



Departamento de Engenharias,  
Arquitetura e Computação

## PROGRAMAÇÃO PARA RESOLUÇÃO DE PROBLEMAS

Prof. Ivan L. Süptitz

Parte Prática da Prova 1  
26/10/2020

Leia **COM ATENÇÃO** as instruções a seguir:

- A interpretação das questões faz parte da prova;
- A prova deve ser realizada das 19h até as 22h20 sem necessidade de comparecimento físico na universidade;
- A prova é **individual**. Os códigos fonte serão submetidos a um **software verificador de plágio**. Em caso de indícios de cópia a nota será zerada ou dividida.
- Além de responder a parte objetiva no questionário do moodle (peso de 40%), faça a entrega do arquivo fonte da questão prática (60%) na tarefa correspondente no ambiente virtual;
- Na questão de implementação será levado em consideração:
  - Correta execução do programa (compilação e execução).
  - Indentação, organização e documentação do código (comentários);
  - Apropriado pensamento computacional (correta interpretação do problema e apropriada proposta da solução computacional – de acordo com os conceitos e estruturas de programação vistos em aula).

**Questão Prática:** Uma indústria de implementos agrícolas convidou você para desenvolver um programa para gerenciar o consumo de energia elétrica da fábrica, sendo que a mesma está caminhando para o conceito de indústria 4.0 de forma que todos as tomadas de energia que alimentam as máquinas estão sendo monitoradas constantemente por meio de uma rede de sensores.

Seu programa recebe um vetor referente aos 10 pontos de energia disponíveis, com cada posição indicando o consumo da máquina ligada naquele ponto. Por exemplo:

```
int vet[] = { 58, 44, 15, 0, 8, 0, 10, 20, 30, 0}; //unidades em MWh
```

### Funcionalidades:

- Ao abrir o programa o sistema imprime a atual potência instalada (soma de todas as tomadas);
- Depois o usuário informa a potência da nova máquina a ser ligada;
- O sistema busca uma tomada que não esteja em uso (potência igual a zero);
  - Se encontrar, a nova máquina deve ser alocada nessa tomada;
  - Caso contrário pedir para o usuário qual máquina deve ser substituída e alocar a nova máquina nessa posição;
- Ao final o programa imprime a nova capacidade instalada e a lista de todas as tomadas (posição e potência).

### Observações:

- Utilize modularização (funções) ao menos para encontrar a tomada livre e para somar a potência total;
- O vetor pode ser fixado no código (não precisa pedir para o usuário) mas lembre-se de testar situações em que haverão tomadas livres (como no exemplo anterior) e situações em que todas as tomadas estão ocupadas (exigindo a interação com o usuário).
- O exemplo a seguir é de uma interação completa com o usuário quando os valores do vetor são: **42, 13, 15, 70, 8, 4, 10, 20, 30, 50** e o usuário escolheu instalar uma nova máquina de **60 MWh**. Como não tinha mais tomadas disponíveis ele escolheu **substituir a posição 9** (última do vetor iniciando em 0).

*Figura 1 - Exemplo de interação do programa*

```
A potencia instalada eh: 262 MWh
Informe a potencia (em MWh) da nova maquina a ser ligada: 60
Nenhuma tomada livre, informe a posicao da que vamos substituir: 9

-----
Listagem das tomadas:
Posicao: 0 Carga: 42
Posicao: 1 Carga: 13
Posicao: 2 Carga: 15
Posicao: 3 Carga: 70
Posicao: 4 Carga: 8
Posicao: 5 Carga: 4
Posicao: 6 Carga: 10
Posicao: 7 Carga: 20
Posicao: 8 Carga: 30
Posicao: 9 Carga: 60
-----
A nova potencia instalada eh: 272 MWh
```

Avaliação
<b>(1,0 pts)</b> Correta soma/impressão da capacidade instalada
<b>(0,5 pts)</b> Correta leitura dos dados de entrada
<b>(0,5 pts)</b> Correta busca por tomada livre
<b>(0,5 pts)</b> Correta substituição da nova máquina no vetor
<b>(1,0 pts)</b> Correta listagem das tomadas
<b>(1,5 pts)</b> Correto uso de Modularização
<b>(1,0 pts)</b> Indentação, documentação e organização
<b>Total 6,0 pontos</b>

**Boa prova!**