

## Grupos

Grupo	Aluno1	Aluno2	Aluno3
1	Camila	Lucas	Mayara
2	Francisco	Leandro	Rodrigo
3	Daniel	Matheus	Henrique
4	Alexandre	Igor	Vitor
5	Lucas	Pedro H.	-
6	Felipe	Gabriel	Márcio
7	Bianca	Luis	Rafael
8	Bruno	Lucas	-
9	Fábio	Leonardo	Pedro Paulo
10	José Eduardo	Vitor	-
11	Cássio	-	-
12	Conrado	Diego	Guilherme
13	Bruno	Cláudio	Julio
14	Fábio	Matheus	Vanessa
15	Tamires	-	-
16	Habner	Victor	-
17	Fábio	Jairo	-
18	Felipe	Mikail	Thiago
19	Caroline	Michele	-
20	Ubiratan	Ulisses	Rafael
21	Adilson	Renato	Rodrigo
22	Huang	Letícia	Luiz
23	Evandro	-	-
24	Danilo	-	-
25	Denis	Sidney	Vicinius
26	Débora	Leandro	-

### Grupo 1

- Cuidado com o uso da palavra "através"!
- Através é usado para representar que algo passou (atravessou) sobre algo, ex.: Viu-se a chuva através do espelho; A luz através da água, etc.
- Então, no texto a sentença "A busca nesse vetor foi realizada através de busca binária", deve ficar assim: A busca nesse vetor foi realizada POR MEIO de busca binária
- Vocês estão especificando os tipos de erro usando expressões regulares para isso, cuidado
- O analisador de vocês está acusado erro em caracteres especiais ( $\backslash n, \backslash t, \backslash r$ ). Provavelmente, essas situações irão ocorrer quando vocês migrarem de uma plataforma para a outra, por exemplo: quebra de linha em Linux =  $\backslash n$ , quebra de linha em Windows =  $\backslash r \backslash n$
- Existem formas de tirar esses caracteres usando algumas macros como SKIP. Com isso, não teriam problemas com esses tipos de caracteres
- Vocês indicaram 1y como um "identificador mal formado, começando com número". Está correto em partes, pois isso também poderia ser um número inteiro mal formado. Cuidado quando forem especificar os tipos de erro!

### Grupo 2

- integer e real são palavras reservadas
- Identificador com apenas um dígito não estão sendo reconhecidos
- (015) caracteres invalido na linha 1 -> No teste, essa linha possuía apenas program nome;
- 1y como entrada, tem-se a saída: 1 DIGSEQ; 'y' CARACTER INVÁLIDO
- Não verifica muito bem a presença de números reais mal formados
- Não utilizaram uma tabela de símbolos reservados

- Não verifica o tamanho do identificador
- Cuidado a forma que vocês especificaram sua linguagem a tornou não sensível a caixa. Ou seja: fOr, For, FOR, foR são todos iguais
- A mensagem de erro de não fechar comentário poderia indicar onde o comentário foi aberto
- Poderia fazer uma expressão regular para comentários. A ideia seria: começa com '{' seguido por qualquer coisa diferente de '}' e terminando com '}'

### Grupo 3

- Palavras reservadas estão sendo identificadas apenas como cmd. Entretanto, é melhor identificar que tipo é o *token*
- `ly` gera a saída `nro_int:l` e `id:y`
- Não identificou erros como `111.11.11` mas reconheceu `111.a`
- Reconheceram identificadores muito grandes
- Estão tratando número real mal formado, mas não tratam número inteiro mal formado
- Não indica comentário não fechado
- Cuidado com as imagens do terminal. Vocês usaram o terminal com efeito de transparência, o que deixou a imagem ruim

### Grupo 4

- Colocar a especificação dos tokens
- O compilador está aceitando identificadores iniciados por `_`
- Comentário não fechado
- Melhorar a identificação dos tokens de expressões matemáticas
- Vocês estão identificando o tamanho do identificador. Ok, estão fazendo um `for` contando quantos caracteres tem no identificador, mas varrem todo o identificador para posteriormente verificar seu tamanho, verifique isso dentro do `for` pois é mais eficiente
- Cuidado, o analisador de vocês ficam em loop quando é passado um arquivo não existente

### Grupo 5

- Relatório incompleto
- Colocar a especificação das expressões regulares
- O analisador léxico aceita números reais dessa forma `134312.`
- Está apresentando erros de caracteres invisíveis (acrescente `\\t\\n\\r`)
- Comentário não fechado
- Melhore as mensagens de erro (indique a linha, o tipo do erro, etc.)
- Falha de segmentação em um dos casos de teste (cuidado)

### Grupo 6

- Colocar a especificação das expressões regulares
- Legal, vocês usaram tabela de símbolos. Porém, está acusando que identificadores de parâmetros de função não foram declarados e procedimentos que foram declarados
- Comentário não fechado
- Não identificam número inteiro/real mal formado

### Grupo 7

- Quando usarem uma tabela que é composta somente por palavras, a tabela deve ser chamada de quadro
- Quando referenciarem uma figura/quadro/tabela no texto, usem letra maiúscula: Figura, Quadro, Tabela, etc.
- Comentário não fechado

- Melhorar as mensagens de erro, é interessante ter mensagens mais explicativas e indicar a linha do erro

#### Grupo 8

- Comentário não fechado
- Não reconhece número real mal formado
- Não limitou identificadores grandes
- Não reconhece identificadores mal formados
- Identificadores não podem ter letra maiúscula?
- Melhore as respostas de erros de caracteres

#### Grupo 9

- Comentário não fechado
- Não verifica identificador mal formado
- Aceita identificador iniciado por \_
- O Compilador está apontando erros de caracteres não imprimíveis. Cuidado com isso

#### Grupo 10

- Comentário não fechado

#### Grupo 11

- Comentário não fechado
- O analisador identificou 1.1.1 como número real
- No Windows, quebra de linha usa `\r\n`. Coloque isso nos caracteres brancos no arquivo `lex`

#### Grupo 12

- Algumas vezes, o analisador está usando tabulações para compor identificadores e outras coisas
- Melhorar as mensagens de erro

#### Grupo 13

- Comentário não fechado
- Na regra NOTDOUB está tratando coisas do tipo `ab.cd` como erro
- Melhorar a indicação das mensagens de erro. Por exemplo: `identificado@` gera a saída carácter inválido

#### Grupo 14

- Comentário não fechado
- Alguns caracteres não reconhecidos não foram alertados erros
- É interessante diferenciar os operadores matemáticos (pelo menos por precedência)

#### Grupo 15

- Comentário não fechado
- Erro em caracteres especiais
- `begin` foi identificado por `BEGN`, acho que esqueceu do `I`
- "Carácter estranho" é uma mensagem ruim para o compilador

#### Grupo 16

- Comentário não fechado
- Caracteres especiais

- Identificadores podem ter letras maiúsculas

#### Grupo 17

- Comentário não fechado
- Erro de comentário onde não há comentário
- Erro de aspas simples onde não há
- Identificador não inicia com \_
- Nessa linguagem não se usa + para identificar que um número é positivo
- Vocês tratam número mal formado, mas apenas tratam após o .

#### Grupo 18

- Comentário não fechado
- Estouro de pilha em alguns casos de teste (cuidado)

#### Grupo 19

- Erro sintático, por quê?
- @ não é um identificador mal formado, é um carácter inválido

#### Grupo 20

- Cadê a tabela de palavras reservadas
- read e write também são palavras reservadas
- Quando tem algum erro, o analisador para de executar
- real é um erro?
- Comentário não fechado
- Um número real é um erro?
- Encontraram identificadores grandes, mas deixe a mensagem de erro melhor

#### Grupo 21

- Identificadores podem ter letra maiúscula
- Comentário não fechado
- Estouro de memória em alguns casos de teste

#### Grupo 22

- Comentário não fechado e fechado, deram a mesma saída de erro léxico
- Acrescente o \r na lista de caracteres em branco
- Não estão usando a tabela de palavras reservadas

#### Grupo 23

- cuidado, aparentemente você não diferenciou maiúsculas de minúsculas.
- coloque \r na lista de símbolos em branco também
- classificou número inteiro como número real
- não identificou caracteres inválidos como @

#### Grupo 24

- Comentário não fechado
- Alguns caracteres especiais estão sendo apontados como erro
- Ao tratar erros de número real mal formado, seu analisador está lendo até quebras de linha após o erro
- Não foi reportado identificadores muito grandes

#### Grupo 25

- Comentário não fechado
- Melhore a mensagem de erro, o que é (6, 7)?
- Não reconheceu números com mais de um dígito
- Identificador iniciando com \_

#### Grupo 26

- write e read são palavras reservadas
- comentário não fechado
- identificador não pode iniciar com \_