

Disciplina	Curso	Turno	Período	Valor
Lab. Linguagens de Programação	Engenharia da Computação	Tarde	4º	15 Pontos
Aluno(a)		Matrícula		
Aluno(a)		Matrícula		

Ao entregar esse exame eu dou minha palavra que eu o fiz somente com minha dupla, entendendo que eu posso consultar qualquer material publicamente disponível, exceto aqueles disponibilizados por colegas que estão fazendo esse curso ou que já o fizeram no passado.

Prova 1

1ª Questão) (15pts) Considere um subconjunto simples do sistema de pré-processamento da linguagem C. Esse pré-processamento inclui as macros `#if`, `#ifdef`, `#ifndef`, `#elif`, `#else`, `#endif`, `#include`, `#define`, `#undef`, `#error` e o operador `defined()`. A seguir é dado um exemplo de cabeçalho que contém apenas diretivas de pré-processamento.

```
#undef VALUE
#define VALUE 42
#ifdef SYSTEM1
# include "system1.h"
#elif defined (SYSTEM2)
# include <system2.h>
# if ! defined (SUB)
# error "Invalid subsystem"
#endif
#ifndef TESTE
#define TESTE "Testing"
#endif
#else
# error "Invalid system"
#endif
```

Detalhes sobre as macros:

Macro	Descrição	Exemplos
<code>#define</code>	<code>#define</code> seguido de um nome e opcionalmente de um valor que pode ser inteiro ou <i>string</i> .	<code>#define TESTE</code> <code>#define TESTE 10</code> <code>#define TESTE "teste"</code>
<code>#undef</code>	<code>#undef</code> seguido de nome.	<code>#undef TESTE</code>
<code>#error</code>	<code>#error</code> seguido de uma <i>string</i> .	<code>#error "Erro"</code>
<code>#include</code>	<code>#include</code> seguido de uma <i>string</i> ou de um nome entre chaves chevron (<code><nome></code>).	<code>#include "cabecalho.h"</code> <code>#include <cabecalho.h></code>
<code>#ifdef</code>	<code>#ifdef</code> seguindo de um nome, um bloco de outras macros, <code>#elif(s)</code> e <code>#else</code> opcionais e termina com <code>#endif</code> .	<code>#ifdef TESTE</code>
<code>#ifndef</code>	<code>#ifndef</code> seguindo de um nome, um bloco de outras macros, <code>#elif(s)</code> e <code>#else</code> opcionais e termina com <code>#endif</code> .	<code>#ifndef TESTE</code>
<code>#if</code>	Similar ao <code>#ifdef</code> , mas seguido do operador <code>defined(nome)</code> ou <code>!defined(nome)</code> .	<code>#if defined (TESTE)</code> <code>#if !defined (TESTE)</code>
<code>#elif</code>	Similar ao <code>#if</code> , mas somente pode existir em conjunto com um <code>#if</code> , <code>#ifdef</code> ou <code>#ifndef</code> acompanhante.	<code>#elif defined (TESTE)</code> <code>#elif !defined (TESTE)</code>
<code>#else</code>	<code>#else</code> seguido de um bloco de macros e termina com <code>#endif</code> sem <code>#elif(s)</code> entre eles.	<code>#else</code>
<code>#endif</code>	Marcador de fechamento de uma macro <code>#if</code> , <code>#ifdef</code> ou <code>#ifndef</code> .	<code>#endif</code>

- a) (5pts) Desenhe um autômato para separar os elementos léxicos dessa linguagem.
- b) (5pts) Projete uma gramática em notação *EBNF* para fazer o *parsing* dessa linguagem.
- c) (5pts) Desenvolva um programa capaz de identificar se um arquivo nessa linguagem está bem formado, respondendo **Sim** em caso positivo e **Não** caso contrário. O programa deve receber o arquivo por parâmetro. O *parser* deve ser desenvolvido de forma própria, não sendo permitido o uso de bibliotecas de terceiros para esse propósito.