

Computação 2

Aula 5

defines

vetor, strings e matriz por parâmetros de função

Prof^a. Fabiany
fabiany1@utfpr.com.br

#define

- A diretiva **#define** define um identificador e uma string que o substituirá toda vez que for encontrado no programa. A forma geral da diretiva é:

#define nome_macro string

Exemplo:

```
#define VERDADEIRO 1
```

```
#define FALSO 0
```

```
printf(“%d %d %d”,FALSO,VERDADEIRO, VERDADEIRO+1);
```

#define

- Uma vez que o nome da macro tenha sido definido, ele pode ser usado como parte da definição de outros nomes de macro.

- Exemplo:

```
#define UM 1
```

```
#define DOIS UM+UM
```

```
#define TRES UM+DOIS
```

#define

- Exemplo:

```
#define TAMANHO_MAX 100
```

```
void main()  
{  
  int i;  
  float vetor[TAMANHO_MAX];  
  ....  
  for(i=0;i<TAMANHO_MAX;i++)  
    printf("%i ",vetor[i]);  
}
```

Vetor, strings e Matriz

- Pode ser passado um vetor ou uma matriz inteira como argumento para uma função.
- Quando um vetor ou matriz é usada como argumento para uma função apenas o endereço é passado, não uma cópia do vetor ou matriz inteira.
- Quando é chamado uma função com um nome de vetor ou matriz, um ponteiro para o primeiro elemento no vetor ou matriz é passado para a função.

Vetor

- Para passar o vetor ou matriz na chamada da função no programa, basta especificar o nome do vetor ou matriz sem um índice.
- Por exemplo:

```
void main() {  
    int vetor[10];  
    func1(vetor);  
    ....  
}
```

Vetor

- Se uma função recebe um vetor, você pode declarar o parâmetro formal em uma entre três formas: como um ponteiro, como um vetor dimensionado ou como um vetor não-dimensionado.
- A forma de trabalhar/modificar o vetor dentro da função será igual, independente da forma que foi passado o vetor.

Vetor

- Por exemplo na declaração da função para receber um vetor:

```
void func1(int *x) //ponteiro
{
...
...
}
void func1(int x[10]) //vetor dimensionado
{
...
...
}
void func1(int x[]) //vetor não-dimensionado
{
...
...
}
```

Exemplo 1

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 12

float media ( float vet[], int n )
{
    int i = 0;
    float media = 0;

    for ( i = 0; i < n; i = i + 1 )
    {
        m = m + vet[i];
    }
    m = m / n;
    return m;
}

void main()
{
    float numeros [MAX];
    float media;
    int i;

    for ( i = 0; i < MAX; i = i + 1 )
    {
        printf ("Digite um valor \n:");
        scanf ("%f", &numeros[i]);
    }
    media = media ( numeros, NUM_ALUNOS );
    printf ("A media e %.2f \n", media);
}
```

Exemplo 2

```
#include <stdio.h>
#include <stdlib.h>
#define NUM_ALUNOS 3

float media ( float notas[], int n )
{
    int i = 0;
    float m = 0.0;

    for ( i = 0; i < n; i = i + 1 )
    {
        m = m + notas[i];
    }
    m = m / n;
    return m;
}
```

```
int main( )
{
    float notas [NUM_ALUNOS];
    float media_turma;
    int i;

    for ( i = 0; i < NUM_ALUNOS; i = i + 1 )
    {
        do
        {
            printf ("Digite a nota do %d o. aluno \n: ", i+1);
            scanf ("%f", &notas[i]);
        } while ( ( notas[i] < 0.0 ) || ( notas[i] > 10.0 ) );
    }
    media_turma = media ( notas, NUM_ALUNOS );
    printf ("A media da turma eh %.2f \n", media_turma);

    system ("Pause");
    return 0;
}
```

```
float media ( float notas[NUM_ALUNOS], int n )
{
    int i = 0;
    float m = 0.0;

    for ( i = 0; i < n; i = i + 1 )
    {
        m = m + ( notas[i] / n );
    }
    return m;
}
```

```
float media ( float notas[], int n )
{
    int i = 0;
    float m = 0.0;

    for ( i = 0; i < n; i = i + 1 )
    {
        m = m + ( notas[i] / n );
    }
    return m;
}
```

```
float media ( float *notas, int n )
{
    int i = 0;
    float m = 0.0;

    for ( i = 0; i < n; i = i + 1 )
    {
        m = m + ( notas[i] / n );
    }
    return m;
}
```

String

- Para passar uma string (ou vetor de caracteres) na chamada da função no programa, basta especificar o nome da string sem o índice.
- Por exemplo:

```
void main() {  
    char nome[10];  
    func1(nome);  
    ....  
}
```

String

- Por exemplo na declaração da função para receber um vetor de caracteres (string):

```
void func1(char *nome) //ponteiro
{
...
...
}
```

```
void func1(char nome[ ]) //vetor não-dimensionado
{
...
...
}
```

Matriz

- Quando uma matriz é usada como um argumento para uma função, apenas um ponteiro para o primeiro elemento é realmente passado.

- Por exemplo:

```
void main() {  
    int matriz[5][10];  
    func1(matriz);  
    ....  
}
```

Matriz

- Porém, uma função que recebe uma matriz como um parâmetro deve definir pelo menos o comprimento da segunda dimensão. Isso porque o compilador C saber o comprimento de cada linha para indexar a matriz corretamente.
- Por exemplo:

```
void func1(int matriz[][10])
```

```
....
```

```
}
```

Matriz

```
void func1(int matriz[][10])
```

```
....
```

```
}
```

- Ou

```
void func1(int matriz[5][10])
```

```
....
```

```
}
```

Exemplo

```
#include <stdio.h>
#include <stdlib.h>
#define NUM_ALUNOS 3
#define NOTAS 2

float media ( float notas[][NOTAS], int n )
{
    int i = 0;
    float m = 0;
    float notaaluno = 0;

    for ( i = 0; i < n; i = i + 1 )
    {
        notaaluno = ( notas[i][0] + notas[i][1] )/2;
        m = m + notaaluno/n;
    }
    return m;
}
```

```
int main()
{
    float notas [NUM_ALUNOS][NOTAS], media_turma;
    int i;

    for ( i = 0; i < NUM_ALUNOS; i = i + 1 )
    {
        do
        {
            printf ("Digite a 1a. nota do %d o. aluno \n: ", i+1);
            scanf ("%f", &notas[i][0]);
        }
        while ( (notas[i][0] < 0.0) || (notas[i][0] > 10.0) );

        do
        {
            printf ("Digite a 2a. nota do %d o. aluno \n: ", i+1);
            scanf ("%f", &notas[i][1]);
        }
        while ( (notas[i][1] < 0.0) || (notas[i][1] > 10.0) );
    }

    media_turma = media ( notas, NUM_ALUNOS );
    printf ( "A media da turma eh %.2f \n", media_turma);

    system ("Pause");
    return 0;
}
```

Exercícios

- 1) Faça uma função que receba como parâmetro uma string (palavra ou frase) depois inverta a string e mostre o resultado no programa principal.
- 2) Faça um programa que leia 5 nomes de cidades diferentes e guarde em um vetor, depois leia uma matriz 5x5 com as distâncias entre as cidades. Passar a matriz e o vetor para uma função que calcule os percursos (distâncias) entre as cidades (ex.: distancia d km entre cidade x ate cidade y). O usuário deverá informar os nomes da cidades para o cálculo da distância.