



MINISTÈRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE ITAJUBÁ

Criada pela Lei nº 10.435 – 24/04/2002

Fundamentos de Programação

Lista 8.1 – Ponteiros e Funções

Uma função em C pode retornar nenhum ou somente um valor, não sendo permitido retornar mais de um valor. Quando a solução do problema exige o retorno de mais de um valor, uma estratégia possível é fazer esse “retorno” através de um ponteiro que, nesse caso, passa a ser um parâmetro de saída da função. Por exemplo, considere o seguinte problema:

Elabore uma função que receba dois números inteiros e retorne o resto da divisão e a multiplicação do primeiro pelo segundo. Sabemos que isso não é possível em C, então implementamos a seguinte função:

```
void retornaValor(int x, int y, int *pr, int *pm){  
    *pr = x%y;  
    *pm = x*y;  
    return;  
}
```

Os parâmetros x e y são de entrada para a função retornaValor porque fornecem dados de entrada (os valores considerados nas operações). Já os parâmetros pr e pm são dois ponteiros que apontam para áreas de memória que deverão receber o resultado de cada operação:

pr: aponta para uma área de memória que vai receber o resultado do resto da divisão

pm: aponta para uma área de memória que vai receber o resultado da multiplicação

pr e pm são ditos parâmetros de saída porque, de fato, não fornecem valores de entrada para a função, mas são usados de maneira que a função possa atualizar dois valores.

A função principal deveria ser implementada da seguinte forma:

```
int main(){  
    int x = 6, y = 12, r, m; //r = resto da divisão e m = multiplicação  
    retornaValor(x,y,&r,&m);  
    printf("%d %d",r,m );  
    return 0;  
}
```

1. Usando a estratégia acima, escrever uma função que retorne o valor mínimo e máximo de uma sequência de n valores fornecidos pelo usuário. A leitura dos valores digitados no teclado deve ser feita dentro da função. Contudo, a quantidade de valores a ser lido (n) deve ser um parâmetro de entrada da função.
2. Usando a estratégia acima, escrever uma função que tem como parâmetros de entrada três valores inteiros a, b e c e retorna a posição do maior e a posição do menor valor. Exemplo: Se a = 7, b = 1 e c = 5, o procedimento deve retornar 2 como a posição do menor e 1 como a posição do maior. Use a função no programa principal considerando três valores digitados pelo usuário.



MINISTÈRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE ITAJUBÁ

Criada pela Lei nº 10.435 – 24/04/2002

3. Faça uma função que receba um valor inteiro como referência e retorne o resto da divisão deste número por 10. Altere também o valor da variável passada por referência, dividindo-a por 10.
4. Faça uma função que receba um parâmetro (inteiro por valor) com o total de minutos passados ao longo do dia e receba também dois parâmetros (inteiros por referência) no qual deve preencher com o valor da hora e do minuto corrente. Faça um programa que leia do teclado quantos minutos se passaram desde meia-noite e imprima a hora corrente (use a sua função).
5. Faça um programa que imprima invertidos os nomes dos algarismos de um número inteiro de três dígitos. Use a sua função no programa principal. Não é necessário validar a entrada do número. Ex: 234 saída: quatro três dois.
6. O que é impresso nos seguintes códigos?

-- Código 1

```
#include<stdio.h>
```

```
int x = 0;
```

```
int main(){
```

```
    int i, x = 1;
```

```
    printf("%d\n", x);
```

```
    for (i=0;i<4;i++){
```

```
        int x = 2;
```

```
        printf("%d/ ", x);
```

```
        {
```

```
            int x = 3;
```

```
            printf("%d/ ", x);
```

```
        }
```

```
    }
```

```
    printf("\n%d", x);
```

```
}
```



MINISTÈRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE ITAJUBÁ
Criada pela Lei nº 10.435 – 24/04/2002

-- Código 2

```
#include<stdio.h>
int w = 1, x=2,y=3,z=4;
void foo(int y){
    int z = 6;
    printf("%d %d %d %d \n", w, x, y, z);
}
int main(){
    int x = 7;
    foo(5);
    foo(x);
    printf("%d %d %d %d \n", w, x, y, z);
}
```