

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACOM - FACULDADE DE COMPUTAÇÃO

COMPILADORES I - 2025/2
PROFA. BIANCA DE ALMEIDA DANTAS

Trabalho Prático
Descrição da 3ª Etapa – Tabela de Símbolos

1 DESCRIÇÃO

A terceira etapa do trabalho prático de nossa disciplina consiste em utilizar a tabela de símbolos apenas para diferenciar os identificadores das palavras reservadas da linguagem X++. Para tal, é necessário compreender o esqueleto de código liberado e adaptar o código para a sua plena utilização.

Não será cobrada análise semântica no trabalho prático, caso queiram utilizar a tabela de símbolos para realizar algo, o grupo pode receber uma bonificação na pontuação.

Os códigos fontes do programa desenvolvido serão compilados usando o compilador g++.

2 EXECUÇÃO E ENTRADA

O seu programa deve ser capaz de realizar a compilação de um arquivo de texto com a extensão **.xpp**, cujo nome será fornecido na linha de comando do terminal logo após o nome do executável de seu compilador. Por exemplo, se seu executável possuir o nome **xpp_compiler** e o arquivo de entrada for **teste1.xpp**, a seguinte instrução será digitada no terminal:

```
./xpp_compiler teste1.xpp
```

3 SAÍDA

O compilador deve emitir mensagens de erros, caso encontre algum, informando claramente o erro e a linha de ocorrência. Caso não sejam encontrados erros, o compilador deve imprimir que a compilação foi encerrada com sucesso. Todas as mensagens devem ser mostradas no terminal. Sugere-se usar como inspiração mensagens geradas por compiladores reais (como o próprio g++).

4 AVALIAÇÃO

A não implementação desta etapa não inviabiliza a correção do trabalho, ao contrário do que ocorre com as etapas anteriores.

O seu programa será compilado usando o comando:

```
g++ *.cpp -o xpp_compiler
```

Caso a compilação gere erros e o executável não seja gerado, o trabalho receberá nota zero.

O programa será executado com n arquivos fontes, podendo conter erros ou não, e a nota atribuída será proporcional ao número de testes cuja execução de seu compilador conseguir detectar

os erros (ou a falta deles) corretamente. Testes em que a execução não gerar o resultado esperado serão zerados.

O programa deve receber a entrada e gerar a saída **exatamente** como especificado nas descrições das etapas, caso isso não ocorra, a nota será penalizada.

5 ESPECIFICAÇÕES

- O trabalho prático poderá ser realizado em grupos de, no máximo, 3 alunos **sem exceções**.
- A linguagem C++ deverá ser utilizada na implementação do trabalho.
- A entrega de todas as etapas deve ser realizada até o dia: **14/11/2025**.

6 AVALIAÇÃO

O trabalho prático será avaliado utilizando os seguintes critérios:

- Trabalhos que não compilarem ou que tiverem apenas o analisador léxico receberão nota **zero**.
- Um conjunto contendo *numTestes* programas de teste será executado, com possíveis erros, e para cada teste seu programa deve emitir mensagens de erro (se houver) e, ao final, informar o término do processo de compilação.
- A cada teste para o qual o seu compilador executar corretamente, o grupo receberá pontuação equivalente $\frac{1}{numTestes} * nota\ total$. Executar corretamente significa terminar o percurso do programa de entrada e emitir mensagens para todos os erros do programa, se houver.
- Execuções parcialmente corretas receberão uma porcentagem da nota total.
- Se o seu programa travar na execução de um teste, a pontuação para tal teste será zerada.