

Lista de exercícios para a avaliação da 1ª unidade

1. Escreva códigos na linguagem C que inicializam uma matriz A quadrada de 100000 linhas em um espaço de endereços contíguo utilizando os 4 tipos de código abaixo. Compile os códigos com seu compilador favorito e temporize a execução dos programas resultantes com o comando `time` do Linux. Observe se há diferenças de desempenho e explique a importância das suas observações no projeto de compiladores.

a.

```
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        A[i][j] = 0;
```

b.

```
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        A[j][i] = 0;
```

c.

```
p = &A[0][0];
t = n * n;
for (i=0; i<t; i++)
    *p++ = 0;
```

d.

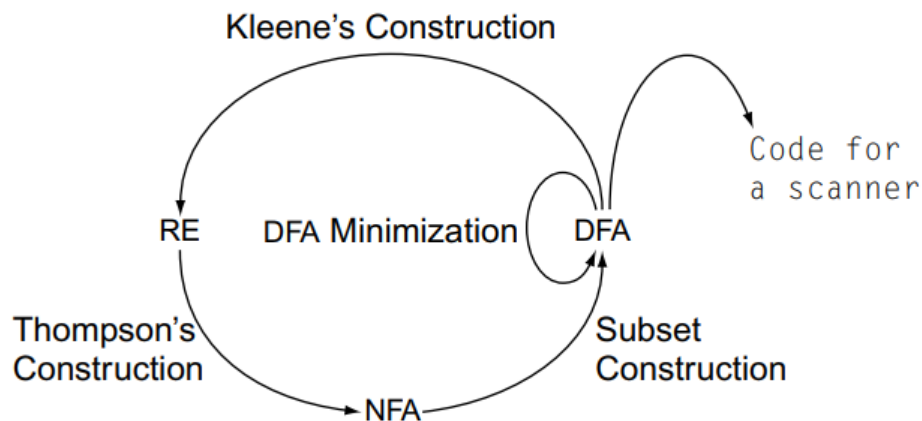
```
bzero((void*) &A[0][0],(size_t) n*n*sizeof(int))
```

2. Exercício 3 do capítulo 1 do livro *Engineering a Compiler*.
3. Quais são as principais etapas no processo de compilação? Elabore a sua resposta com um diagrama de blocos. Explique qual a função de cada etapa e cada sub-bloco.
4. Dado o seguinte código reconhecedor de nome de registradores abaixo e as tabelas ACTION e STATE sobrepostas abaixo do código, explique o processo de reconhecer ou não as seguintes palavras:
 - a. r100
 - b. r3
 - c. r15
 - d. r07

```
Char <- next character
State <- s0
while (Char ≠ EOF)
    Next    <- STATE(State,Char)
    Act <- ACTION(State,Char)
    perform action Act
    State <- Next
    Char <- next character
if (State is a final state )
    then report success
    else report failure
```

Action	State	r	0,1	2	3	4,5,6 7,8,9	other
0	1 start	e	e	e	e	e	e
1	e	2 add	2 add	5 add	4 add	e	e
2	e	3 add	3 add	3 add	3 add	e exit	e
3,4	e	e	e	e	e	e exit	e
5	e	6 add	e	e	e	e exit	e
6	e	e	e	e	e	x exit	e
e	e	e	e	e	e	e	e

5. Discorra sobre a figura abaixo enfatizando o papel e a importância de cada uma das construções mencionadas para o projeto de um *scanner*.



6. Apresente o algoritmo de Construção de Subconjuntos para transformação de um NFA em um DFA e explique seu funcionamento com um exemplo.
7. Explique, com um exemplo, como funciona a abordagem de minimização de um AFD.
8. Apresente o algoritmo de *scanner* controlado por tabela e codificado diretamente para o reconhecimento de um fluxo de palavras e explique seu funcionamento com um exemplo.