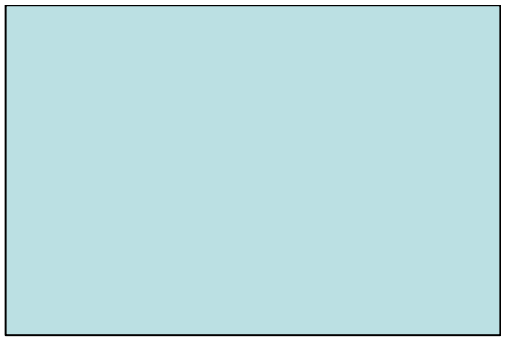


Exercício: Construa um programa, na linguagem C, que declare uma matriz 5x5 de números naturais, a inicialize com valores fornecidos pelo usuário, através da entrada padrão, e, após a inicialização, efetue uma pesquisa nos elementos retornando na saída padrão a soma dos elementos localizados abaixo da diagonal principal.

```c
#include <stdio.h>
#define nl 5
#define nc 5
int main() {
   int matriz[nl][nc], i, j, aux;
   for (i=0;i<nl;i++)
      for (j=0;j<nc;j++)
      do {
            printf ("\nEntre com um valor natural para matriz[%d][%d]=",i+1, j+1);
            scanf ("%f",&matriz[i][j]);
      } while (matriz[i][j]<0);
   for (aux=0, i=0;i<nl;i++)
      for (j=0;j<nc;j++)
         if (i>j)
            aux+= matriz[i][j];
   printf ("A soma dos elementos posicionados abaixo da diagonal principal é %d\n",aux);
}
```

# Vetores de Strings

# Vetores de Strings



Se fizermos um vetor de strings estaremos construindo um vetor de vetores.

Esta estrutura é uma matriz bidimensional de **char**'s.

Podemos ver a forma geral de uma vetor de strings como sendo:

***char nome_da_variável [num_de_strings][compr_das_strings];***

# Vetores de Strings

Aí surge a pergunta: como acessar uma string individual?

Fácil. É só usar apenas o primeiro índice. Então, para acessar uma determinada string faça:

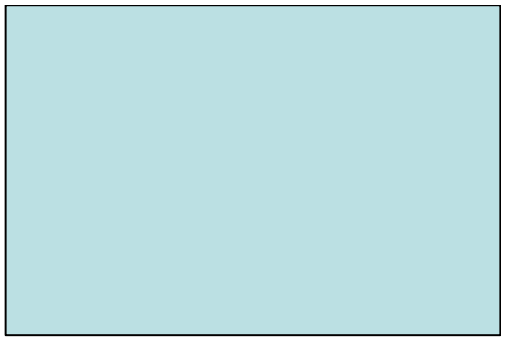
*nome_da_variável [índice]*

# Vetores de Strings

**Exemplo:** O programa a seguir declara um vetor de string's, o inicializa com string's fornecidas através da entrada padrão e no final de seu processamento o retorna na saída padrão.

```c
#include <stdio.h>
int main ()
{
   char strings [5][100];
   int count;
   for (count=0;count<5;count++)
   {
      printf ("\n\nDigite a %dº string do vetor: ", count+1);
      scanf ("%99s", strings[count]);
   }
   printf ("\n\nAs strings que voce digitou foram:\n");
   for (count=0; count<5; count++)
      printf ("%s\n",strings[count]);
}
```

# Vetores de Strings



**Exercício:**

Construa um programa que, com base no exemplo anterior, além de ler as 5 string's do vetor de strings leia mais uma string, a qual ele verificará se pertence ao vetor, caso esta pertença ele retornará a posição da string no vetor, caso contrario ele retornará uma mensagem indicando que ela não se encontra no vetor.

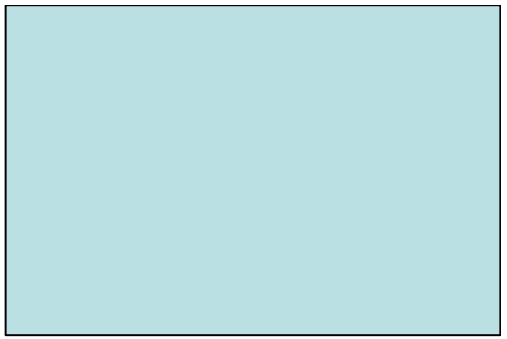
UNIVASF

```c
#include <stdio.h>
#include <stdlib.h>
int main () {
   char strings [5][100],string [100];
   int count,count2;
   for (count=0;count<5;count++) {
      printf ("\n\nDigite a %dº string do vetor: ", count+1);
      scanf ("%99[^\n]",strings[count]); /*Se necessário limpar o buffer*/
   }
   printf ("\nEntre com a string que voce deseja saber se pertence ao vetor:");
   scanf ("%99[^\n]", string); /*Se necessário limpar o buffer*/
   for (count=0;count<5;count++)
      for (count2=0;count2<100;count2++) {
         if (strings[count][count2]!=string[count2])
            break;
         if (string[count2]=='\0') {
            printf ("\nA string %s esta na posicao %d do vetor    de strings.\n", string, count+1);
            exit(0);
         }
      }
   printf ("\nA string %s nao esta contida no vetor de strings.\n", string);
}
```

```c
#include <string.h>
#include <stdio.h>
int main () {
   char strings [5][100],string [100];
   int count;
   for (count=0;count<5;count++) {
      printf ("\n\nDigite a %dº string do vetor: ", count+1);
      gets (strings[count]);
   }
   puts ("\nEntre com a string que voce deseja  saber se pertence ao vetor:");
   gets(string);
   for (count=0;count<5;count++)
      if (!strcmp(strings[count],string))
         break;
   if (count<5)
      printf ("\nA string %s esta na posicao %d do vetor    de strings.\n", string, count+1);
   else
      printf ("\nA string %s nao esta contida no vetor de   strings.\n", string);
}
```

# Vetores de Strings

Funções de entrada e saída formatada que trabalham sobre strings



**- sprintf e sscanf**

sprintf e sscanf são semelhantes a printf e scanf. Porém, ao invés de escreverem na saída padrão ou lerem da entrada padrão, escrevem ou lêem em uma string. Suas formas gerais são:

***sprintf (string_destino, string_de_controle, lista_de_argumentos);***

***sscanf (string_origem, string_de_controle, lista_de_argumentos);***

# Vetores de Strings

Funções de entrada e saída formatada que trabalham sobre strings (continuação)

Estas funções são muito utilizadas para fazer a conversão entre dados na forma numérica e sua representação na forma de strings e vice-versa. No programa a seguir, por exemplo, a variável i é "impressa" em string1. Além da representação de i como uma string, string1 também conterá "Valor de i = " .

# Vetores de Strings

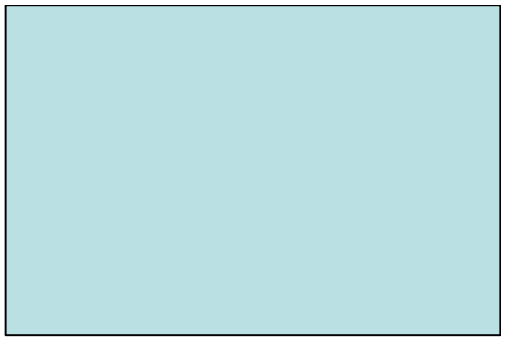
Funções de entrada e saída formatada que trabalham sobre strings (continuação)

**Exemplo:**

```c
#include <stdio.h>
int main()
{
    int i;
    char string1[25];
    puts( "Entre um valor inteiro: ");
    scanf("%d", &i);
    sprintf(string1,"Valor de i = %d", i);
    puts(string1);
}
```

# Vetores de Strings

Funções de entrada e saída formatada que trabalham sobre strings (continuação)



Já no próximo programa, foi utilizada a função *sscanf* para converter informações armazenadas em string1 em valores numéricos:

# Vetores de Strings

Funções de entrada e saída formatada que trabalham sobre strings (continuação)

Exemplo:

```c
#include <stdio.h>
int main()
{
   int i, j;
   float k;
   char string1[]= "10 20 5.89";
   sscanf(string1, "%d %d %f", &i, &j, &k);
   printf("Valores lidos: %d, %d, %.2f", i, j, k);
}
```

# Vetores de Strings

Funções de entrada e saída formatada que trabalham sobre strings (continuação)

**Exercício:**

Construa um programa que declare um vetor de strings e outro de inteiros, ambos com 10 elementos. O programa deve inicializar os vetores com valores fornecidos pelo usuário através da entrada padrão. Depois, deve acrescentar os inteiros, do vetor de inteiros, no final das strings correspondentes no vetor de strings. Ao termino do processamento o vetor de strings, com seus valores atualizados, deve ser apresentado na saída padrão.

UNIVASF

```c
#include<stdio.h>
#include<string.h>
int main() {
   char strings[10][100],aux[10];
   int inteiros[10],i;
   for (i=0;i<10;i++) {
      printf("\nEntre com a string[%d]: ",i+1);
      scanf("%99[^\n]", strings[i]);
      printf("\nEntre com o inteiro[%d]: ",i+1);
      scanf("%d",&inteiros[i]);
   }
   for (i=0;i<10;i++) {
      sprintf(aux,"%d",inteiros[i]);
      strcat(strings[i],aux); /*verificar possível falha de segmentação*/
   }
   for (i=0;i<10;i++)
      printf("\n%s\n",strings[i]);
}
```