

Atividade III - 4 pontos

Leia e responda as perguntas com cuidado.

O e-mail do participante (**izaell.official@uni9.edu.br**) foi registrado durante o envio deste formulário.

Nome *

IZAEL ALVES DA SILVA

RA *

922114939

Perguntas sobre C++

Qual é a função da diretiva using em C++? *

- ☒ Simplificar o acesso a elementos ou namespaces
- ☐ Simplificar o acesso a namespaces
- ☐ Simplificar o acesso a elementos
- ☐ Simplificar o acesso a classes



O que é uma exceção em C++? *

- ☐ Um erro de compilação.
- ☐ Uma instrução usada para sair de loops.
- ☒ Um objeto que representa um erro ou condição excepcional.
- ☐ Uma forma de controlar o fluxo de execução em um programa.

O que é o tempo de execução em C++? *

- ☐ O momento em que o código-fonte é escrito.
- ☒ O período durante o qual o programa está sendo executado pelo usuário.
- ☐ A fase de otimização do código durante a compilação.
- ☐ O tempo necessário para escrever um programa em C++.

O que são erros de tempo de execução em C++? *

- ☐ Erros que ocorrem durante a compilação e impedem a criação do executável.
- ☒ Erros lógicos que não são identificados até que o programa esteja em execução.
- ☐ Erros que são corrigidos automaticamente pelo compilador.
- ☐ Erros relacionados à sintaxe do programa.



O que é o tempo de vida de uma variável em C++? *

- ☒ O período durante o qual uma variável pode ser usada em um programa.
- ☐ O tempo necessário para compilar um programa.
- ☐ O tempo durante o qual uma variável permanece na memória após a execução do programa.
- ☐ O tempo entre a declaração e a inicialização de uma variável.

O que o seguinte código irá fazer? *

```
#include <iostream>
#include <regex>

int main() {
    std::string texto = "A linguagem C++ foi criada para construir sistemas operacionais";
    std::regex cppRegex(R"(C\+\+)");
    std::string substituirTexto = std::regex_replace(texto, cppRegex, "C");
    std::cout << "Texto apos a substituicao: " << substituirTexto << std::endl;
    return 0;
}
```

- ☐ Irá substituir a palavra C por C++, formando uma nova frase
- ☒ Irá substituir a palavra C++ por C, formando uma nova frase
- ☐ Irá lançar uma exceção



O que o seguinte código irá fazer? *

```
#include <iostream>
#include <regex>

int main() {
    std::string text = "O telefone é (11) 98765-4321 e o número de emergência é (11) 99123-1234.";

    std::regex phoneRegex(R"(\d{2}\) \d{5}-\d{4})");

    std::sregex_iterator phoneIterator(text.begin(), text.end(), phoneRegex);
    std::sregex_iterator end;

    while (phoneIterator != end) {
        std::smatch match = *phoneIterator;
        std::cout << match.str() << std::endl;
        ++phoneIterator;
    }
}
```

- ☐ Irá extrair informações que dê match com o padrão textual passado
- ☒ Irá extrair informações que dê match com a expressão regular passado
- ☐ Irá validar informações que dê match com a expressão regular passada
- ☐ Irá substituir informações que dê match com a expressão regular passada

Pergunta sobre síntese em compiladores

O que é síntese em compiladores? *

- ☒ Processo de transformar código-fonte em código de máquina diretamente.
- ☐ Etapa do compilador que otimiza o código-fonte.
- ☐ Geração de código-fonte a partir de uma descrição de alto nível.
- ☐ Análise léxica do programa.



Qual é o objetivo principal da síntese em compiladores? *

- ☐ Melhorar a eficiência da execução do programa.
- ☒ Transformar o código-fonte em código de máquina.
- ☐ Facilitar a leitura do código-fonte.
- ☐ Encontrar erros de sintaxe no programa.

O que a fase de otimização de código realiza? *

- ☐ Tradução direta para código de máquina.
- ☐ Geração de código-fonte a partir de uma descrição de baixo nível.
- ☐ Verificação de erros léxicos.
- ☒ Tenta melhorar a eficiência do código gerado.

O que é a geração de código intermediário? *

- ☐ Tradução direta para código de máquina.
- ☐ Representação do código-fonte após otimizações.
- ☐ Criação de um código-fonte mais legível.
- ☒ Conversão do código-fonte para uma forma intermediária antes da geração final de código de máquina.



Qual é a principal diferença entre a análise léxica e a síntese em um compilador C++? *

- ☐ A análise léxica lida com erros de sintaxe, enquanto a síntese gera código de máquina.
- ☐ A análise léxica verifica a semântica do programa, enquanto a síntese otimiza o código.
- ☐ A análise léxica analisa a estrutura do código-fonte, enquanto a síntese gera representações intermediárias.
- ☒ A análise léxica transforma o código-fonte em tokens, enquanto a síntese traduz os tokens em código de máquina.

O que é uma instrução de máquina no contexto da síntese em compiladores? *

- ☐ Um comando específico em C++.
- ☐ Uma etapa durante a análise léxica.
- ☐ Um tipo de otimização de código.
- ☒ Uma instrução escrita em linguagem de montagem.

O que é uma "pilha de execução"? *

- ☐ Uma estrutura de dados usada durante a análise léxica.
- ☐ Uma técnica de otimização de código.
- ☐ Um local de armazenamento temporário para variáveis.
- ☒ Uma área de memória utilizada para controlar a execução de funções.



Qual é a finalidade da geração de código intermediário durante a síntese em compiladores? *

- ☐ Facilitar a leitura do código-fonte.
- ☐ Permitir a execução direta do código em linguagem de máquina.
- ☒ Realizar otimizações antes da geração final de código de máquina.
- ☐ Identificar erros léxicos no programa.

Pergunta sobre Compilador, gramática e Linguagem

A seguinte afirmação é verdadeira ou falsa? *

"Apesar de haver fases específicas na implementação de um compilador, é possível personalizar o compilador para necessidades específicas".

- ☐ Não é possível. Um compilador de uma linguagem deve ser único e conter tudo o que a linguagem foi especificada para fazer
- ☒ Sim, é possível personalizar um compilador para tarefas específicas, por exemplo, podemos ter um compilador especializado em otimizar código e outro especializado em depuração
- ☐ Não é possível. Um compilador de uma linguagem deve seguir sempre os mesmos estágios e conter tudo o que a linguagem foi especificada para fazer
- ☐ Sim, é possível personalizar um compilador para tarefas específicas, desde que siga o padrão americano ASCII, por exemplo, podemos ter um compilador especializado em otimizar código e outro especializado em depuração



Compiladores e interpretadores são maneiras de traduzir o código fonte para a máquina, logo em computação, podemos dizer que são a mesma coisa? *

- ☐ Sim, é uma prática muito comum e possui diversos motivos, alguns deles são:
Arquitetura do processador, motivação (release ou debug) entre outros
- ☐ Não, um compilador analisa parte do código com objetivo de traduzi-lo de parte em parte, enquanto o interpretador faz o mesmo trabalho de conversão aos poucos, sempre que uma declaração ou função é executada
- ☐ Sim, é uma prática muito comum e possui diversos motivos, alguns deles são:
Arquitetura do processador, motivação (release ou debug) entre outros. Ambos significam a mesma coisa só que para linguagens diferentes
- ☒ Não, um compilador analisa todo o código com objetivo de traduzi-lo de uma vez, enquanto o interpretador faz o mesmo trabalho de conversão aos poucos, sempre que uma declaração ou função é executada

Das opções abaixo, qual faz parte da fase de síntese de um compilador tradicional? *

- ☐ Análise sintática
- ☐ Análise léxica
- ☐ Análise Semântica
- ☒ Otimizador de código



Em computação, especificamente na disciplina de compiladores, o que é um token? *

- ☐ Tokens são etapas no processo de análise semântica
- ☐ Tokens são conjuntos de palavras sem associação com a linguagem de programação
- ☒ Tokens são conjuntos de caracteres associados para produzir uma saída
- ☐ Tokens não são padrões de sequências responsável por associar a palavras chaves

O que é um alfabeto? *

- ☐ É um conjunto infinito, vazio, de símbolos
- ☐ É um conjunto finito, não vazio, de símbolos Gregos
- ☒ É um conjunto finito, não vazio, de símbolos
- ☐ É um conjunto finito, não vazio, de símbolos latinos

Dado as opções abaixo, qual é o único parâmetro que NÃO faz parte da formulação de uma gramática? $G(_, _, _, _)$ *

- ☐ Símbolo inicial
- ☒ Linguagem
- ☐ Alfabeto
- ☐ Elementos terminais



Na disciplina de compiladores, mais especificamente em teoria de linguagem e gramática, o que significa criar uma gramática ambígua? *

- ☐ Quando a gramática NÃO permite gerar mais de uma derivação da mesma sentença
- ☒ Quando a gramática permite gerar mais de uma derivação da mesma sentença
- ☐ Quando a gramática permite gerar somente uma derivação da mesma sentença
- ☐ Quando a gramática permite gerar exatamente duas sentenças

Na disciplina de compiladores, o que é uma linguagem? *

- ☐ Linguagem é um conjunto de elementos (símbolos) e um conjunto de métodos (regras) para montar uma frase com alfabetos latinos
- ☒ Linguagem é um conjunto de elementos (símbolos) e um conjunto de métodos (regras) para combinar tais elementos
- ☐ Linguagem é uma forma de realizar comunicação entre o computador e o ser humano
- ☐ Linguagem é uma forma de realizar comunicação entre o computador e o ser humano utilizando alfabetos

O que são expressões regulares e qual é o seu propósito principal? *

- ☒ Sequências específicas de caracteres usadas para definir padrões de texto.
- ☐ Ferramentas de compilação utilizadas para otimizar código fonte.
- ☐ Algoritmos de ordenação para strings.
- ☐ Linguagens de programação especializadas em manipulação de texto.



O que significa o operador * em uma expressão regular? *

- ☒ Corresponde a zero ou mais ocorrências do caractere anterior.
- ☐ Indica uma correspondência exata do caractere anterior.
- ☐ Representa uma classe de caracteres.
- ☐ Força a correspondência ao início da string.

Como você indica que uma parte da expressão regular deve corresponder ao final * da string?

- ☒ \$
- ☐ \Z
- ☐ ^
- ☐ +

O que são expressões regulares em C++? *

- ☒ Sequências específicas de caracteres usadas para definir padrões de texto.
- ☐ Variáveis que armazenam valores constantes em um programa.
- ☐ Operadores utilizados para realizar operações matemáticas com padrões de texto.
- ☐ Funções para gerar aleatoriedade em strings.



Qual biblioteca em C++ é comumente utilizada para trabalhar com expressões regulares? *

- ☐ iostream
- ☐ stdlib
- ☒ regex
- ☐ math

Como você declara um objeto de expressão regular em C++ usando a biblioteca <regex>? *

- ☐ regex exp = "padrão";
- ☒ regex exp("padrão");
- ☐ exp_regex = new regex("padrão");
- ☐ regex exp; exp.pattern("padrão");

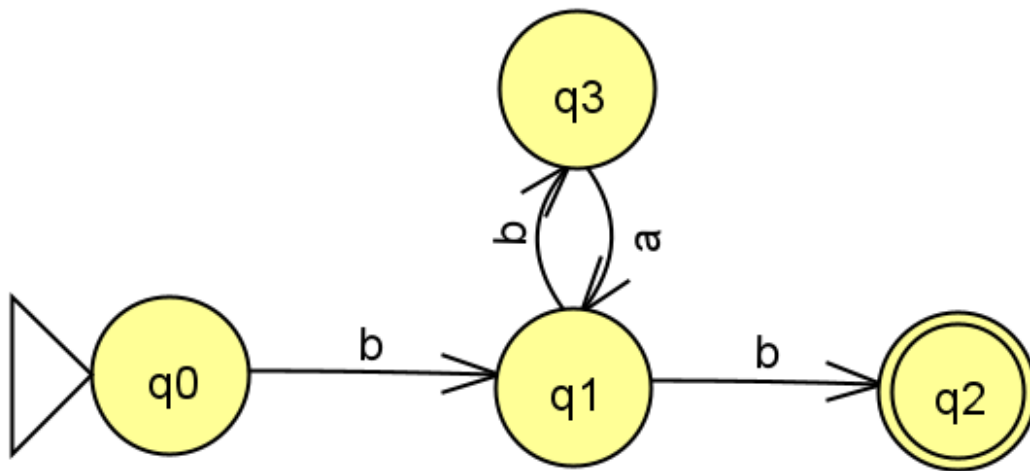
Qual função é usada para verificar se uma string corresponde a uma expressão regular em C++? *

- ☒ match()
- ☐ search()
- ☐ find()
- ☐ regex_match()

Perguntas sobre AFD



Use o seguinte AFD para responder as próximas perguntas



Escolha abaixo qual é a a linguagem que o AFD acima descreve *

- ☐ $L = \{b(ab)^nb \mid n \geq 0\}$
- ☒ $L = \{ba^nba \mid n \geq 0\}$
- ☐ $L = \{a^mb^n \mid m+n \text{ é par}\}$
- ☐ $L = \{ab^mba(ab)^n \mid m \geq 0, n \geq 0\}$

Dado o AFD da imagem acima, quais são seus estados? (Mais de um resposta) *

- ☐ s
- ☒ q0
- ☒ q1
- ☒ q2
- ☒ q3

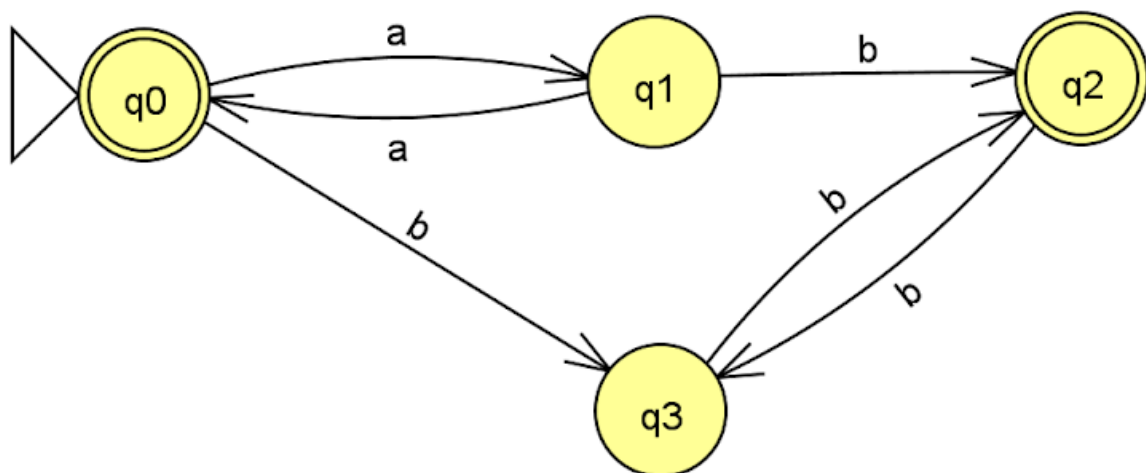
Dado o AFD da imagem acima qual é o estado inicial? *

- ☐ s
- ☒ q0
- ☐ q1
- ☐ q2
- ☐ q3

Dado o AFD da imagem acima qual é o estado terminal? *

- ☐ s
- ☐ q0
- ☐ q1
- ☒ q2
- ☐ q3

Use o seguinte AFD para responder as próximas perguntas



Escolha abaixo qual é a a linguagem que o AFD acima descreve *

- ☐ $L = \{b(ab)^nb \mid n \geq 0\}$
- ☐ $L = \{ba^nb a \mid n \geq 0\}$
- ☐ $L = \{a^mb^n \mid m+n \text{ é par}\}$
- ☒ $L = \{ab^mba(ab)^n \mid m \geq 0, n \geq 0\}$

Dado o AFD da imagem acima, qual estado não é válido? *

- ☐ q0
- ☐ q1
- ☐ q2
- ☒ q3
- ☐ q4

Dado o AFD da imagem acima qual é o estado terminal? (Mais de uma resposta) *

- ☐ S
- ☐ q0
- ☐ q1
- ☒ q2
- ☒ q3



Dado a seguinte linguagem $L = \{ab^m ba(ab)^n \mid m \geq 0, n \geq 0\}$, selecione abaixo * uma sentença válida

- ☐ bbbaabab
- ☒ abab
- ☐ abbbaab
- ☐ abbb
- ☐ abbaa

Este formulário foi criado em Uninove. [Denunciar abuso](#)

Google Formulários

