**UEMG – Universidade do Estado de Minas Gerais – Unidade Frutal**

**Curso – Sistemas de Informação, 1º Período, noturno**

**Disciplina – Introdução à programação**

**Docente – Prof. Sergio Portari**

**Discentes – Estevam Gelain Anselmo**

**Trabalho – Matrizes e Vetores**

**Data – 13/08/2021**

**Strings**

Pode se dizer que uma string é uma cadeia de caracteres, podendo ser usada para representar palavras, frases ou textos de um programa. Os caracteres irão ser armazenados na memória que é representado por um valor numérico. Em uma declaração de string precisa pôr o seu índice, por exemplo quantas letras irá conter uma determinada palavra, nesta declaração ira precisar conter um numero a mais porque sempre o ultimo será destinado ao “\0”.

Para ler uma string podemos utilizar o “gets(string)” onde nela ao contrário do “scanf” não precisa endereçar na memória. Já para nos mostrarmos podemos usar o “printf(“%c”,string[x])”.

**Matrizes e Vetores**

Podemos definir matrizes como uma série de variáveis de uma mesma espécie referenciadas por um mesmo nome podendo diferenciar eu através de um numero à sua frente dentre colchetes que pode ser chamado de índice.

Ex:

Float casa[10][2];

Já o vetor é quase a mesma coisa que uma matriz, mas com apenas uma dimensão.

Ex;

Int livro[5];

**Exemplo explicativo de vetor:**

#include <stdio.h>

int main()

{

float corrida[10]; // aqui está uma declaração de um vetor onde ele só possui uma "linha"

for(int i = 0; i < 4; i++)

{

printf("Digite o tempo do piloto %d :",i+1);

scanf("%f",&corrida[i]);// aqui está o scanf com a variável corrida e onde ela vai armazenar que seria o valor da repetição.

}

for(int i = 3; i >=0 ; i--)

{

printf("\nO tempo do piloto %d foi: %f",i+1,corrida[i]); //na atribuição precisou colocar apenas o i para ser identificado na repetição.

}

return 0;

}

**Exemplo explicativo de Matriz:**

#include <stdio.h>

int main()

{

float corrida[10][10]; //aqui está a declaração da variável onde se encontra o número de karts e o número de pilotos.

int k, p; // k= kart e p= piloto

printf("O programa ira falar o número do kart e do piloto e pedir o tempo do piloto\n");

for (k = 0; k < 3; k++) //aqui se inicia uma repetição para avisar qual é o kart que o piloto testou.

{

for (p = 0; p < 3; p++) // aqui se inicia uma repetição para avisar qual o piloto que usou o kart informado primeiro.

{

printf("Digite o tempo do kart %d com o piloto %d ",k+1, p+1);

scanf("%f",&corrida[k][p]); // no scanf precisa prestar muita atenção para que não se inverta as atribuições.

}

}

printf("\n\n\n");

for (k = 0; k < 3; k++) //aqui se inicia uma repetição para avisar qual é o kart que o piloto testou.

{

for (p = 0; p < 3; p++) // aqui se inicia uma repetição para avisar qual o piloto que usou o kart informado primeiro.

{

printf("O tempo do kart %d com o piloto %d foi de: %f \n",k+1, p+1, corrida[k][p]);

//primeiro foi atribuída as máscaras para informar o piloto e o tempo e depois veio a variável corrida, onde ela contém o tempo.

}

}

return 0;

}

**strcpy()**

A sua forma é:

 strcpy (string\_destino,string\_origem);

A sua função é copiar uma string origem para uma string de destino então quando você vai apresentar ele vai mostrar a string destino que tem os mesmos dados da origem. Geralmente em um algoritmo simples declara 2 strings onde a “string1” vai receber o dado, após com a função strcpy ela vai mandar a informação para a “string2”. Também podemos usar essa função para apresentar na tela, basta declarar a variável e escrever o nome dela antes da sua frase.

#include <stdio.h>

#include <string.h>

int main ()

{

char string1[100],string2[100],string3[100];

printf ("Entre com uma string: ");

gets (str1);

strcpy (str2,str1);

strcpy (str3,"Voce digitou: ");

printf ("\n\n%s%s",str3,str2);

return(0);

}

**strcat**

A sua forma é:

 strcat (string\_destino,string\_origem);

Essa função continuará inalterada e será adicionado o dado no fim da primeira string. A primeira “string destino” é a frase para pedir o dado e a “string origem” é o dado.

#include <stdio.h>

#include <string.h>

int main ()

{

char str1[100],str2[100];

printf ("Entre com uma string: ");

gets (str1);

strcpy (str2,"Voce digitou a string ");

strcat (str2,str1); /\* str2 armazenara' Voce digitou a string + o conteudo de str1 \*/

printf ("\n\n%s",str2);

return(0);

}

**Strchr**

A função retorna um ponteiro para a primeira ocorrência do caracter procurado ou NULL caso não encontre.

#include <stdio.h>

#include <string.h>

int main () {

const char str[] = "http://www.bommm.com";

const char ch = 'b';

char \*ret;

ret = strchr(str, ch);

printf("String after |%c| is - |%s|\n", ch, ret);

return(0);

}

Neste exemplo ela só vai ser realizada se o “ch” for igual a um caracter descrita na função onde ele vai mostrar o que vem após dele.

**strcmp()**

A sua forma é:

strcmp (string1,string2);

A função strcmp() compara a string 1 com a string 2. Se as duas forem idênticas a função retorna zero. Se elas forem diferentes a função retorna não-zero.

#include <stdio.h>

#include <string.h>

int main ()

{

char str1[100],str2[100];

printf ("Entre com uma string: ");

gets (str1);

printf ("\n\nEntre com outra string: ");

gets (str2);

if (strcmp(str1,str2))

printf ("\n\nAs duas strings são diferentes.");

else printf ("\n\nAs duas strings são iguais.");

return(0);

}

No exemplo podemos usar uma condição “if” para fazer a comparação e informar se elas são iguais ou diferentes.

**Stricmp**

A função strcmp() é usada para comparar dois strings. Strings devem ser comparados caractere por caractere. A função strcmp() tem como argumento dois strings e retorna um inteiro.

#include <stdio.h>

#include <string.h>

int main(void)

{

char palavra1[100], palavra2[100];

int resultado;

printf("entre com uma palavra: ");

gets(palavra1);

printf("entre outra palavra: ");

gets(palavra2);

resultado = strcmp(palavra1, palavra2);

if (resultado == 0)

printf("igual\n");

else if (resultado > 0)

printf("o primeiro e' maior\n");

else

printf("o segundo e' maior\n");

}

Nesse exemplo também usamos o if para informar o resultado e comparar junto com a função.

**Strlen**

Esta função irá retornar a quantidade de caracteres existentes em uma string, não considerando o caractere NULL na contagem dos caracteres.

#include <stdio.h>

#include <string.h>

int main(void)

{

char nome[100];

int comprimento;

printf("Entre seu nome: ");

gets(nome);

Comprimento = strlen(nome);

printf("Seu nome tem %d caracteres.\n", comprimento);

}