



MC-102 ALGORITMOS E PROGRAMAÇÃO
DE COMPUTADORES
IC-UNICAMP

AULA 19 - EXERCÍCIOS FUNÇÕES E PROCEDIMENTOS

1 Objetivos

Exercitar funções e procedimentos.

2 Motivação

Fortalecer a aprendizagem do aluno, fazendo com que esse tente encontrar soluções para os problemas usando as ferramentas aprendidas.

3 Exercícios

1. Mostre o que o programa abaixo imprime:

```
#include <stdio.h>

void f(int a);

int main()
{
    int a;
    a = 1;
    printf("1: a = %d \n", a);
    f(a);
    printf("3: a = %d \n", a);
    return 0;
}

void f(int a)
{
    a = 2;
    printf("2: a = %d \n", a);
    return;
}
```

Resposta:

```
1: a=1
2: a=2
3: a=1
```

2. Modifique a função `void f(int a)` e o programa principal acima para que a resposta agora seja:

```
1: a=1
2: a=2
3: a=2
```

Resposta:

```
#include <stdio.h>

void f(int *a);

int main()
{
    int a;
    a = 1;
    printf("1: a = %d \n", a);
    f(&a);
    printf("3: a = %d \n", a);
    return 0;
}

void f(int *a)
{
    *a = 2;
    printf("2: a = %d \n", *a);
    return;
}
```

3. Simule detalhadamente a execução do programa abaixo, indicando na a saída deste programa (o que aparecerá na tela ao executar este programa).

```
#include<stdio.h>

void f1(int *pa, int b, int c);
int f2(int a, int c);

int main()
{
    int a, b, c;
    a = 3;  b = 5;  c = 2;
    f1(&b, a, c);
    printf("a = %d b = %d c = %d \n\n", a, b, c);
    f1(&c, a, b);
}
```

```

    printf("a = %d b = %d c = %d \n\n", a, b, c);
    printf("f2(%d, %d) = %d \n\n", b, a, f2(b, a));
    printf("a = %d b = %d c = %d \n\n", a, b, c);
    return 0;
}

void f1(int *pa, int b, int c)
{
    printf("Inicio da funcao f1 \n");
    printf("a = %d b = %d c = %d \n", *pa, b, c);
    *pa = *pa + b + c;
    b = *pa + b + c;
    printf("a = %d b = %d c = %d \n", *pa, b, c);
    printf("Fim da funcao f1 \n\n");
}

int f2(int a, int c)
{
    int i, aux = 0;
    for (i = 1; i <= c; i++)
        aux = aux + a;
    return aux;
}

```

Resposta:

```

Inicio da funcao f1
a = 5 b = 3 c = 2
a = 10 b = 15 c = 2
Fim da funcao f1

```

```

a = 3 b = 10 c = 2

```

```

Inicio da funcao f1
a = 2 b = 3 c = 10
a = 15 b = 28 c = 10
Fim da funcao f1

```

```

a = 3 b = 10 c = 15

```

```

f2(10, 3) = 30

```

```

a = 3 b = 10 c = 15

```

4. Faça uma função que receba um vetor de inteiros de tamanho **N** e retorne o menor elemento do vetor. A função não deve ler nada do teclado, nem mostrar nada na tela. Depois faça um `main()` para testar a função.

Resposta:

```

#include<stdio.h>
#include<stdlib.h>
#define TAM 100

int menor(int v[],int tam)
{
    int m=v[0];
    int i;
    for(i=1;i<tam;i++)
        if(v[i]<m)
            m=v[i];
    return m;
}

int main()
{
    int vetor[TAM];
    int i;

    printf("\nGerando vetor com nums aleatorios entre 0 e 999 ... \n");
    for(i=0;i<TAM;i++)
        vetor[i]=rand()%1000;

    printf("O menor número gerado foi: %d \n\n",menor(vetor,TAM));

    return 0;
}

```

5. Faça uma função que receba um vetor de inteiros de tamanho **N** e retorne o menor e o maior elemento do vetor. A função não deve ler nada do teclado, nem mostrar nada na tela. Depois faça um `main()` para testar a função.

Resposta:

```

#include<stdio.h>
#include<stdlib.h>
#define TAM 100

void MenorMaior(int v[],int tam,int *m,int *M)
{
    int i;
    *m=v[0];
    *M=v[0];
    for(i=1;i<tam;i++){
        if(v[i]<*m)
            *m=v[i];
        if(v[i]>*M)
            *M=v[i];
    }
}

```

```

}

int main()
{
    int vetor[TAM];
    int i, menor, maior;

    printf("\nGerando vetor com nums aleatorios entre 0 e 999 ... \n");
    for(i=0; i<TAM; i++)
        vetor[i]=rand()%1000;

    MenorMaior(vetor, TAM, &menor, &maior);

    printf("Menor: %d\nMaior: %d \n\n", menor, maior);

    return 0;
}

```

6. Faça uma função que retorne o produto escalar de dois vetores do tipo inteiro que possuam o mesmo tamanho. Se os vetores tiverem tamanhos diferentes, retorne um código de erro. Dica, use um parâmetro para retornar o erro. A função não deve ler nada do teclado, nem mostrar nada na tela. Depois faça um `main()` para testar a função.

Resposta:

```

#include<stdio.h>
#include<stdlib.h>
#define TAM 100

int escalar(int v1[], int tam1, int v2[], int tam2, int *ok)
{
    int prod=0;
    int i;

    *ok=1;
    if (tam1!=tam2) {
        *ok=0;
        return -1;
    }
    for(i=0; i<tam1; i++)
        prod=prod+(v1[i]*v2[i]);
    return prod;
}

int main()
{
    int vetor[TAM];
    int vetor2[TAM];
    int i, certo, produto;

```

```

printf("\nGerando 2 vetores com nums aleatorios entre 0 e 999 ... \n");
for(i=0;i<TAM;i++) {
    vetor[i]=rand()%1000;
    vetor2[i]=rand()%1000;
}
produto=escalar(vetor,TAM,vetor2,TAM,&certo);
if(certo)
    printf("Vetor . Vetor2 = %d \n\n",produto);
else
    printf("ERRO!! Vetores de tamanho diferentes. \n\n");
return 0;
}

```