



## MAPA – Material de Avaliação Prática da Aprendizagem

|   |  |
|---|--|
| <b>Acadêmico: Marcos Vinicius de Moraes</b>               | <b>R.A. 20127542-5</b>                                     |
| <b>Curso: ENGENHARIA DE SOFTWARE</b>                      |  |
| <b>Disciplina: PARADIGMAS DE LINGUAGEM DE PROGRAMAÇÃO</b> |  |
| <b>Valor da atividade: 3,5</b>                            | <b>Prazo: 01/10/2021 à 23h59<br/>(horário de Brasília)</b> |

### Instruções para Realização da Atividade

1. Todos os campos acima deverão ser devidamente preenchidos;
2. É obrigatória a utilização deste formulário para a realização do MAPA;
3. Esta é uma atividade individual. Caso identificado cópia de colegas, o trabalho de ambos sofrerá decréscimo de nota;
4. Utilizando este formulário, realize sua atividade, salve em seu computador, renomeie e envie em forma de anexo;
5. Formatação exigida para esta atividade: documento Word, Fonte Arial ou Times New Roman tamanho 12, Espaçamento entre linhas 1,5, texto justificado;
6. Ao utilizar quaisquer materiais de pesquisa referencie conforme as normas da ABNT;
7. No ambiente virtual da disciplina você encontrará orientações importantes para elaboração desta atividade. Confira!
8. Critérios de avaliação: Utilização do template; Atendimento ao Tema; Constituição dos argumentos e organização das Ideias e atendimento às normas ABNT.
9. Procure argumentar de forma clara e objetiva, de acordo com o conteúdo da disciplina.

**Em caso de dúvidas, entre em contato com seu Professor Mediador.**

**Bons estudos!**

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    float peso, altura, imc;
    printf("\nInforme seu peso em Kg:");
    scanf ("%f", &peso);
    printf("\nInforme sua altura em m:");
    scanf ("%f", &altura);
    imc = peso / (altura*altura);
    if (imc < 19) {
        printf("\n\t\t\t MUITO MAGRO");
    } else {
        if (imc >= 19 && imc < 25) {
            printf("\n\t\t\t NORMAL");
        } else {
            if (imc >= 25 && imc < 30) {
                printf("\n\t\t\t SOBRE PESO");
            } else {
                if (imc >= 30 && imc <= 40) {
                    printf("\n\t\t\t NORMAL");
                } else {
                    if (imc > 40) {
                        printf("\n\t\t\t OBESIDADE GRAVE");
                    }
                }
            }
        }
    }
    printf("\n\t\t\t Seu IMC é: %.2f", imc);
    return 0;
}
```

Trecho de código 1

Trecho de código 2

Trecho de código 3

Trecho de código 4

Sub-rotinas são trechos de código independentes, que são desenvolvidos fora do corpo principal do programa(main), e podem ser chamadas quantas vezes forem necessárias de acordo com a necessidade do programa principal. Também conhecida como métodos, procedimentos ou funções, as sub-rotinas proporcionam uma redução significativa de código, uma vez que um bloco de comandos pode ser executado várias vezes em um programa, sem que o código seja repetido em toda chamada da rotina, o que torna também as manutenções mais simples.

Aplicando esse conceito no código acima, percebemos que é possível reduzir a complexidade do código com a possibilidade de criar sub-rotinas em quatro trechos diferentes do código, conforme especificado na imagem.

Trecho de código 1 – Existe a possibilidade de declarar a variável “peso” como global, permitindo que essa variável seja utilizada dentro de todas as sub-rotinas sem a necessidade de uma nova declaração ou de passagem de parâmetros. Porém isso torna o código mais vulnerável e lento, uma vez que a variável global é criada no início da execução do programa e seu espaço de memória fica reservado até o encerramento do mesmo. Por outro lado, uma variável local, é declarada dentro da sub-rotina desejada, sendo criada durante a chamada



da sub-rotina, e destruída quando a tarefa é encerrada, liberando seu espaço de memória. Portanto, no trecho de código 1, sugiro que seja criado um procedimento chamado de “lePeso”, utilizando a passagem de parâmetro por referência, referenciando a variável “peso” na memória.

O procedimento é um trecho de código que é executado sem a necessidade de ter um valor de retorno, somente executa as tarefas a ele designadas. Isso é possível devido à variável “peso” ter sido passada por referência, sendo assim, toda manipulação sofrida por essa variável, terá impacto não só dentro da sub-rotina, mas também no programa principal, deixando de ser necessário o retorno de qualquer valor através da sub-rotina. Entretanto, quando o sistema solicitar a leitura do peso em KG, e o usuário informar seu peso, o valor será atribuído diretamente à variável “peso” declarada no programa principal, devido a essa passagem de parâmetros por referência.

Ex: Trecho de código que cria o procedimento “lePeso”:

```
void lePeso(float *peso){  
    printf("\nInforme seu peso em Kg:");  
    scanf ("%f", &*peso);  
}
```

Trecho de código 2 – Assim como acontece no “Trecho de código 1”, a variável “altura”, que terá seu valor informado pelo usuário, deverá ser passada como parâmetro através do procedimento “leAltura”, sendo passado também por referência, excluindo a necessidade de retornar algum valor através da sub-rotina, pois o valor informado pelo usuário será atribuído diretamente à variável “altura” declarada dentro do corpo principal do programa, função main.

Ex: Trecho de código que cria o procedimento “leAltura”:

```
void leAltura(float *altura){  
    printf("\nInforme sua altura em m:");  
    scanf ("%f",&*altura);  
}
```



Trecho de código 3 – Para efetuar o cálculo do imc, poderá ser criada uma função que contenha a fórmula para que seja realizada a operação matemática, e posteriormente retorne o resultado para a variável “imc”, criada dentro do corpo principal do programa. Nesse caso, a sub-rotina utilizada deve ser uma função, porque ela vai retornar um resultado após a realização do cálculo, diferentemente do procedimento, que somente executa as instruções sem o retorno de valores.

Deverá ser declarada uma variável chamada imc, dentro do corpo principal do programa que receberá o resultado da função “calculaIMC”. Para que seja realizado esse cálculo, a função precisa receber os valores de peso e altura. Esses valores serão passados como parâmetro, por valor, método esse que manipula somente as variáveis dentro da função, diferentemente da passagem por referência que manipula as variáveis também fora da sub-rotina, em sua origem.

Exemplo: Chamada da função calculaIMC, passando peso e altura como parâmetros e retornando o resultado para a variável “imc”:

```
imc = calculaIMC(peso, altura);
```

Exemplo: Trecho de código que cria a função “calculaIMC”:

```
float calculaIMC(float peso2, float altura2){  
    float imc2;  
    imc2 = peso2 / (altura2*altura2);  
    return(imc2);  
}
```

Trecho de código 4 – Após realizar a atribuição de valores e cálculo do imc, o sistema deverá mostrar ao usuário o resultado de sua avaliação. De acordo com o valor de imc calculado, o sistema deverá informar ao usuário se ele está muito magro, normal, sobrepeso ou em estado de obesidade. Para isso, poderá ser criado um procedimento chamado de “retornoIMC”, o qual deve receber como parâmetro, o valor de imc obtido em sub-rotinas anteriores e através das estruturas condicionais, retornar a mensagem na tela, informando o quadro atual do usuário, juntamente



com o imc obtido. Nesse propósito, deverá ser feito através de procedimento, pois não será necessário o retorno de valores, somente a realização de comandos designados à sub-rotina, que se encerra ao final da sequência de comandos.

Exemplo: Trecho de código que cria o procedimento “retornoIMC”:

```
void retornoIMC(float imc){  
    if (imc <19 ){  
        printf("\n\t\t MUITO MAGRO");  
    }else{  
        if(imc >= 19 && imc<25){  
            printf("\n\t\t NORMAL");  
        }else{  
            if(imc >= 25 && imc<30){  
                printf("\n\t\t SOBRE PESO");  
            }else{  
                if(imc >= 30 && imc<=40){  
                    printf("\n\t\t NORMAL");  
                }else{  
                    if (imc>40){  
                        printf("\n\t\t OBESIDADE GRAVE");  
                    }  
                }  
            }  
        }  
    }  
}
```