PETROBRAS

Engenharia de Equipamentos - Eletrônica
Engenheiro(a) Júnior - Área: Automação

Algoritmos Computacionais e Estruturas de Dados

Questões Resolvidas Passo a Passo

Produzido por Exatas Concursos
www.exatas.com.br
v4

Índice De Questões

Engenharia de Equipamentos - Eletrônica - CEBRASPE - Petrobras 2021

Q111 (pág. 1) Q112 (pág. 2) Q113 (pág. 2) Q114 (pág. 3)

Engenheiro(a) de Equipamentos Júnior - Eletrônica - Petrobras 2018.1

Q56 (pág. 4) Q57 (pág. 5) Q60 (pág. 7)

Engenheiro(a) Júnior - Area: Automação - Transpetro 2018.1

Q68 (pág. 8)

Engenheiro(a) de Equipamentos Júnior - Eletrônica - Petrobras 2014.2

Q58 (pág. 10) Q59 (pág. 11) Q61 (pág. 12) Q67 (pág. 13)

Engenheiro(a) de Equipamentos Júnior - Instrumentação - Innova 2012

Q52 (pág. 9)

Engenheiro(a) de Equipamentos Júnior - Eletrônica - Petrobras 2012.1

Q58 (pág. 13) Q59 (pág. 15) Q62 (pág. 14) Q63 (pág. 16) Q65 (pág. 17)

Engenheiro(a) de Equipamentos Júnior - Eletrônica - Petrobras 2011

Q56 (pág. 17) Q59 (pág. 18) Q62 (pág. 19)

Engenheiro(a) de Equipamentos Júnior - Eletrônica - Petrobras 2010.2

Q63 (pág. 19) Q66 (pág. 20)

Engenheiro(a) de Equipamentos Júnior - Eletrônica - Petrobras 2010.1

Q28 (pág. 21) Q29 (pág. 22) Q30 (pág. 23)

Engenheiro(a) Júnior - Area: Automação - Transpetro 2011

Q55 (pág. 24)

Engenheiro(a) Júnior - Area: Automação - Transpetro 2008

Q33 (pág. 25)

Engenheiro(a) Júnior - Area: Automação - Transpetro 2006

Q36 (pág. 27)

ENGENHEIRO(A) DE EQUIPAMENTOS JÚNIOR - ELETRÔNICA - TERMOAÇU 2008.1

Q52 (pág. 29) Q54 (pág. 28) Q55 (pág. 30)

ENGENHEIRO(A) DE EQUIPAMENTOS JÚNIOR - ELETRÔNICA - REFAP 2007

Q35 (pág. 31)

ENGENHEIRO(A) DE TERMELÉTRICA JÚNIOR - ELETRÔNICA - TERMOCEARÁ 2009

Q38 (pág. 31) Q40 (pág. 26)

PROFISSIONAL JÚNIOR - ENGENHARIA ELETRÔNICA - BR DISTRIBUIDORA 2008

Q56 (pág. 32) Q57 (pág. 35) Q58 (pág. 33) Q59 (pág. 34)

ENGENHEIRO(A) DE EQUIPAMENTOS PLENO - ELETRÔNICA - PETROBRAS 2006

Q39 (pág. 34) Q40 (pág. 36) Q42 (pág. 37)

QUESTÕES RESOLVIDAS NESTA APOSTILA: 42

QUESTÃO 4

ENGENHARIA DE EQUIPAMENTOS - ELETRÔNICA - CEBRASPE - PETROBRAS 2021

IV) Na linguagem de programação C++, com a finalidade de evitar *loops*, a diretiva `#include` é substituída pelo conteúdo de um arquivo que é indicado sempre após o programa ser compilado.

RESOLUÇÃO

A diretiva `#include` é utilizada em linguagens de programação como o C++ para incluir código de outras bibliotecas ou arquivos no código fonte atual. Esses arquivos podem conter declarações de funções, classes, constantes ou outras definições que são necessárias para o código em questão.

A sintaxe da diretiva `#include` é bastante simples. Ela é precedida pelo caractere # e seguida pelo nome do arquivo a ser incluído. Geralmente, esse nome é colocado entre aspas duplas (“”) ou entre sinais de menor e maior (< >), dependendo de onde o arquivo se encontra.

Aqui está um exemplo simples de uso da diretiva `#include`:

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```

Nesse exemplo, a diretiva `#include <iostream>` é usada para incluir a biblioteca padrão de entrada e saída do C++, que contém as definições das classes `iostream`, `ostream` e `istream`. A linha `using namespace std;` é usada para facilitar o uso dessas classes no código, permitindo que você utilize `cout` em vez de `std::cout`, por exemplo.

Quando o compilador encontra uma diretiva `#include`, ele procura pelo arquivo especificado na lista de diretórios de inclusão do sistema e inclui o seu conteúdo no arquivo de origem. Isso significa que, se você incluir vários arquivos com a mesma função ou classe, o compilador irá incluir todas as definições dessas funções ou classes em seu arquivo final.

É importante observar que a diretiva `#include` não é uma função ou instrução do C++, mas sim **uma diretiva do pré-processador**, que é responsável por realizar operações de pré-processamento no código fonte **antes da compilação**. Além disso, é importante utilizar a diretiva `#include` com cuidado, para evitar conflitos de definição de funções ou classes e para garantir que o código seja mantido organizado e fácil de ler.

AFIRMAÇÃO ERRADA