

1. Considerando as variáveis declaradas na tabela abaixo e mais a variável booleana TESTE, com valor FALSO, avalie as expressões a seguir, para cada uma das três combinações de valores apresentadas:

	A	B	Nome	Profissão
01	3	16	"MIRIAM"	"ADVOGADO"
02	5	64	"PEDRO"	"MEDICO"
03	2,5	9	"ANA"	"PROFESSOR"

- I. $(A + 1 \geq ((B)^{(1/2)}))$ OU $(NOME \neq 'ANA')$
 II. $(A + 1 \geq ((B)^{(1/2)}))$ E $(PROFISSAO = 'MEDICO')$
 III. $(NOME \neq 'ANA')$ OU $(PROFISSAO = 'MEDICO')$ E $(A + 1 \geq ((B)^{(1/2)}))$
 IV. NÃO TESTE E $((A + 1) \geq ((B)^{(1/2)}))$ OU NÃO $(PROFISSAO = 'MEDICO')$
 V. NÃO $(A + 1 \geq ((B)^{(1/2)}))$ E TESTE

	I	II	III	IV	V
01					
02					
03					

2. O IMC – Índice de Massa Corporal é um critério da Organização Mundial de Saúde para dar uma indicação sobre a condição de peso de uma pessoa adulta. A fórmula é $IMC = \text{peso} / (\text{altura}^2)$. Elabore um algoritmo, na linguagem que se sentir mais confortável que leia o peso e a altura de um adulto e mostre sua condição de acordo com a relação abaixo:

- Abaixo de 18,5 **Abaixo do Peso**
- Entre 18,5 e 25 **Peso normal**
- Entre 25 e 30 **Acima do peso**
- Acima de 30 **obeso**

3. No contexto orientação a objeto, explique os tópicos a seguir:

- Classe
 - Objeto
 - Herança
 - Polimorfismo
 - Interfaces
4. Escreva um algoritmo, na linguagem que se sentir mais confortável, que leia um vetor de 15 posições e, em seguida, leia também dois valores X e Y quaisquer correspondentes a duas posições no vetor. Ao final seu programa deverá escrever a soma dos valores encontrados nas respectivas posições X e Y. Se X ou Y forem posições inválidas, informar o usuário “Posições inválidas”, pedir uma posição válidas novamente.
5. Existe uma classe responsável pelo salário de acordo com sua função e cargo. Abaixo está o código responsável por calcular o salário. Você acha que é um bom

```
public double calcular Salario(String tipoFuncionario, String cargo) {  
    if(tipoFuncionario.equals("desenvolvedor") {  
        if(carg.equals("fullstack1") {  
            return 2500,00;  
        }  
        if(carg.equals("fullstack2") {  
            return 4500,00;  
        }  
        if(carg.equals("fullstack3") {  
            return 6500,00;  
        }  
    }  
    if(tipoFuncionario.equals("gerente") {  
        return 12000,00;  
    }  
}
```

código? Você mudaria algo? Explique.

6. Faça um programa, na linguagem que achar mais confortável, que calcule o fatorial de n! (fatorial de n), sendo que o valor inteiro de n é fornecido pelo usuário. lembrando:
- $$4! = 1 * 2 * 3 * 4 = 24$$
- $$6! = 1 * 2 * 3 * 4 * 5 * 6 = 720$$
- $$1! = 1 \text{ (por definição)}$$

$0! = 1$ (por definição)

Não existe fatorial $N < 1$ (por definição)

7. Volte no exercício 6, e desenvolva exercício de fatorial usando recursividade.
8. Usando MySQL crie o banco a seguir:
 - endereco: (id: int, rua: varchar(30), cep: varchar(8), bairro: varchar(100), cidade:varchar(100), uf_estado:varchar(2), complemento:varchar(255));
 - editora: (id: int, nome: varchar(100), id_endereco: int referência endereço (id));
 - autor: (id:int, nome:varchar, id_endereco: int referência endereço(id));
 - livro: (id: int, titulo:varchar(100), id_editora: int referência editora (id), id_autor: int referência autor (id), categoria:varchar(30));
 - leitor: (id: int, nome: varchar(100), id_endereco: referência endereço(id));
 - empréstimo: (id: int, data_empresatimo :date, id_leitor: int referência leitor(id), id_livro: int referência livro (id));
9. Com base no exercício 8, execute uma consulta em MySQL, obtendo: o nome e endereço do leitor e data do empréstimo que pegou o livro “Criptografia e Segurança de Redes”
10. Com base no exercício 8, execute uma consulta em MySQL, obtendo: Quantos empréstimos houve neste mês.
11. Existem algumas ferramentas, arquiteturas que me permitem trabalhar com sistemas em rede: websocket, rest, soap, cliente servidor. O que significa estas tecnologias? Mencione ao menos um caso ou cenário para cada tecnologia, ferramenta ou arquitetura que se adeque o uso da mesma.

12. Considere uma matriz de distância entre cidades 6 x 6:

	1.(Cáceres)	2.(BBUGres)	3.(Cuiabá)	4.(VGrande)	5.(Tangará)	6.(PLacerda)
1.(Cáceres)	-1	63	210	190	0	190
2.(BBUGres)	63	-1	160	150	95	0
3.(Cuiabá)	210	160	-1	10	0	0
4.(VGrande)	190	150	10	-1	0	0
5.(Tangará)	0	95	0	0	-1	80
6.(PLacerda)	190	0	0	0	80	-1

Observe as regras a seguir:

- Primeira linha e primeira coluna representam cidades, que devem ser inseridos apenas por números. 1 representa Cáceres, 2 BBUGres, 3 Cuiabá, e assim por diante.
- A Intersecção entre as cidades representa a distância entre elas
- Tanto o número que representa a cidade quanto as distâncias devem ser números inteiros
- A distância igual a -1 representa que é a distância entre uma cidade e ela mesmo, o que não seria possível (diagonal principal).
- A Distância igual a 0 quer dizer que não existe um “caminho direto” entre as duas cidades.

Desenvolva um algoritmo, na linguagem que preferir, no qual o programa deverá perguntar a cidade de origem e cidade de destino, o usuário deverá informar o número correspondente a cada cidade, e o programa principal irá chamar uma função que deverá ser responsável por calcular a distância entre as duas cidades. Se a distância entre as cidades for igual a -1, seu programa deve informar que não é possível sair da cidade até ela mesmo, se a distância for diferente de 0, só informar o valor entre as cidades, se for igual a 0, o programa deverá passar por outras cidades para chegar no seu destino. Por exemplo, se a cidade de origem escolhida pelo usuário é 1- Cáceres e a cidade de destino é 5- Tangará distância é igual a 0. Por isto, seu programa deverá calcular outra rota, passando entre outras cidades, por exemplo: 1- Cáceres até 2- BBUGres e 2- BBUGres até 5- Tangará. Ao final o programa deve exibir a distância percorrida.