

FUNÇÕES

Modelo Top-Down de projeto: problemas grandes e complexos são quebrados sucessivamente em pequenos pedaços.

- A divisão do problema em subproblemas mais simples é feita por meio de refinamentos sucessivos.
- Alguns refinamentos sucessivos são transformados em módulos.
- Na linguagem C os módulos são implementados por meio de funções.

Exemplo de refinamentos sucessivos para um algoritmo para cálculo de salário líquido de um empregado (Farrer, 1989):

Algoritmo

 Leia os dados do funcionário

 Determine o salário

 Escreva o salário

Fim algoritmo

Ref. Determine o salário

 calcule as vantagens

 calcule as deduções

 salarioliquido \leftarrow vantagens-deduções

Fim ref.

A forma geral de uma função

tipo_de_retorno nome_da_função (lista_de_parâmetros)

tipo_de_retorno: especifica o tipo de valor que o comando return da função devolve;

lista_de_parâmetros: é uma lista de nomes de variáveis separados por vírgula e seus tipos associados que recebem os valores dos argumentos quando a função é chamada.

*tipo_de_retorno nome_da_função (tipo nome_var1,
tipo nome_var2, ..., tipo nome_varN)*

Exemplo: para a função

```
long int quadrado(int x);
```

Identifique os seguintes elementos:

- tipo de retorno;
- nome da função;
- lista de parâmetros;

Exercício: Escreva um programa que use uma função para determinar o quadrado de um inteiro.

```
#include <stdio.h>
long int quadrado(int x);
int main()
{
    int num;
    long int val;
    scanf("%d",&num);
    val = quadrado(num);
    printf("O quadrado de %d eh %ld\n",num,val);
    return (0);
}
long int quadrado(int x)
{
    long int valor;
    valor = x*x;
    return (valor);
}
```

Exercício escreva uma função que receba dois valores reais (ponto flutuante) e retorne o maior dentre eles.

```
#include <stdio.h>

float maior(float x,float y);

int main()
{
    float x,y,val;
    scanf("%f%f",&x,&y);
    val = maior(x,y);
    printf("O maior dentre %f e %f eh %f\n",x,y,val);
    return (0);
}

float maior(float a, float b)
{
    if ( a >= b )
        return(a);
    else
        return(b);
}
```

Argumentos de Funções e Variáveis Locais

As variáveis que compõem a lista de argumentos de uma função são chamadas de parâmetros formais da função.

Exemplo: `float maior(float x, float y);`

No exemplo `x` e `y` são parâmetros formais e se comportam como quaisquer outras *variáveis locais* dentro da função:

- são criadas na entrada e destruídas na saída e só podem ser referenciadas dentro da função.

O que são variáveis globais?

Chamada por valor e chamada por referência

- Chamada por valor: Copia o valor de um argumento no parâmetro formal da função

```
#include <stdio.h>
int incremento(int x);
int main()
{
    int val, num;
    scanf("%d",&num);
    val = incremento(num);
    printf("O incremento de %d eh
           %d\n",num,val);

    return (0);
}
```

```
int incremento(int x)
{
    x = x + 1;
    return (x);
}
```

- Chamada por referência

O endereço de um argumento é copiado no parâmetro formal.

```
#include <stdio.h>
void incremento(int *x);
int main()
{
    int num;
    scanf("%d",&num);
    incremento(&num);
    printf("O incremento de %d eh
           %d\n",num,num);

    return (0);
}
```

```
void incremento(int *x)
{
    *x = *x + 1;
}
```

Se x é o endereço de uma variável *x é o valor no endereço x

Resumo de chamada por valor e chamada por referência:

- Chamada por valor: alterações feitas nos parâmetros formais não afetam as variáveis usadas para chamar a função
- Chamada por referência: alterações feitas nos parâmetros formais da função afetam as variáveis usadas para chamar a função

Exercício: utilizando passagem por referência escreva uma função que troque (permuta) os valores de duas variáveis inteiras.

Passando matrizes para funções

Para passar uma matriz para uma função, especifique o nome da matriz sem colchetes.

Exemplo:

```
int vet_idades[30];  
mostra_idades(vet_idades,30);
```

Para uma função receber uma matriz por meio de uma chamada de função, sua lista de parâmetros deve especificar que uma matriz será recebida.

Exemplo:

```
void mostra_idades(int b[ ], int comp);
```

Exemplo: ler um vetor e usar uma função para o cálculo da média.

$$m = \frac{1}{N} \sum_{i=1}^N x[i]$$

Exemplo: ler um vetor e calcula a média e o desvio padrão. Usar a função para cálculo da média no código para cálculo do desvio padrão.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x[i] - m)^2}$$

Exemplo: ler uma cadeia de caractere. Ler um caractere. Usar uma função para contar o número de ocorrências do caracteres na cadeia.

Passando matrizes bidimensionais para funções

Uma função que recebe uma matriz bidimensional como parâmetro deve definir pelo menos o comprimento da segunda dimensão.

Exemplo: `void mostra_mat(int mat[][10]);`

Exemplo: ler uma matriz com no máximo 5 linhas e 5 colunas. Usar uma função para exibir a matriz.

Exercícios

- 1) Escreva uma função para calcular o volume de uma esfera a partir do raio. Usar $\pi = 3.1416$.
- 2) Escrever uma função para calcular o fatorial do inteiro N ($N \geq 0$)
- 3) Usar uma função para calcular

$$S = \sum_{i=1}^N \frac{i}{N - i + 1}$$

- 4) Usar uma função para ler uma cadeia de no máximo 40 caracteres usando `fgets()`. Usando uma função buscar pela presença do caractere `'\n'` antes do final de cadeia marcado por `'\0'`. Caso encontrado, removê-lo e marcar o novo final da cadeia.

5) Escrever uma função para buscar e retornar o maior valor em um vetor de números reais.

6) Usando uma função e passagem por referência escreva uma função para receber uma cadeia e copiar seus caracteres em ordem invertida em uma segunda cadeia. Obs. atentar para o marcador de final de cadeia.

7) Escreva um programa em que uma matriz com no máximo 7 linhas e no máximo 7 colunas seja lida por meio de uma função. Exiba a matriz usando também uma função.

Referencias Bibliográficas

Farrer, H. et. al., Algoritmos Estruturados. Editora Guanabara, 1989.

Schildt, H., C completo e total. McGraw Hill, 1991.