



## RECUPERAÇÃO – PRIMEIRO BIMESTRE

### Linguagem de Programação

Comandos de Entrada e Saída / Comandos de Condição

Paranaguá, 08 de Maio de 2018

---

### Instruções para resolução da Avaliação:

1. A prova é resolvida de forma individual, não havendo troca de informações entre os alunos, nem mesmo troca de material como cadernos, livros ou *pen drive*.
2. As saídas dos programas a serem desenvolvidos devem, obrigatoriamente, ser idênticas as apresentadas nos exemplos de cada questão, caso contrário o programa não será considerado correto.
3. O professor não resolverá dúvidas com relação a possíveis erros de compilação ou sintaxe, nem mesmo com relação à interpretação das questões, por isso todas elas possuem exemplos de entradas com suas correspondentes saídas.
4. Para resolução das questões, devem ser utilizados, obrigatoriamente, apenas os conceitos de linguagem C vistos até o momento na disciplina, ou seja, laços, vetores, funções, entre outros não serão considerados.
5. A cada questão resolvida o aluno deve solicitar (imediatamente) ao professor que efetue a sua correção. O professor executará o código-fonte e, se necessário, fará alguma pergunta sobre a codificação apresentada. Caso a resolução esteja correta, o professor já computará o conceito do exercício para aluno.

---

### QUESTÕES

1. **(Médio – 20%)** Em um campeonato de pontos corridos todos os clubes participantes jogam entre si. De acordo com os resultados das partidas são somados os pontos de cada uma das equipes, é através dessa pontuação que são definidos: o campeão, o vice-campeão e o terceiro colocado. Considerando um torneio envolvendo três clubes: Barcelona, Real Madrid e Bayern de Munique, onde o usuário insere o resultado dos três jogos: Barcelona x Real Madrid, Barcelona x Bayern de Munique e Real Madrid x Bayern de Munique. Considere que em caso de vitória o clube vencedor recebe três (3) pontos, e o perdedor nenhum, já em caso de empate ambos os clubes recebem um ponto (1). A partir dessas informações construa uma aplicação em C que calcule a pontuação de cada um dos três times e indique o campeão do torneio. Em caso de empate entre dois ou mais clubes, o aplicativo deve mostrar essa informação. Veja o exemplo a seguir:

- Barcelona x Real Madrid	- Placar: 5 4
- Barcelona x Bayern de Munique	- Placar: 3 3
- Real Madrid x Bayern de Munique	- Placar: 3 2
<b>- Campeão: Barcelona (4 pontos)</b>	

- Barcelona x Real Madrid	- Placar: 4 3
---------------------------	---------------

- Barcelona x Bayern de Munique - Placar: 3 3
- Real Madrid x Bayern de Munique - Placar: 1 3
- *Campeão: Empate (4 pontos)!*

- Barcelona x Real Madrid - Placar: 4 4
- Barcelona x Bayern de Munique - Placar: 1 2
- Real Madrid x Bayern de Munique - Placar: 1 4
- *Campeão: Bayern de Munique (6 pontos)*

- Barcelona x Real Madrid - Placar: 4 4
- Barcelona x Bayern de Munique - Placar: 1 1
- Real Madrid x Bayern de Munique - Placar: 2 2
- *Campeão: Empate (2 pontos)*

2. **(Fácil – 20%)** Os computadores são construídos através de circuitos lógicos. Dentre esses circuitos temo-se como principais portas: AND (E), OR (OU), NAND (NÃO E), NOR (NÃO OU) e XOR (OU EXCLUSIVO). Essas portas recebem valores binários de entrada e de acordo com os mesmos produzem uma saída também binária, como mostram as tabelas a seguir.

ENTRADAS		PORTA / SAÍDA				
A	B	AND	OR	NAND	NOR	XOR
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0

Baseado na tabela apresentada, codifique um programa em C que receba as quatro saídas binárias para as entradas (0 0, 0 1, 1 0 e 1 1) e indique a porta lógica que produz essas saídas. Caso as saídas não se encaixem em nenhuma das cinco portas tabeladas, a mensagem: “Porta Desconhecida!” deve ser apresentada. Veja os exemplos a seguir:

- Digite as quatro saídas: 0 0 0 1  
- *Porta AND*
- Digite as quatro saídas: 0 1 0 1  
- *Porta Desconhecida!*
- Digite as quatro saídas: 0 1 1 1  
- *Porta OR*
- Digite as quatro saídas: 0 1 1 0  
- *Porta XOR*
- Digite as quatro saídas: 1 0 0 0  
- *Porta NOR*

- Digite as quatro saídas:
- *Porta NAND*

1 1 1 0

3. **(Fácil – 20%)** Construa uma aplicação em C que efetue a conversão das escalas de medidas, milímetro (l), centímetro (c) e metro (m). O programa deve receber do usuário: o valor e a medida (l, c, m) de entrada e a medida (l, c, m) para o qual ele deseja efetuar a conversão. Se o usuário escolher uma medida diferente de l, c ou m, o aplicativo deve apresentar a mensagem: “*Medida de Entrada Inválida!*” ou “*Medida de Saída Inválida!*”. A tabela abaixo mostra várias conversões entre milímetro, centímetro e metro que servem de referência para que o aluno entenda como funciona esse tipo de conversão.

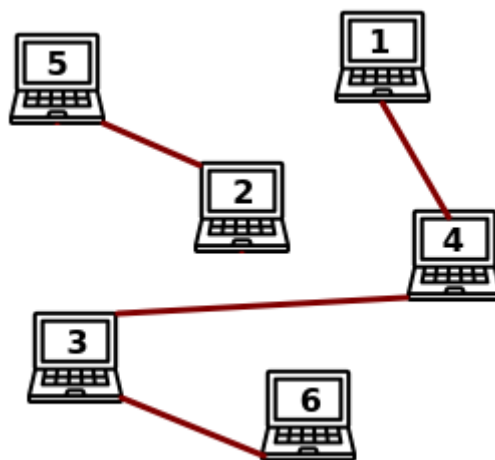
Milímetro (l)	Centímetro (c)	Metro (m)
1000	100	1
5000	500	5
10	1	0,01
5	0,5	0,005
10000	1000	10

**Veja os exemplos a seguir:**

- Valor e Medida de Entrada: 2 m
- Medida de Saída: c
- *2,000 metros em centímetros: 200,000*
  
- Valor e Medida de Entrada: 50 l
- Medida de Saída: m
- *50,000 milímetros em metros: 0,050*
  
- Valor e Medida de Entrada: 10 c
- Medida de Saída: l
- *10,000 centímetros em milímetros: 100,000*
  
- Valor e Medida de Entrada: 15 c
- Medida de Saída: m
- *15,000 centímetros em metros: 0,150*
  
- Valor e Medida de Entrada: 15 m
- Medida de Saída: m
- *15,00 metros em metros: 15,000*
  
- Valor e Medida de Entrada: 15 h
- Medida de Saída: m
- *Medida de Entrada Inválida!*

- Valor e Medida de Entrada: 15 m
- Medida de Saída: x
- *Medida de Saída Inválida!*

4. **(Médio – 20%)** Uma rede de computadores permite conectar várias máquinas entre si. Para tal, são criadas várias rotas que possibilitam que uma mensagem trafegue de um computador até outro. Considere a possibilidade de termos uma rede de computadores onde todos são ligados diretamente uns aos outros, ou seja, sem a utilização de dispositivos eletrônicos intermediários de comunicação (roteador, switch, etc.). Nessa configuração, as rotas são criadas de acordo com as conexões existentes entre as máquinas. Considere a rede a seguir:



A partir da configuração de rede apresentada na figura anterior, codifique um programa em C que solicite ao usuário os números de duas máquinas da rede (origem e destino) e indique se elas são capazes de trocar mensagens entre si (se há uma rota entre elas). Caso o usuário digite o número de uma máquina que não existe na rede ou a máquina de origem for igual à máquina de destino, a mensagem: “*Entrada Inválida*” deve ser exibida. Veja os exemplos a seguir:

- Máquinas de origem e destino: 2 5
  - *[OK] é possível enviar uma mensagem*
  - *[Rota] 2-5*
- Máquinas de origem e destino: 5 2
  - *[OK] é possível enviar uma mensagem*
  - *[Rota] 5-2*
- Máquinas de origem e destino: 1 6
  - *[OK] é possível enviar uma mensagem*
  - *[Rota] 1-4-3-6*

- Máquinas de origem e destino: 6 4  
- *[OK] é possível enviar uma mensagem*  
- *[Rota] 6-3-4*
- Máquinas de origem e destino: 1 1  
- *Entrada Inválida!*
- Máquinas de origem e destino: 2 1  
- *[ERRO] não é possível enviar uma mensagem!*
- Máquinas de origem e destino: 4 5  
- *[ERRO] não é possível enviar uma mensagem!*
- Máquinas de origem e destino: 1 10  
- *Entrada Inválida!*

5. **(Difícil – 20%)** Um palíndromo é uma palavra ou número que pode ser lido da esquerda para a direita ou da direita para a esquerda e ainda possui o mesmo significado. Por exemplo: 12321, 5665, 909, 44, 1, arara, Ana. Considerando o conceito de palíndromo, codifique um programa em C que solicite ao usuário um número de **até 5 dígitos** e informe se ele é um palíndromo ou não. Veja os exemplos a seguir:

- Digite um número inteiro de até 5 dígitos: 1  
- *(É) um número Palíndromo!*
- Digite um número inteiro de até 5 dígitos: 12  
- *(NÃO É) um número Palíndromo!*
- Digite um número inteiro de até 5 dígitos: 626  
- *(É) um número Palíndromo!*
- Digite um número inteiro de até 5 dígitos: 622  
- *(NÃO É) um número Palíndromo!*
- Digite um número inteiro de até 5 dígitos: 4884  
- *(É) um número Palíndromo!*
- Digite um número inteiro de até 5 dígitos: 4848  
- *(NÃO É) um número Palíndromo!*
- Digite um número inteiro de até 5 dígitos: 16261  
- *(É) um número Palíndromo!*